UNIVERSITÄT ZU LÜBECK
INSTITUTE OF COMPUTER ENGINEERING

From the Institute of Computer Engineering
of the University of Lübeck

Director: Prof. Dr.-Ing. Erik Maehle

Biologically Inspired Approaches for Hexapod Walking Robots Based on Organic Computing Principles and Proprioception Feedback

Dissertation
for Fulfillment of
Requirements
for the Doctoral Degree
of the University of Lübeck

from the Department of Computer Science

Submitted by

Ahmad Al Homsy

From Hama (Syria)

Lübeck 2016

IM FOCUS DAS LEBEN

First refree:	Prof. Dr.-Ing. Erik Maehle
Second refree:	Prof. Dr.-Ing. Alexander Schlaefer

Date of oral examination:	January 12, 2016
Approved for printing. Lübeck,	January 15, 2016

Statutory Declaration

I hereby certify that I am the sole author of this thesis and I declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. To the best of my knowledge, my thesis does not infringe upon copyright law. I also declare that all material and results that are not original to this work have been cited and referenced. This work was done under the guidance of prof. Dr.-Ing. Erik Maehle at the Institute of Computer Engineering.

Ahmad Al-Homsy

Acknowledgement

First of all, I would like to thank my advisor, Prof. Dr.-Ing. Erik Maehle, for giving me the opportunity to study in his institute and for giving me his trust in the completion of my research. I express my sincere thanks for his continued guidance throughout this process. During the years that I have spent in his institute, I have learned a lot from him. First and foremost, he taught me how to deal with problems.

Many thanks go to Dr.-Ing. Adam El Syed Auf for his support and many interesting discussions, especially at the beginning of this research. I also thank Jan Frost for helping me to solve some problems that faced me during my research.

A special thank is also due to my friend Amar Amory for his continued support and for the interesting discussions. We have spent most enjoyable times during this work.

Moreover, special thanks go to Ines Schiebahn, Anjes Kiencke, Dr.-Ing. Christian Renner, and Dr.-Ing. Patrick Weiss for their special support during this work.

I thank all my colleagues at the Institute of Computer Engineering for providing a comfortable atmosphere and for having wonderful times.

Special thanks and my highest appreciation and respect go to the people who are dearest to my heart: my mother and my father. They have provided me continued support and constant encouragement.

Most importantly, from the bottom of my heart, I would like to thank my lovely wife Abeer, the love of my life. She has been accompanying me for 8 years and always been showing her strong support. She has always been patient with me with regard to the difficulties of life and is doing her utmost in bringing up our two children, my son Tamim and my daughter Jana, who fill my heart with happiness and love of live. In the glitter of their eyes, I always see the innocence of children and magnificence of life. I feel so lucky to have such a wonderful family.

Last, I thank all those who contributed to the completion of my PhD thesis.

Abstract:

Insect-like walking of six-legged robots on rough and complex terrain is considered a challenging task. To achieve statically stable walking on such terrain, a six-legged robot has to cope with various types of walking substrate. First of all, the robot has to infer the current walking substrate to ensure stable and a safe walking.

This work focuses on the most common tasks that the walking robot may face in natural environments such as walking on inclined, slippery, sandy, and inclined sandy surfaces. Walking on inclined surface requires defining the direction of the inclination and controlling the robot body posture according to the inclination direction to avoid falling down. Walking on compliant surfaces such as a sandy surface requires identifying the walking substrate and executing adaptive walking.

This work will shed light on our applied decentralized controller approaches that enable the hexapod robot to detect and execute adaptive walking on the above mentioned terrain. Our approaches enable the hexapod robot to detect inclined and sandy surfaces and to achieve adaptive walking on these terrains. Furthermore, this work presents experiments, which combine three approaches to enable a hexapod robot to walk on one of the most difficult terrains, which is sandy inclined surface.

Additionally, this work demonstrates an approach to detect slippery surfaces and proposes a strategy to overcome this challenge.

Finally, the effect of leg disturbances such as hitting an object or a leg fault on stable walking is presented in the context of this work.

The novelty of our approaches is the evaluation of the local current consumption and angular position of each leg's joint as the only proprioception feedback. Our biologically inspired approaches are based on our Organic Robot Control Architecture (ORCA) and were tested on a low-cost version of the Organic Self-Configuring and Adaptive Robot (OSCAR).

Zusammenfassung:

Insektenartiges Laufen sechsbeiniger Roboter auf unwegsamem, komplexem Gelände stellt eine anspruchsvolle Herausforderung dar. Um auf solchem Gelände statisch stabil zu laufen, muss der sechsbeinige Roboter verschiedene Arten von Untergründen bewältigen. Zur Sicherstellung des stabilen und sicheren Laufens muss der Roboter zunächst den jeweiligen Untergrund identifizieren. Der Schwerpunkt dieser Arbeit liegt auf den häufigsten Aufgaben, die der Laufroboter in natürlichen Umgebungen zu bewältigen hat, wie z.B. das Laufen auf geneigten, rutschigen, sandigen und geneigten sandigen Oberflächen. Beim Laufen auf einer geneigten Oberfläche muss zur Vermeidung eines Sturzes die Neigungsrichtung ermittelt und die Körperhaltung des Roboters entsprechend der Neigungsrichtung reguliert werden. Beim Laufen auf nachgiebige Oberflächen, wie z.B. sandigen Oberflächen, müssen der Untergrund identifiziert und der Gang entsprechend angepasst werden. Diese Arbeit beleuchtet die eingesetzten Ansätze unserer dezentralisierten Controller, die es dem Hexapod-Roboter ermöglichen, die oben genannten Geländearten zu erkennen und seinen Gang daran anzupassen. Durch unsere Ansätze kann der Hexapod-Roboter geneigte und sandige Oberflächen erkennen und seinen Gang diesen Geländeformen anpassen. Zudem stellt diese Arbeit Versuche vor, die drei Ansätze kombinieren, um dem Hexapod-Roboter das Laufen auf einem der schwierigsten Gelände, der sandigen, geneigten Oberfläche, zu ermöglichen.

Darüber hinaus stellt diese Arbeit eine Methode zur Erkennung von rutschigen Oberflächen vor und schlägt eine Strategie zur Bewältigung dieser Herausforderung vor. Abschließend wird im Kontext dieser Arbeit die Auswirkung von Beeinträchtigungen der Beine, wie z.B. das Stoßen gegen einen Gegenstand oder ein defektes Bein, auf das stabile Laufen vorgestellt. Das Novum bei unserem Ansatz ist die Auswertung des lokalen Stromverbrauchs und der Winkelposition der Gelenke aller Beine als einziges propriozeptives Feedback. Unsere biologisch inspirierten Methoden basieren auf unserer Softwarearchitektur ORCA (Organic Robot Control Architecture) und wurden auf einer Lowcost-Version des OSCAR-Roboters (Organic Self Configuring and Adaptive Robot) getestet.

Contents

List of Figures

Nomenclature

α	The alpha joint
β	The beta joint
δ	A small change in the angular position
ΔP_{β}^{L}	The change in the β-angular position when it touches the substrate
γ	The gamma joint
D_{sand}^{L}	The leg that detects sand surface
D_{slip}^{L}	The leg that detects slippery surface
FL	The front left leg
FR	The front right leg
HL	The hind left leg
HR	The hind right leg
I	The joint current consumption
$I_S^L(t)$	The current consumption of servo S of leg L
L	A specific leg, $L \in \{FR, FL, ML, MR, HL, HR\}$
ML	The middle left leg
MR	The middle right leg
P	The joint angular position
P^F	The position of the front legs
P^H	The position of the hind legs
$P_S^L(t)$	The angular position of servo S of the leg L
S	A specific servo motor, $S \in \{\alpha, \beta, \gamma\}$
$T - down$	The threshold for downward inclination
$T - up$	The threshold for upward inclination
T_{max}	The threshold for hard ground detection
T_{min}	The threshold for sandy ground detection

CHAPTER 1

Introduction

1.1. Motivation

To enable mobile vehicles to move in their environment, they have to possess a proper locomotion mechanism. Therefore, different possible mechanisms for mobile vehicles have been developed. Different types of mobile vehicles have been produced some of which can walk, jump, run, swim, fly, and roll. Most of these locomotion mechanisms have been inspired by mobile biological counterparts either in hardware or software or both. There are also wheeled and tracked vehicles. They have been invented by humans. Wheeled vehicles have some advantages. Compared to others, they achieve an extremely high efficiency on flat hard ground. The manufacturing and control systems of wheeled vehicles are simple and provide lower-cost solutions to achieve locomotion compared to other mobile machines. They are able to move at fast speed and to carry high payloads and they have efficient energy consumption. Wheeled locomotion is very successful on predetermined flat hard surfaces. Because of these advantages, they are of great importance. But they are not proper for rough, uneven surfaces. They do not have the ability to overcome obstacles that are higher than the radius of their wheels or to cross wide holes on the surface. They lose traction in muddy and sandy or even in slippery surfaces. To solve some problems that are faced by wheeled vehicles, another type of vehicle, which is tracked vehicles, was invented. They present an important advantage for walking on loose surfaces, in unstructured environments with big gaps and steep hills and obstacles. Their palettes provide sufficient traction with the surface. They can move on sandy and muddy surfaces. Tracked vehicles have a simple mechanism and control system. They have a good payload capacity. Tracked vehicles are, however, heavy as they carry the whole palette mechanism, which causes damages of wide parts of the surface due to the traction of their pallets with the ground. Another important disadvantage is the high-energy consumption, and they are not suitable for sharp inclined grounds.

However, more than 50% of the earth's land is rough and inaccessible to traditional vehicles (wheeled and tracked)[**1**] according to [**120**] and [**98**]. Therefore, the researchers and designers were very motivated to find other mobile machines that are able to cope with rough and unstructured environments. These environments are considered very difficult for wheeled or tracked vehicles to move within. As a result of this, the legged vehicles are a solution of all the above mentioned problems.

Legged vehicles present superior mobility in natural terrains. The great advantage of these vehicles is mostly the use of discrete footholds for each foot, in contrast to wheeled vehicles, which move on continuous support surface. For walking, legged machines do not need a continuous support surface.

Many experimental walking machines with two, four, six, or eight legs have been developed since the 1950s [**24**]. This interest in legged machines aims to employ them in a broad range of applications, especially on rough terrain that is unsuitable for tracked and wheeled vehicles.

Animals and humans are a source of inspiration for legged machine designers and researchers, because biological walkers have the ability to achieve incredible adaptability while walking on uneven terrain. They reconfigure their legs in order to adapt themselves in a proper way according to the level changes. They can change their gait pattern to fit the nature of the walking substrate. Therefore, current developments in legged robots are mostly based on biological inspirations [**24**].

Moreover, legged machines have the ability to choose the best placement for their foot according to the terrain conditions causing less damage to the surface compared to the wheeled or tracked ground. Additionally, they have a good adaptation to rough surfaces, they can step over obstacles or gaps and change the speed and gait pattern.

Legged machines use multiple degrees of freedom (DoF) in each leg. The large number of degrees of freedom enable the legged machines to achieve active suspension, which does not exist in tracked or wheeled robots. This means, they have the ability to control the force distribution through the foothold points, which provides a good interaction between vehicle and ground [**56**], and also to vary the body height to enable the machine to adjust the body posture according to the type of the walking terrain. The other important benefit is about the failure tolerance during static stable walking. However, legged machines may contain several legs and for this reason, they have the ability to maintain static stability and continue their walk even if one or more legs are damaged, e.g. leg amputation or motor failure [**27, 54, 78, 11**].

Finally, it should be mentioned that legs can be used as tools to fix the vehicle body and provide a support base that helps the manipulator to move [**109, 60**] and in some cases, one or more legs are used to manipulate objects [**110, 141, 86, 135**].

Despite all the aforementioned benefits of legged vehicles, they still suffer from some difficulties such as the complexity of kinematics and dynamics of their mechanisms and they have redundant actuators, which have to be controlled in continuous coordination. Thus, they have a complicated control system. Moreover, they are difficult to build and need a large number of motors and sensors. Therefore, they have a high energy-consumption and are considered very expensive in design [**139**]. But they are still most effective on rough and unstructured terrains. Due to the great advantages of legged vehicles over wheeled and tracked vehicles, they are used in various fields such as [**139**]:

- Explore remote locations, e.g. in volcanoes [**150**], in space [**21, 79, 80**], on the bottom of the sea [**20, 19**].
- In hazardous environments, e.g. in places with nuclear radiations [**84**], in mining exploration and prospecting [**123**], in disaster areas [**84, 92, 82**], in search and rescue operations [**92**], in military operations [**31**].
- Excavation and construction [**71**].
- In forest works [**70**].
- In payload transport operations [**153**].
- Services [**129, 108**].

Since legged vehicles are designed to walk in rough environments and to response to changes of the surrounding environment, they have to possess a strong walking control system that can coordinate the movement of their legs and joints according to the nature of the faced situations. The control system has to be reliable and robust to maintain stable walking on rough terrain such as slippery, inclined surfaces or walking on sandy ground. If the legged machine research is successful, it will enable legged machines to traverse rough terrain quickly and efficiently.

1.2. Contribution

The main goal of this thesis is improving the hexapod robot locomotion on rough terrain to insure stable adaptive walking based on organic computing principles and proprioception feedback. Our biologically inspired approaches enable the hexapod robot to recognize the type of the walking surface and to adapt itself while walking on varying rough terrain such as slippery, sandy, inclined and compliant surfaces. The novelty of our approaches is the evaluation of the local current consumption and angular position of each leg's joint as proprioception feedback. Additionally, these approaches enable the robot to sense the type of walking substrate without using additional sensors, which would increase the complexity of the control system and increase the cost of the robot. Moreover, our introduced approaches could be used as alternative sensing in the event of a malfunction of the used hardware sensors. This research supports the control systems that are based on fault tolerance.

1.3. Thesis Outline

In the 2nd chapter, a general overview of the historical progress of the mobile machines will be introduced. Then, the current state of the art of the biologically inspired hexapod robot will be demonstrated. In this chapter, different prototypes of legged robots will be presented.

The 3rd chapter introduces the commonly used robot control architectures. It demonstrates the deliberative control architecture, the reactive and behavior-based architecture, and hybrid reactive-deliberative architecture. Then, it gives an overview about the biologically inspired control architecture such as a biologically inspired network (Walknet) and the software framework MCA2. This chapter also describes the autonomic computing systems and organic computing principles. Finally, the Organic Robot Control Architecture (ORCA) will be discussed in detail. ORCA is related to our research experiments.

The 4th chapter introduces the OSCAR (Organic Self Configuring and Adaptive Robot) series which is used as a demonstrator for testing the ORCA architecture. This chapter describes the hexapod robots OSCAR1, OSCAR2, OSCAR3, OSCAR-X, and the latest version of OSCAR that is used as a demonstrator of our research experiments. This chapter introduces a brief description of the biological locomotion and the types of gait patterns that are used by an insect to achieve its forward walking. Then, it shows how the OSCAR robot moves its legs. It describes the stance and swing phases and the coordination between all legs. Finally, it describes the OSCAR emergent walking based on the implemention of the biological rule that is introduced by Cruse.

The 5th chapter describes our inclination detection and adaptive walking approach for hexapod robots. This chapter shows how OSCAR detects the direction of ground inclination and avoids obstacles. Two experiments, up and downhill walking, are introduced in this chapter. Finally, the effect of leg amputation while the robot walks on inclined surfaces are discussed at the end of this chapter.

The 6th chapter introduces our applied decentralized controller approach for slippery ground detection. Then, it presents our proposed strategies to avoid walking on slippery surfaces. This chapter discusses two experiments: The first one is walking on rough and slippery terrain. The second one shows the reaction of the robot due to the slippery surface detection. The obtained results are discussed at the end of this chapter.

The 7th chapter introduces our applied decentralized controller approach for sandy ground detection and the robot adaption on sandy ground. It shows how the OSCAR robot detects sandy surfaces and then reacts to the detected ground. Two experiments are demonstrated in this chapter, the first one shows how the robot detects hard grounds and the second one shows how it detects sandy surface and achieves adaptive walking. The obtained results are discussed at the end of this chapter.

The 8th chapter demonstrates our novel approach for adaptive control of the leg position for the hexapod robot while walking on a compliant surface, such as sandy or gravel surface. This chapter discusses this approach and shows how the OSCAR robot controls its leg position to be commensurate with the type of the walking terrain. Two experiments are conducted, walking on gravel and sandy surfaces. The obtained results are presented and discussed at the end of this chapter.

The 9th chapter introduces the conducted experiment that combines the three approaches, which have been presented previously. This chapter shows and discusses the obtained results while the robot walks on an inclined sandy surface. Also, this chapter introduces the effect of leg amputation on statically stable walking while the robot walks downhill on a sandy inclined surface.

The 10th chapter gives a conclusion on the presented research experiments which are introduced in this thesis and gives future outlooks.

CHAPTER 2

A Historical Overview and State of the Art

2.1. Legged Vehicles Evolution

The idea of building machines to emulate features of humans or animals is one of the oldest ideas. Between 1495 and 1497, the universal genius Leonardo Da Vinci designed the first articulated anthropomorphic robot [**139, 125, 126**]. The entire control system of this knight robot was operated by gears and wheels that were connected to a series of pulleys and cable systems. This mechanism enabled the knight robot to stand up, to sit down, to maneuver its arms, and to move its neck and visor. Wood, leather, and brass or bronze were the constituent materials of this robot. Da Vinci's inspiration was an incentive for researchers to draw their inspiration from nature. But this idea was not implemented until the middle of the 19th century [**118**]. Around 1870, an early walking model was introduced. This model depended on a linkage to move the body forward on a straight horizontal path. It had four legs which steeped up and down to exchange the support [**118**]. The originality of the used linkage was designed by the famous Russian mathematician Chebyshev some year before [**118**]. During the 80 or 90 years that followed, all walker machines have been developed based on the linkage and have been driven by a source of power [**118**] e.g. [**101, 138**]. But the efficiency of this type of walkers was limited because they had a fixed gait pattern, they could not adapt themselves to the varying terrains such as facing obstacles, gaps, or inclined terrain. Therefore, in the late 50s of the last century, the designers noticed and realized the importance of having a control system for these machines [**90**].

One of the used approaches to control the designed walkers was a human. Ralph Mosher relied on this approach to build his four legged machine in 1968 [**91**]. The driver controls this machine through handles and pedals that are connected hydraulically to the four legs. Each handle or pedal controls one of four machine's limps. The force feedback enables the driver to sense whether one of the four legs encounters an obstacle or not. These machines were a landmark in legged robot technology despite their dependence on humans in order to control these machines.

Later, due to the development of technology, a computer control is used as alternative instead of human control. The computer control was used to control the legged machines in the 1970s. The first use of computer control was introduced by Robert McGhee's group at the Ohio University [**94**]. They built an insect-like six legged robot in 1977. This robot was able to walk with different step gait patterns. It could turn, walk sideways and overcome small obstacles. The task of the computer was to solve kinematic equations to coordinate the leg movements. This coordination ensured that the center of mass lay in the polygon of support spanned between the walkers's ground touching feet. The first introduced model was OSU (Ohio State University) Hexapod [**95, 96**]. Then it was followed by a larger hexapod known as the Adaptive Suspension Machine (ASM) [**146**]. ASM is able to carry a person while walking over rough terrain. In the early 70s, the former Soviet Union carried out some experiments in walking machines [**24**], where six-legged machines were built in the early 1980s [**46**]. In the same time, the United States introduced the six-legged robot Odetics (Functionoid). Odetics was used for inspection purposes in power plants. It has the ability to climb stairs [**127**].

Since this research is specialized in bio-inspired robots, therefore, the following section will show the state of the art of the current biologically inspired robots.

2.2. State of the Art of Biologically Inspired Robots

The living walkers with six or eight legs, e.g. insects, crabs, and spiders have the ability to walk freely on rough terrain without any effort. The excellent behavior of insects in their environment makes the engineers and researchers jealous to benefit from the biological walkers. But it is difficult to emulate the biological organisms in their entirety. So the focus lies only on the transfer of some biological principles to the walking machines. For 40 years, researchers have been developing different types of legged vehicles to be incorporated in rough terrains that are difficult for wheeled and tracked vehicles. Six- and eight- legged vehicles are considered important from the point of view of engineers due to their ability to carry out statically stable walking.

The beginning of biologically inspired robotics dates back to the last view decades of the 20th century due to the scientific development. Then, the researchers tried to realize and to implement their ideas in a technical field [**119, 45**]. The term of biologically inspired robotics appeared in Beer's title journal article for the first time [**23, 45**].

So, some researchers have focused on implementing as many biological features of animals in their robots as possible, even if the functional usefulness of the added feature was not clear [**45, 33, 47, 122**]. The purpose of adding these features is to give the robot useful attributes, even if this usefulness is not clearly visible. In addition to the use of biological principles as important source of inspiration for building robots, some researches focused on studding robots to understand the biological principles[**22, 122, 147**].

Since multi-legged animals are both anatomically and functionally complex, it is difficult to reproduce them completely in hardware and software [**45, 24**]. Therefore, the designers or robot builders usually chose one of two approaches. Some focused on building their walking machines based on kinematics and dynamics and also mechanical parts such as joints, legs, and the skeletal are similar to their living counterparts. Other robotic researchers focused on implementing the biologically inspired approaches, attempting to emulate the movement of an insect's leg and finding the optimal gait for robots [**29, 24**].

Most walking vehicle, whether quadruped or hexapod, are statically stable, whereas some others show both dynamic and static stability. The statically stable walking is related with the number of the legs of the legged vehicle. The six-legged vehicles are considered the most stable at all. Therefore, there is a great interest in building six-legged robots.

Since this work is limited to introduce adaptive walking in rough environment for hexapod robot, we will show some examples of the current researches in the same area, i.e. DLR-Crawler, LAURON, AMOS-II, ANTON, RHex and TARRY.

2.2.1. DLR-Crawler.

DLR-Crawler is a six-legged walking robot. It is designed based on the fingers of the DLR-Hand II [**69**] (see Figure 2.2.1). It was introduced in 2008 by the DLR-Germany Aerospace Center, institute of robotic and mechatronics [**63**]. DLR Crawler is a laboratory test bed to evaluate the performance of different position- and force-based gaits as well as to evaluate the control algorithm. Each leg consists of four joints and three actively controlled degrees of freedom. Each joint is actuated by a permanent magnet synchronous motor. Thus, the Crawler has a total of 18 DoF. DLR-Crawler is equipped with a rich set of proprioceptive sensors such as position sensors, joint torque sensors, and force-torque sensors at the tip of each leg. Crawler is equipped with a stereo camera. DLR-Crawler is used to implement active joint compliance, reflexes [**67**] and adaptive walking in case the robot loses one or two legs [**62**].

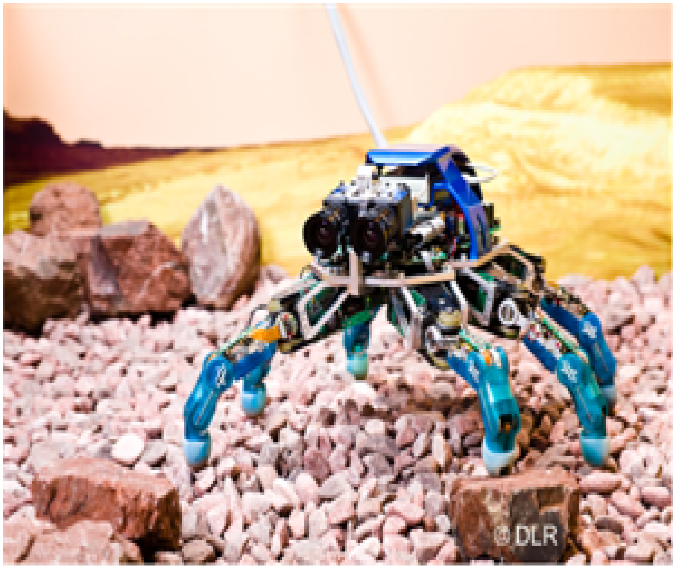


FIGURE 2.2.1. DLR-Crawler [**10**]. ©2016 German Aerospace Center (DLR), used with permission from Martin Görner.

2.2.2. LAURON.

LAURON (Legged Autonomous Robot Neural Controlled) is a biologically motivated six-legged walking machine series (see Figure 2.2.2). It is designed at FZI (Research Center for Information Technology) in Karlsruhe, Germany [**73**]. The current researches are carried out on the latest version LAURON-V [**124**]. Its construction is based on the biological model of a stick insect. Each of the six identical legs has four joints. Its head can be moved by two independent axes (roll, pitch), so that LAURON has a total 26 degrees of freedom. LAURON is equipped with a rich set of sensors. In each foot, there is a 3D force sensor and a spring-measuring system. These components in conjunction and the motor current consumption measurements are used to define the collision and ground contact. Each motor has a high resolution encoder that provides information on the joint angles. The inclination sensor gives information about the body position. To provide information about the surrounding environment of the robot, LAURON has two camera systems, a stereo camera and an IR camera fixed on the head of the robot. On the back of the robot, there is a small rotational 3D-laser scanner. The robot control software in LAURON is a hierarchical architecture. It consists of several levels that depend on the MCA2 framework [**143**]. In the first years, the research focused on statically stable walking in rugged and harsh environments. In addition to the development of the control software, the mechanic and sensors were also continuously developed. Current work focuses on bio-inspired gait analysis, navigation strategies, robot autonomy, manipulation with its front legs, and energy efficient walking [**73**].

FIGURE 2.2.2. Six-legged walking robot LAURON V[**73**]. ©FZI (Research Center for Information Technology, Karlsruhe, Germany), used with permission from the Arne Rönnau.

2.2.3. AMOS-II.

The six-legged robot AMOS-II is the latest version of the robot AMOS. It is designed and developed in the Bernstein Center for Computational Neuroscience (BCCN) in Göttingen, Germany, in collaboration with the Fraunhofer Institute for Intelligent Analysis and Information Systems IAIS, Germany [**18**] (see Figure 2.2.3). The neural network system is used to coordinate many degrees of freedom of the robot

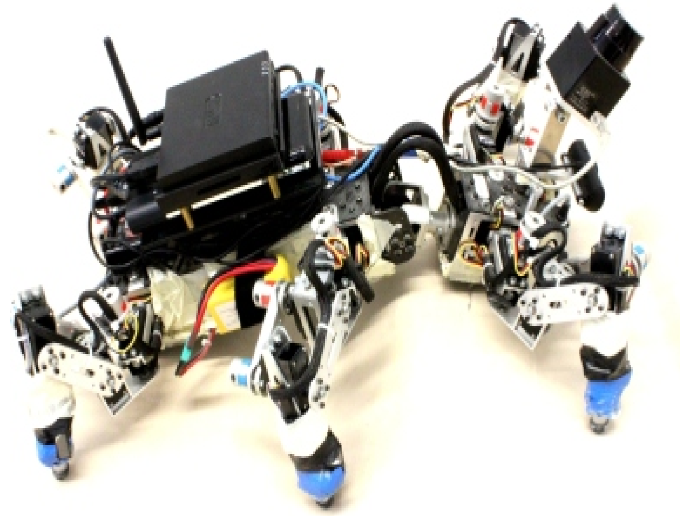


FIGURE 2.2.3. Six-legged robot AMOS-II [18]. ©2011 -2013 by Poramate Manoonpong & BCCN Goettingen, used with permission from the P. Manoonpong.

while walking in different scenarios [154]. AMOS-II is a biologically inspired machine that mimics the morphology of a cockroach. Each leg has three joints which are driven by digital servo motors. The body consists of two segments connected by a joint (backbone joint) that is driven by a digital servo motor. The backbone joint provides enough mobility to climb an obstacle or to lift the front legs that are connected with the front segment. The middle and the hind legs are attached to the second segment. This machine has six foot contact sensors, seven infrared (IR) sensors, three light dependent resistor (LDR) sensors, one inclometer sensor (IM), one laser scanner sensor (LS), one camera sensor (CM), six reflexive optical sensors, and one upside-down (UD) sensor. The neural controller of AMOS II consists of three main modules which are a Central Pattern Generator (CPG), a Phase Switching Network (PSN) and two Velocity Regulation Networks (VRNs) [154].

2.2.4. ANTON.

ANTON is an articulated six-legged walking robot (see Figure 2.2.4). It has been developed in the institute of Electrical Power Systems at Otto-von-Guericke university of Magdeburg in cooperation with the virtual engineering department of the Fraunhofer institute [85]. This robot consists of three modular segments with the scalability to add other modular segments. The modular segments are connected with each other through joints. Each modular segment is linked with two legs, one leg on the right side and the other one on the left. Each leg is

FIGURE 2.2.4. ANTON six-legged robot [74]. ©2007 RobotsLab, used with permission from the Ulrich Schmucker.

inspired by an insect's leg. The leg is actuated by three servomotors. The total degree of freedom of the robot with three modular segments is 20 DoF. The robot has a rich set of sensors that are standard for mobile robots. These sensors enable the robot to achieve autonomous walking in an uneven environment. These sensors are 24 potentiometers and IGRs (installed in each robot joint), 6 three-component force sensors (mounted in each leg's shank), one 2-axis gyroscopic sensor (located in body), and 2 mono cameras in the head of the robot. The control system of ANTON is completely located on the PC-side that receives and produces control signals for driving all actuators and sensors of the robot [**85**]. The control system of ANTON is hierarchically organized and has a modular structure [**53**]. ANTON serves as prototype for testing the intelligent and adaptive motion control system, optimizing and adapting the mechanic and electronic equipment for implementing in several legged robots.

2.2.5. RHex.

RHex is a biologically inspired hexapod robot. Many universities and disciplines contribute to the development of this robot. This project is funded by the Defense Advanced Research Projects Agency (DARPA) [**111**]. This robot was designed to emulate the kinematics and dynamics of cockroach locomotion [**9**]. The robot has six tuned compliant half-circle legs. Each leg has one degree of freedom with a single rotary actuator at the hip. RHex robot can achieve various locomotion tasks such as walking and running [**131**], pronking [**97, 83**], leaping and flipping [**132**], running upright on its rear legs [**107**], and even climbing stairs [**100**]. Several platforms based on RHex concept have been developed, i.e. Rugged RHex [**15**] built by Boston Dynamics, EduBot [**148**] developed at the University of Pennsylvania, the amphibious robot AQUA [**49**], SensoRHex [**130**] developed at Bilkent University. The latest version of the highly mobile RHex platform is X-RHex [**59**].

2.2.6. TARRY.

The six-legged robot Tarry is designed by bio robotics pioneer Holk Cruse at the University of Bielefeld [**2**]. The construction of the walking machine is based heavily on the body of a stick insect. The six legs are equipped with three joints which are positioned in a way similar to the joints of the insect's legs. The dimensions of the individual segments and the joint positions are similar to those of the biological model. The machine has more than 18 degrees of freedom. The joints of Tarry llb are equipped with a spring contracture to enable the legs to achieve passive compliance [**136**]. The robot is equipped with antennas as novel active tactile sensors.

A research priority of Tarry robot is the evaluation and verification of theories of biological cybernetics [**12**] and testing the biological hypotheses about the neural networks used by insect [**2**].

The insect-like robot platform is established for testing the walking control [**134**] and for testing the functionality of the Antennas [**87**] by insects. The results of the biological research go directly into the insights of robotics.

2.2.7. Mechanic and Control.

The mechanics and sensors by the robot DLR crawler have been originally developed of the DLR hand. The sensible sensors and the sophisticated mechanics of the fingers of the DLR hand are used for the legs of the DLR-Crawler. The research project will benefit directly from the developments that are conducted on the DLR hand [**63**]. The control of the DLR crawler is based on the effective coordination principles that are introduced by Cruse and that are also used to control Tarry IIb robot.

The robots LAURON and ANTON possess an advanced mechanical design compared to AMOS and TARRY. In ANTON and LAURON, special mechanics and sensors are tested, adjusted and optimized. The robots ANTON and LAURON use a a hierarchical control structure to control the legged machine. This hierarchical control system is not used by other legged robots that are listed here. The current research focuses on the evaluation of various sensor signals to address real scenarios, such as rescue operations and inspection operations, visions. Additionally, mission planning and mapping are examined in combination with the motion control of the six-legged mobile platform.

In contrast to ANTON and LAURON, the optimization of material and inserted mechanics in AMOS and TARRY is not the subject of the research. The control concepts of the machines Tarry IIb and AMOS are strongly based on the biological models of the stick insect and the cockroach. Both controllers use a reactive, decentralized concept based on the use of artificial neural networks. They implement the findings of experimental results of the biology on the robot controller. Both robots are used as a platform for the investigation of theories of biology and also to develop the locomotion of walking machines.

In the latest version of X-RHex, there was an intense focuse on the improvement of the mechanical and electronic systems to enable the robot to achieve highly dynamic maneuverability and sensor-based autonomous behaviors in rough environments [**59**]. The X-RHex control software architecture is based on dynamism. Dynamism is a custom-developed open source software library implemented in C [**59**].

Finally, each robot is controlled by a control system. There are different types of robot control systems. In the following chapter, we will show the different commonly used control architectures for robotics.

CHAPTER 3

Robot Control Architecture

3.1. Introduction

The engineering science is continuing to develop technical systems, the basic principles of which depend on biological principles. The structure and performance of these systems are evaluated according to technical benchmarks and not according to biological standards. In addition to the development of mechanical structures, the control architectures that control the walking mechanism play an increasingly important role. These control architectures presuppose a basic intelligence of walking machines. This intelligence allows to achieve flexible adaptation while walking on rough terrain. Due to the remarkable development in technical and mechanical sides, the control architecture is becoming more and more important.

In this chapter, we will show some of the important control architectures used in the field of robotics. Different mobile robots implement different kinds of control architectures which are divided into following groups:

- Hierarchical and Deliberative Architectures
- Reactive and Behavior-Based Architectures
- Hybrid Reactive- Deliberative Architectures

Then, some inspired control architectures that are inspired strongly by biological systems will be described such as:

- Walknet: A biologically inspired network
- MCA2: A biologically inspired framework

The term of "Robot Control Architecture" refers to the organized software of robot system that controls the entire range of control issues, from low-level to the high-level. Several definitions were issued to define the robot control architecture. Bekey [**24**] defines it as follows:

„Architecture represents the structure of the software, the way in which the robot processes sensory inputs, performs cognitive functions, and provides signals to output actuators, independently how it was designed".

Another definition given in a textbook on artificial intelligence [**128**] according to Bekey is [**24**]:

"The architecture of a robot defines how the job of generating actions from precepts is organized".

Robot control architecture defines the way in which the robot senses, recognizes and reacts. The following section describes the deliberative control architecture.

3.2. Deliberative Control Architecture

Deliberative architecture has a hierarchical structure based on the Sense-Plan-Act principles [**24, 76**]. Each layer supplies the layer below it with explicit instructions. In other hierarchical designs, a layer sometimes communicates with several layers below it in the hierarchy. Furthermore, these types of control architecture usually need to include a model of the environment in which the robot moves. In this type of control architecture, the robot firstly senses the environment, then plans possible solutions based on the included model before it achieves the action. In other words, the action depends on the planning and reasoning from the model not directly from the perception. The structure of this type of architecture is shown in Figure 3.2.1.

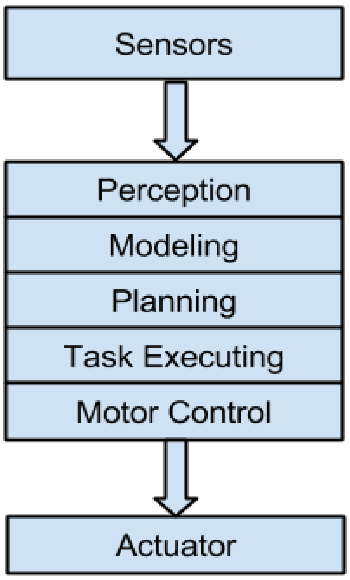


FIGURE 3.2.1. Sense-Plan-Act of deliberative control architecture [**76**].

This type of control architecture needs time to respond due to the required time of the planning and reasoning phase. Therefore, such a type of control architecture is suitable for stationary environments and not suitable for a task that needs a fast response. Another drawback appears in a changed environment. The model of the environment has to be updated in case the environment is changed to achieve a compatible reaction in the new environment. Therefore, this control architecture cannot be used in the changed environment. The following section will discuss other control architectures, which are the Reactive and Behavior-Based Architectures.

3.3. Reactive and Behavior-Based Architectures

Contrary to what was mentioned previously about Deliberative Control Architecture, Reactive and Behavior-Based Architectures are considered a close coupling between perception and action without using cognitive layers as an intermediate between sensors and action. This type of control architecture was originally proposed by Brooks [**30, 24**]. It is known as subsumption architecture. Its behaviors are horizontal decompositions and they work simultaneously, while the deliberative control architecture uses a "vertical" structure.

In this architecture, there is no higher level that supervises the other layers and also it is difficult to determine which layer has priority [**76**]. Due to the direct coupling between sense and action, the speed of response of this type of control architecture is rather high. Therefore, this architecture is may be suitable for the use in changed environments that need fast response and in case the time of reaction is very important. Current versions of this architecture describe the robot's behavior based on the rules that determine the robot's response according to a given input from sensors (e.g. in case the robot's leg hits an object, it will move back and then move up and then forward). This behavior of a robot with a direct coupling between the sensors and action appears "intelligent" [**24**]. Figure 3.3.1 shows the design of this type of architecture.

FIGURE 3.3.1. Reactive and Behavior-Based Architecture [**24**].

However, it is important here to mention that in some complex tasks, this purely reactive architecture does not perform satisfactorily.

Some cases need to reason and plan. For better results, the two above-mentioned control architectures are combined to form so-called hybrid architectures which will be discussed in the following section.

3.4. Hybrid Reactive-Deliberative Architecture

This architecture is considered as a combination between deliberative and reactive architectures. It is used to mitigate the drawback and limitation seen by deliberative and reactive control architectures. A typical structure of hybrid architecture consists of three layers as shown in Figure 3.4.1. The upper layer is the deliberative layer including the interface with the human being and planning and localization. The lower layer is the reactive layer, in which sensors and actuators are directly coupled. The middle layer in this architecture is the intermediate layer or the so-called supervisory layer [**24**].

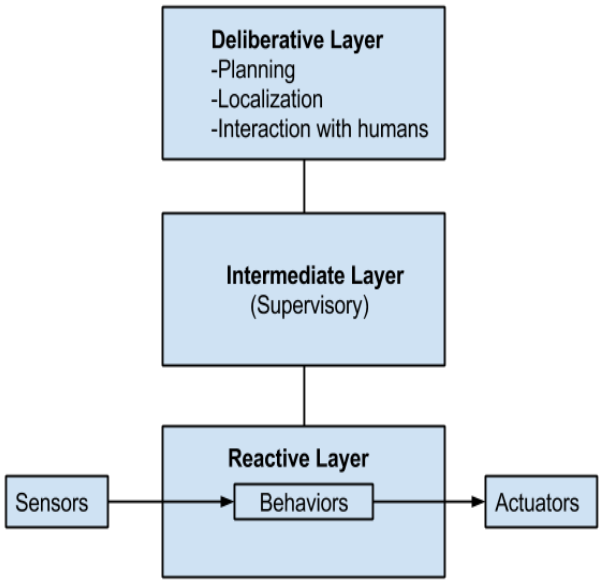


FIGURE 3.4.1. Hybrid Reactive –Deliberative Architecture [**24**].

The upper layer has more time for planning and reasoning based on the good modeling of the world. The bottom layer reacts rapidly to the quickly changing environments. The benefit of using such a type of architecture is the combination of the advantages of the deliberative and the reactive control architectures. Additionally, it mitigates the drawbacks arising due to using each one separately.

In addition to the above mentioned designed control architectures, there is a trend to benefit from biology to design control architectures similar to the one used by biological walker.

In the early fifties of the last century, researchers started the systematic analysis of biological control systems. The first work was introduced by Norbert Wiener [**151**]. He was the father of cybernetics and was among the first to model control phenomena in both animals and machines [**24**]. The structure and features of typical biological systems differ significantly from engineering systems. Some characteristics of biological control systems are described in Table 1.

Adapts itself to changes in external and internal environment
Generally nonlinear
Hierarchically organized
Control is frequently distributed
Contains multiple control loops
May display limit cycle oscillations
Includes redundancy

Table 1. Some characteristics of a biological control system [**24**].

In this work, two biologically inspired control systems will be discussed, the Walknet architecture and the MCA2 architecture. The following section shows the Walknet architecture that is strongly inspired by insect leg controllers.

3.5. Walknet- A Biologically Inspired Network

"Walknet" is a software framework to control a kinematic model of a stick insect (Carausius morosus) based on artificial neural networks. It was developed in the department of Biological Cybernetics at the University of Bielefeld and serves in evaluations of biological hypotheses for insect walking. Walknet is designed based on the behavioral rules that were extracted by conducting experimental studies on the stick insect and based on the study of the neural connections of the insect nervous systems [**38**]. The main goal of Walknet is to control the simulated locust and study the behavior of insect walking. The simulated stick insect has six legs. Each leg has three joints. These joints are denoted by α, β, and γ. The joint α is closest to the body and γ is farthest. Walknet is a decentralized controller, which consists of various modules. Each leg of the simulated stick insect is controlled by its own leg controller. Figure 3.5.2 shows the schematic diagram of a leg controller. The leg controller consists of three modules. Each module is implemented by an artificial neural network. These modules are [**38**]: The swing net, the stance net, and the selector net. The selector net determines whether the stance or the swing net has to be activated. In addition to the three major networks of the leg controller, there is the target net, height net, PEP (Posterior Extreme Position) net, and coordinating influences.

3.5.1. The Selector Network.

This network is responsible for choosing the network that will move the leg. This net selects either swing or stance net. The ground contact and the leg position as well as the local coordination rules determine the output of this network.

3.5.2. The Swing Network.

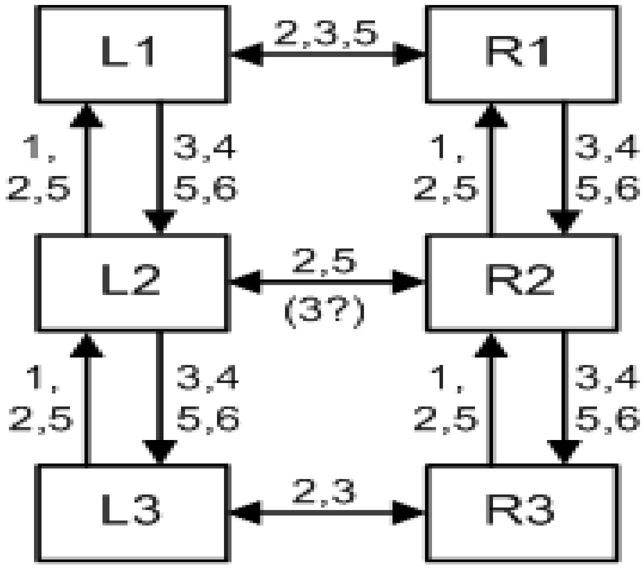
The responsibility of this network is the execution of the swing phase. In this phase, the leg is mechanically uncoupled from the environment. The swing net is a two-layer feed-forward artificial neural network. It has three output neurons and seven input neurons. Three of these seven input neurons are for the current angular position of the leg's α, β, and γ joints, and the other three inputs represent the target of the angular position, which is set by the target network. The three output neurons represent the angular velocities of the joints $\frac{d\alpha}{dt}$, $\frac{d\beta}{dt}$, and $\frac{d\gamma}{dt}$. As shown in Figure 3.5.2, the swing net is extended by four input neurons (r1, r2, r3, and r4) in case there are some disturbances such as hitting an obstacle. Each of the four neurons, r1, r2, r3, and r4, is connected to a physical sensor for detecting the leg contact.

3.5.3. The Stance Network.

This network is responsible for moving the legs that are in contact with the ground and carry the simulated insect's body. The angular velocity of α- and γ-joints is controlled by the stance network. The β-joint is controlled by an individual net (height network). The β-joint determines the distance between the insect's body and the walking substrate. The value for the β-joint is provided by a height net that consists of three feed-forward layers with three input units (α ,β, and γ) and one output unit. In the stance phase, when one joint of the stance leg moves actively, all other joints will move passively due to the mechanical interaction. The stance network controls the straight walking and negotiates curves and also determines the speed of the insect walking.

3.5.4. Coordination Between Legs.

The coupling between legs is defined by a set of six coordination rules which have been found by Cruse based on behavioral experiments on stick insects [**37**]. These rules coordinate the legs' movement during the insect walking. These rules are summarized at the bottom of Figure 3.5.1 [**38**]. Each rule is active during a certain part of the step cycle. Each rule can be considered as an information channel, through



Rule action	Goal
1 suppress lift-off	avoid static instability
2 facilitate early protraction	favour temporal coherence
3 enforce late protraction	maintain temporal coherence
4 aim touch-down location	exploit prior foothold
5 distribute propulsive force	share load efficiently
6 enforce correction step	avoid stumbling

FIGURE 3.5.1. The leg coordination rules for the six legs. The left legs are denoted by (L1, L2, and L3). The right legs are denoted by (R1, R2, and R3). The coordination rules are numbered according to the list that is mentioned in the table of rules. The arrows between the legs indicate the direction in which affects the rule. The rule that is marked with a question mark indicates that the neural connections are not yet clarified in detail [**38**].

which a leg communicates with its current ipsilateral and contralateral neighbors. The front and hind legs exchange their current state with two neighboring legs, while the middle legs send information to the front, rear, and middle legs [**48, 38**] as shown in the Figure 3.5.1 above, where each leg is represented by a box.

Since Walknet is an entirely kinematic simulation, it uses only the first four rules 1 to 4 in the present model. The other rules such as rule 5 and rule 6 are neglected. Rule 5 includes the effect of load, which means that Walknet does not monitor the loading of the legs during insect walking. The rule 6 serves to correct errors in leg placement. Walknet controls only three joints per leg because the tarsus is ignored. Therefore, Rule 6 is not implemented [**38**]. Several researches have used the Walknet model and its expansion to control the hexapod walking and simulate the insect walking [**48, 39, 133**]. In 2008, the functionality of Walknet was first implemented on the real robot platform Tarry [**12, 134**]

.

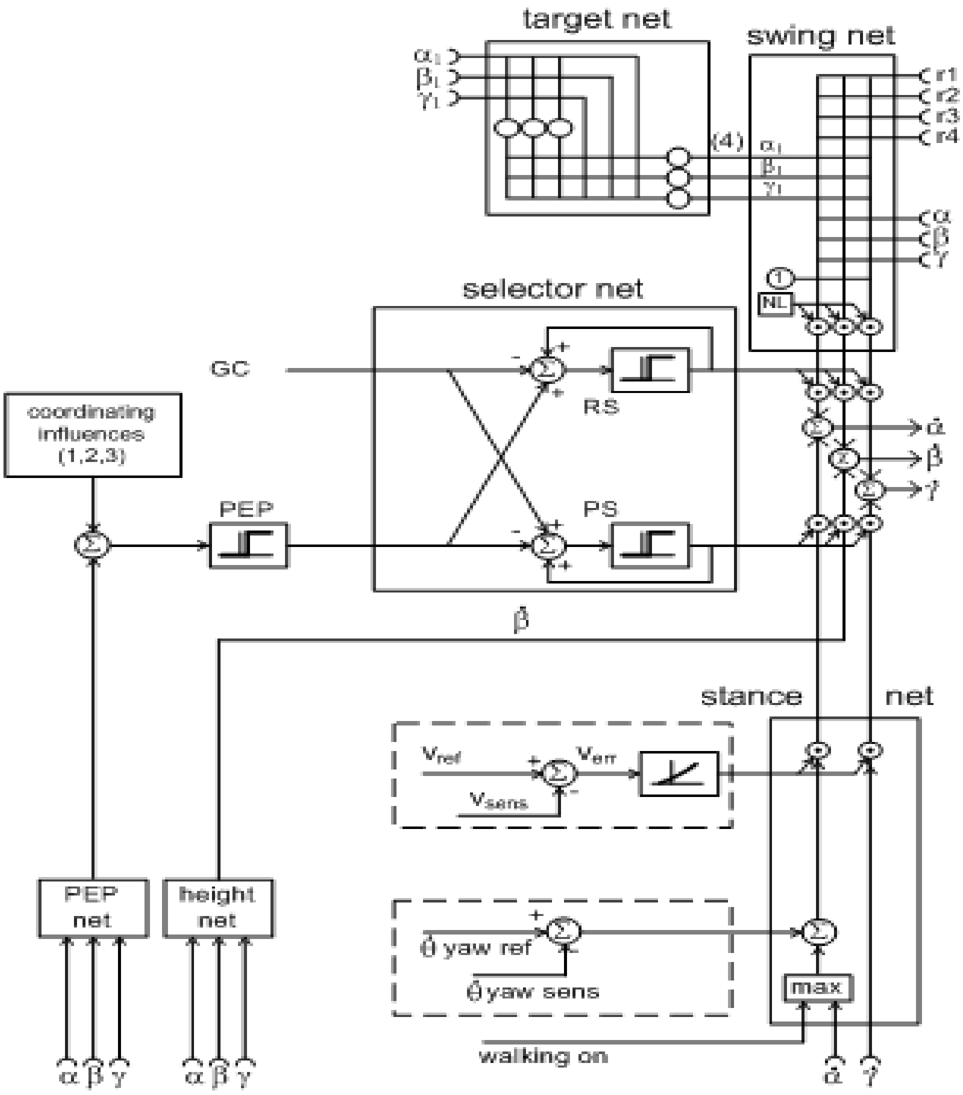
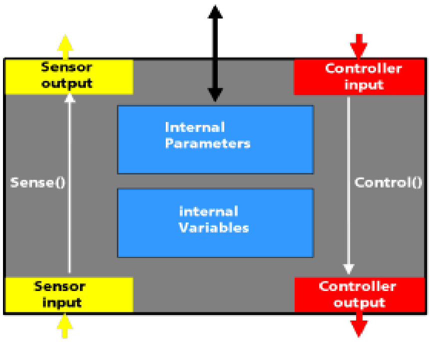


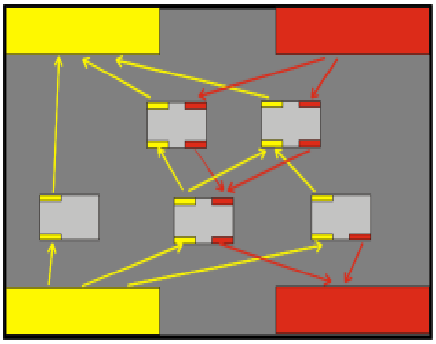
FIGURE 3.5.2. The Walknet leg controller consists of several modules: the swing and stance net, which moves the leg in the swing and the stance phase. The target net determines the next position goal. The selector net selects either swing net or stance net. The height net controls the distance between the insect's body and the ground. [**40**].

3.6. The Software Framework MCA2

MCA2 (Modular Controller Architecture 2) is a modular software framework (MCA2) that is developed for mobile robotics and other kinds of hardware at the Research Center for Informatics in Karlsruhe, Germany [**137, 143**]. MCA2 has the ability to execute real-time requirements. The Framework depends on C++ as the program language. The first version was MCA. It is designed for a special hardware architecture. The second version MCA2 is not a software architecture, and it is no longer coupled with MCA. The architecture name comes from the first version [**112**]. In MCA2, all functionalities could be achieved by simple modules that are connected with each other. Each module has the same standard structure. It has five communication interfaces: sensor input, sensor output, controller input, and controller output. The fifth interface is used to read and to modify the internal parameters of a module while the software is executed. These interfaces are arranged either above or below the module, what allows for the receipt and sending of data in two directions. In addition to the internal parameters and internal variables, a module has two routines, which are a sense routine and a control routine. The sense routine is responsible for processing sensor data. The control routine is responsible for handling control data. The modules can be grouped together to form a groupe module. The groupe module is a module with a complex functionality. The internal modules that form the groupe module are organized hierarchically. Within the groupe module, the modules are executed iteratively level by level. The sense routines are run from the bottom to the top level and the control routines are run from the top to the bottom level. Figure 3.6.1 (a) shows the schematic structure of a single module of MCA2 and Figure 3.6.1 (b) shows a groupe of the modules which form the groupe module.



(A) Basic structure of the module



(B) Hierarchically organized group module

FIGURE 3.6.1. (a) Schematic structure of the module MCA2 (left). The outline of a group of modules together in a group module (right) [**112**].

The edge connection between the modules guarantees the transfer of values between the input and the output interfaces. This framework, can achieve more complex behaviors by combining modules and groupes. MCA2 also provides some modules that can be used by developers immediately without testing and implementing, e.g. sensor value filters, standard controllers, path planers, and image processing. These modules can be transferred easily from one robot to another. The latest version of MCA2 is the MCA2-KL. It is developed in the Robotic Research Lab at the University of Kaiserslautern in cooperation with the original authors at Karlsruhe University [**14**]. MCA2-KL operates only on Linux and there is no support for Microsoft Windows. MCAGUI, MCA Browser, and JMCAGUI [**14**] are the graphical user interfaces that corresponded to MCA2.

Several biologically inspired robots use the MCA2 framework. The hierarchical control software of KAIRO II (modular snake-like robot) is implemented in the robot framework MCA2 [**73**]. LAURON, the six-legged robot, uses MCA2 for implementing the behavior-based control and the hardware abstraction layer [**73**]. Other researches are using MCA2-KL e.g. RAVON (Testbed for Motion Control, Localization, and Navigation in rough and vegetated terrain), MARVIN (Testbed for Motion Control), CROMSCI (Climbing robot for inspection tasks of concrete walls), and ARTOS (Indoor robot for transport and service tasks) [**112**].

Our work is closely related to the concepts of the reactive controller Walknet and is implemented in a modular control architecture that is strongly oriented towards the MCA-2 software framework. In the following section, our control architecture will be discussed in detail. But before proceeding to explain our control architecture, we will provide a simple overview of Autonomic Computing systems and Organic Computing.

3.7. Autonomic Computing Systems

Technical systems are becoming more and more complicated. For this reason, it becomes necessary to create autonomic systems that have the ability to manage themselves without human intervention. The autonomic concepts are inspired by the human autonomic nervous systems that take care of unconscious reflexes. These concepts are used to build autonomic computing systems that cope with large-scale complex technical systems [**75**].

In 2001, IBM's senior vice president of research, Paul Horn, suggested the first idea of Autonomic Computing [**81, 72**]. According to Paul Horn's definition, the Autonomic Computing System is a self-management system with several characteristics such as self-configuring, self-healing, self-optimization, self-protection, and self-awareness [**105**]. By adding these characteristics to complex technical systems, they become able to achieve some tasks independently and they reduce the workload on the system administrators [**75**].

The IBM autonomic systems consist of interactive collections of autonomic elements. Each autonomic element manages its internal behavior and its relationship with other autonomic elements based on the predefined policies established by human operators or other elements. Figure 3.7.1 shows the structure of an autonomic element. The autonomic element consists of one or more managed elements. This managed element is coupled with the autonomic manager that controls and monitors it. The managed element can be hardware resources like storage, a CPU, or software resources like a data base or a large legacy system. The autonomic manager monitors the managed element and its external environment and executes the constructed plan based on the analyzed information and its knowledge. On the other hand, the autonomic element can communicate and contribute other autonomic elements to achieve its goals [**81**].

The fully autonomic computing systems with the property of self-organization may lead sometimes to unanticipated emergent behavior or undesired results. The emergent behavior can be positive or negative. Therefore, other initiatives were not interested in building their computing system in fully autonomic systems such as the the German priority program on Organic Computing (OC) [**34**] that will be discussed in more detail in the following section .

3.8. Organic Computing (OC)

The first vision of OC has been discussed in the position paper of the section of computer engineering ("Technische Informatik") of the

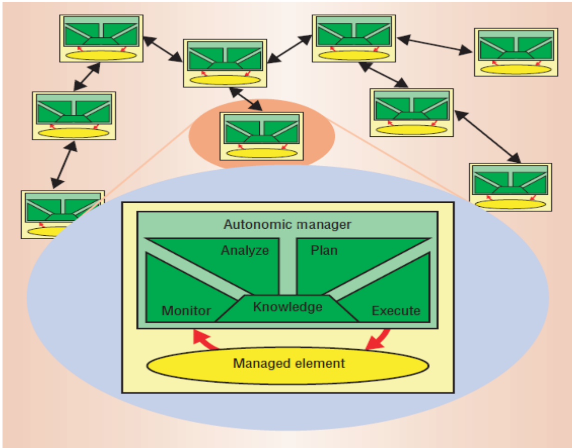


FIGURE 3.7.1. Structure of an autonomic element [**81**].

"Gesellschaft für Informatik" (GI) [**26**]. In July 2005, the special priority program Organic Computing was started by the German research foundation DFG (Deutsche Forschung Gemeinschaft) [**26**]. The objective of this priority program was to design specific concepts and tools that enable the developed systems to exhibit self-x properties such as self-adapting, self-reconfiguring, self-protecting, and self-organizing. These self-x properties enable the developed system to adapt itself either to the environmental condition changes or to the changes in the external goals and constraints, and at the same time to ensure that the self-organization does not produce undesired behavior [**99**]. The Organic Computing principles are designed based on a deeper understanding of emergent global behavior in self-organizing systems [**26**]. The OC system is very similar to Autonomic Computing systems, but it focuses so far on controlled self-organization. It does not allow to emerge unwanted behavior. The systems are based on human requirements, and are adaptive and flexible as well as trustworthy. The name "Organic Computing" comes from the presented self-x properties, which generally characterize organic structures. Organic materials for computing are not used in Organic Computing.

For the design and analysis of OC systems, a generic architectural concept has been developed within the German Priority Program on OC. Figure 3.8.1 shows the generic Observer/Controller architecture [**104, 99**]. In this architecture, the self-organized system under observation and control (SuOC) has a higher level of governance. This level consists of an observer and a controller. The observer monitors the underlying system. It sends a report about the system's state and properties of the different components to the controller. The controller evaluates the obtained data from the observer and decides whether it is necessary to take appropriate control actions to influence the underlying system. This architecture guarantees the fulfilment of the aimed

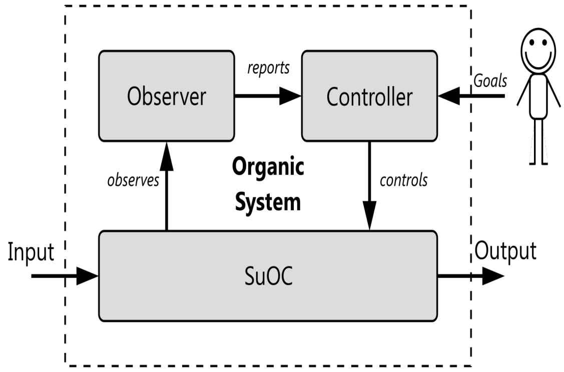


FIGURE 3.8.1. Generic Observer/Controller Architecture [99].

goal and keeps the behavior of the underlying system within the aimed constraints that are predefined externally by the user or developer. The controller intervenes in case the system behavior is undesired. Otherwise, the SuOC runs autonomously while it behaves well and achieves its goal.

Organic Computing principles have been implemented on our Organic Robot Control Architecture (ORCA) and tested on our hexapod robot OSCAR (Organic Self-Configuring and Adaptive Robot). Whereas, this work will focus on implementing these OC principles for achieving adaptive walking on rough terrain based on Organic Computing concepts and proprioceptive sensing. The controlled self-organization and other properties such as self-reconfiguration and self-adaptation are implemented while the robot walks on rough terrain. The following section presents the basic characteristics of ORCA architecture.

3.9. Organic Robot Control Architecture (ORCA)

The Organic Robot Control Architecture (ORCA) is a project of the Institute of Computer Engineering at the University of Luebeck in cooperation with the Institute of Computer Engineering at the University of Osnabrueck and the Fraunhofer Institute for Autonomous Intelligent Systems (AIS) [**12**]. ORCA was funded in the framework of a Special Priority Program (SPP) by the German research foundation DFG [**27, 77, 103, 102**]. As mentioned before, these OC systems exhibit so-called self-x properties such as self-adapting, self-configuring, self-protecting, self-organizing, and self-healing. ORCA is therefore built to satisfy these criteria [**76**].

Our work is a part of the ORCA project which examines the applicability of this control architecture on six-legged walking machines while walking on rough terrain. ORCA is a modular and hierarchically organized software framework, which enables the design of biologically inspired systems [**27**]. ORCA's architecture consists of two main units, the BCU (Basic Control Unit) and the OCU (Organic Control Unit). The modules BCU and OCU communicate and interact with each other via a unified interface. The various BCUs interact with each other by exchanging the data upwards as well as downwards in the hierarchy. BCUs form the backbone of a system and provide sufficient functionalities for the system's operation under normal conditions. Each BCU encapsulates a specific functionality and achieves a predefined task. For the robotic system, the BCU can implement different functionalities, for example PID-controllers for servo motors, or read data form sensors by encapsulating perception and proprioception. The BCUs can also encapsulate e.g. reflexes like obstacle avoidance.

OCUs are supplemented units to the BCUs. They monitor and observe the BCUs' and other OCUs' signals. The OCUs exchange data or influence the other BCUs and OCUs via the same unified interface. In contrast to the BCU, no functions are implemented in the OCU for the robot controller. The OCU is used to monitor the BCUs and other OCUs units. The monitoring functionality of the OCUs is inspired by the biological immune system. In biological immune systems, the system reacts as soon as it detects unwanted signals or anomalies. The same thing is implemented in the OCUs. They monitor the signals generated by one or more BCUs. The OCUs detect anomalies when for example the monitored signals of BCUs deviate from their normal range. The normal or "good" range is combined in the OCUs. As soon as the generated BCU signals deviate from this range, the OCU reacts by changing some parameters of the BCU to enforce the system to return to its normal or "healthy" state. This procedure ensures that the system works correctly, even if it faces a malfunction or detects unwanted signals that lead to unwanted behavior. From this respective, our ORCA is a controlled self-organized architecture. It does not allow the emergence of the unwanted behavior. Figure 3.9.1 shows the ORCA architecture.

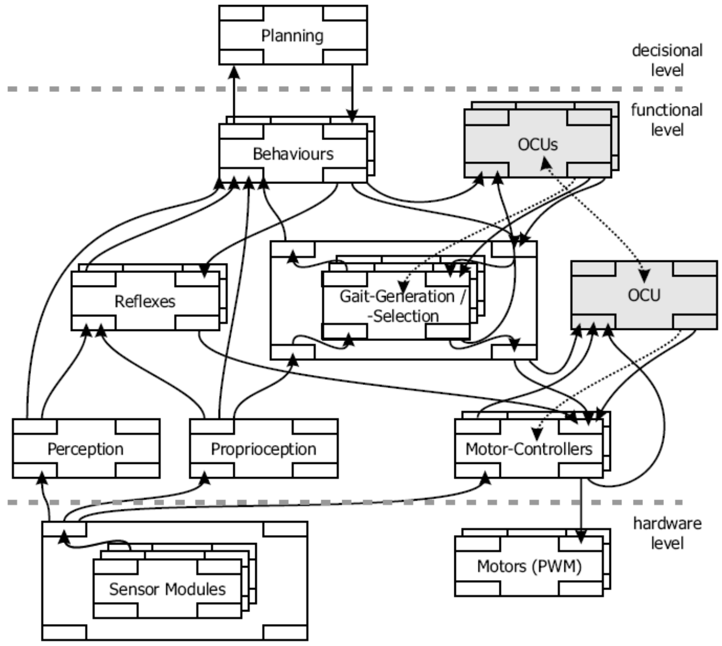


FIGURE 3.9.1. ORCA- Organic Robot Control Architecture [**27**].

3.10. Distributed ORCA Architecture to Control Hexapod Robots

ORCA has been adapted to suit our hexapod robot OSCAR (Organic Self-Configuring and Adaptive Robot), which will be discussed in the next chapter. Figure 3.10.1 shows an exemplary part of the distributed leg control according to the ORCA architecture. The required functionalities are implemented in a modular and hierarchical way by various BCUs. OCUs are used to monitor the normal state of the system. They react as soon as the system shows anomalies. As shown in Figure 3.10.1, there is one BCU for each joint of a leg that has three joints. These BCUs are responsible for controlling the respective servos. At the same level, there are also the perception and proprioception BCUs, which are responsible for collecting data from the servo motors e.g. current consumption, angular position, and current speed or from sensors such as the ground contact sensor. In the next level, there are the swing and stance BCUs that are responsible for moving the leg based on the received signals from the perception or proprioception BCUs. Each leg BCU has the same structure. The leg's BCU communicates with its neighboring legs' BCUs (left and right leg's BCUs) to guaranty a coordination between all legs. Additionally, each leg's BCU has the ability to extend its communication with the two front and two hind legs, e.g. in case of leg amputation. Each leg has one OCU unit that is responsible for monitoring the health status of the leg's BCUs.

It can react in case of malfunction such as leg amputation or undesired signals[**93**].

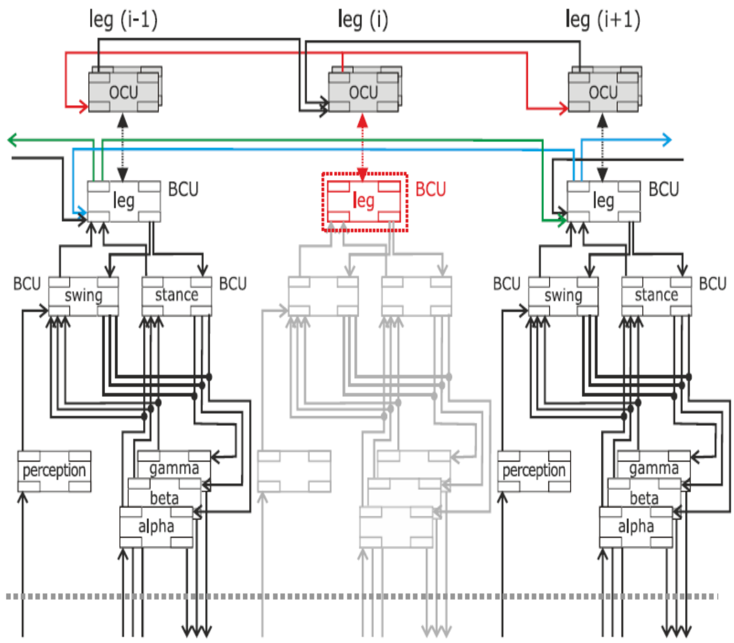


FIGURE 3.10.1. Distributed leg control with ORCA. Leg(i) has been amputated and bypassed [**93**].

Several biological approaches have been implemented on ORCA and tested on our hexapod robot OSCAR. The OSCAR robot is our hexapod robot demonstrator that will be discussed in the following chapter.

CHAPTER 4

The Six-Legged Robot Demonstrator

4.1. Six-Legged Robot Platform OSCAR

The six-legged walking machine OSCAR (Organic Self Configuring and Adaptive Robot) is used as platform for testing our ORCA architecture. For OSCAR a series of versions has been developed at our institute. This series includes the following six-legged robot demonstrators: OSCAR-1, OSCAR-2 OSCAR-3, OSCAR-X, and the low-cost version of OSCAR. The last one represents the latest version of OSCAR and the demonstrator of our work. These platforms will be described briefly in the following subsections. The robot OSCAR has at least 18 DoF (Degrees of Freedom). All components of the robot are commercially available.

4.1.1. OSCAR-1.

OSCAR-1 represents the first version of the series of OSCAR robots. It was built in the year 2006 [**102**]. Its skeleton parts and servo motors are commercially available at lynxmotion.com. The robot hardware is based on the AH3-R robot walker [**16**]. OSCAR has a symmetric shape and its six legs are assembled in a 60 degree distance around its round body as shown in Figure 4.1.1. OSCAR's body diameter is 27.5 cm and its span length is 74.5 cm when all legs are spanned. The range of the width while walking varies between 30 and 40 cm depending on OSCAR's leg posture [**54**]. Each leg is supplied with a binary switch for detecting ground contact. Due to the symmetric leg distribution, there is no obvious front and back. Thus, OSCAR is able to walk omnidirectionally (see Figure 4.1.1). The robot uses 18 HS-645 analoge servo motors. Each leg has three servo motors. Three legs of the robot's legs are equipped with an ultrasonic sensor for measuring the distance between the robot's leg and the faced object. The on-board control hardware consists of a JControl (a Java based embedded system for robot control) and a SD21 Servo Driver Module. These parts are also commercially available and are connected by an I2C-Bus.

4.1.2. OSCAR-2.

OSCAR-2 (Figure 4.1.2) represents the second prototype in the series of OSCAR robots. It is similar to OSCAR-1 but some modifications have been added to OSCAR-2. This robot has pressure sensors instead

FIGURE 4.1.1. The hexapod robot OSCAR-1 [**77**].

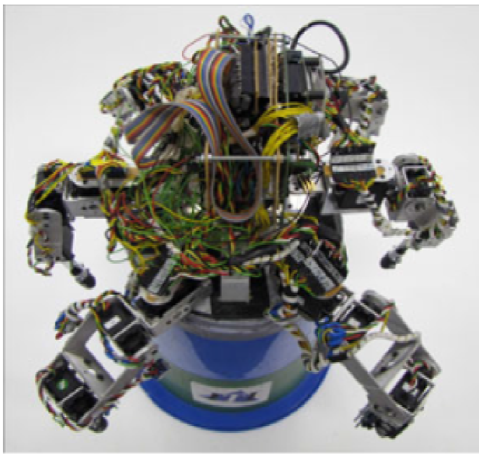


FIGURE 4.1.2. Hexapod robot OSCAR-2 [**76**].

of binary sensors attached to the robot's feet. Due to this modification, the robot can maintain good contact with the ground and sense a variable pressure on the robot's feet. Another modification has been made on the HS-645 servo motors. This modification enables the robot system to measure the current consumption and real positions of each servo motor [**102**]. The modification is clearly visible, as shown in Figure 4.1.2, as a result of the large number of wires that emerge from the servo motors. These wires are used to measure some of servo motor's parameters. Additionally, OSCAR-2 is equipped with an accelerometer sensor that is used to measure the robot's acceleration and its inclination [**76**].

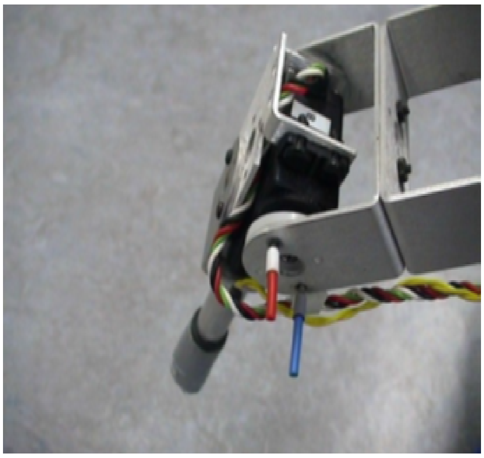


FIGURE 4.1.3. The red and blue pins will drop after a few steps [**76**].

Another modification has been conducted (on the robot leg) to simulate a faulty situation of the robot's legs as shown in Figure 4.1.3. The inserted pins drop down after some steps. This experiment has been conducted to show that the robot has the ability to detect malfunctions and to react autonomously [**76**].

4.1.3. OSCAR-3.

OSCAR-3 (Figure 4.1.4 left) has the same structure as OSCAR-1 and OSCAR-2. But OSCAR-3 uses the HiTech servo motors HS-985MG. The original boards of all HiTech servo motors have been replaced with Open Servo [**17**] boards. With this modification, the servo becomes suitable for the requirements of our work. It enables to measure the consumed current, accurate position, and current speed. The measured values are used to detect whether the leg touches the ground, hits an obstacle, or moves in the normal swing phase.

Furthermore, OSCAR-3 is equipped with a wireless camera, ultrasonic sensor, and a heat sensor. The wireless camera is responsible for sending pictures to a connected personal computer. The ultrasonic sensor is used for measuring the distances between OSCAR and an obstacle, the heat sensor is used to detect the heat source. All servo motors are connected to a Diolan board by an I^2C bus. This Board is used as a bridge to ensure the communication between the servo motors and the controller. The used Java controller in the previous version reached its limit and is not able to implement more complex algorithms. Therefore, it has been replaced by a more powerful controller. The used controller is a personal computer (eee PC). This PC controls and drives OSCAR by using a java program. It processes all received signals from the sensors and then sends a suitable command

FIGURE 4.1.4. The robot platform OSCAR-3 [11].

to achieve a stable gait. Using a PC as a controller makes OSCAR more scalable to adjust its program and add modifications.

4.1.4. OSCAR-X.

OSCAR-X was rebuilt from scratch. It is designed to provide a better robot demonstrator for testing the biologically inspired approaches [78]. This robot has several features [93].

- OSCAR-X's diameter including the spanned legs is about 110 cm, the weight of the robot including the batteries is about 7.5 kg.
- The robot is equipped with a mechanism to detach its legs in case of a leg's malfunction.
- The leg's joints are actuated by digital RX-64 servos with digital feedback for torque, current consumption, temperature, etc.
- The robot's foot is desiged for reliable ground detection based on binary contact sensors.
- The robot is equipped with a powerful controller based on Embedded Linux or Windows.
- The robot is equipped with orientation sensor (3D compass, accelerometer, gyro)
- It has expandable design with interfaces for additional sensors.

Figure 4.1.5 shows OSCAR-X. It has six legs. Each leg has three joints. It is obviously larger than the last three OSCAR generations.

The hallmark of this robot is the designed legs which can be detached from the body of the robot in case of a faulty leg [76]. The leg amputation mechanism is controlled by software. This procedure obscures any negative influence on the robot walking due to carrying a defective leg while the robot achieves its task [93, 76].

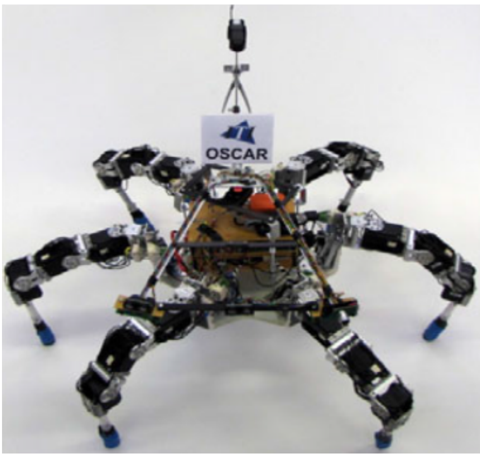


FIGURE 4.1.5. The walking robot demonstrator OSCAR-X [**76**].

4.1.5. The Low-Cost OSCAR.

ORCA has been previously applied to several versions of our hexapod robot OSCAR [**93**]. In this work, our experiments will be conducted on the latest version that is based on the Bioloid robot kit[1]. This robot can achieve different complex tasks like walking on sandy, gravelly ground, slippery, inclined ground and controlling the leg position. It is shown in the top of Figure 4.1.6 . The new version has six roughly symmetrically distributed legs which are divided into front, middle, and hind legs as shown in the bottom of Figure 4.1.6. Each leg has three joints, which will, as in the previous versions, be called alpha (α), beta (β), and gamma (γ), with α being closest to the robot's body, γ furthest away from the body, and β the middle joint. The foot is located at the tip of the leg with a hemispherical shape made of rubber. It is about 1.5 cm in diameter. Each joint is actuated by a Dynamixel AX-12 digital servo motor, which makes it possible to read, among other parameters, consumed current, the actual servo position, speed, and other parameters. Additionally, each leg is equipped with a binary switch for detecting ground contact as an additional option, but in this work the contact with ground is detected based on analyzing the current consumption in the middle joint (β-joint). The robot is equipped with an inclination sensor as a reference to our adjustment of the body posture. All robot components use two batteries (NiMH3900) as power supply. The servo motors are connected via TTL level half duplex UART. The control software runs either on a personal computer or on an ASUS eeePC that can be carried. The servo motors are controlled by PC via USB2Dynamixel as shown in Figure 4.1.7.

[1]http://www.robotis.com/xe/bioloid_en

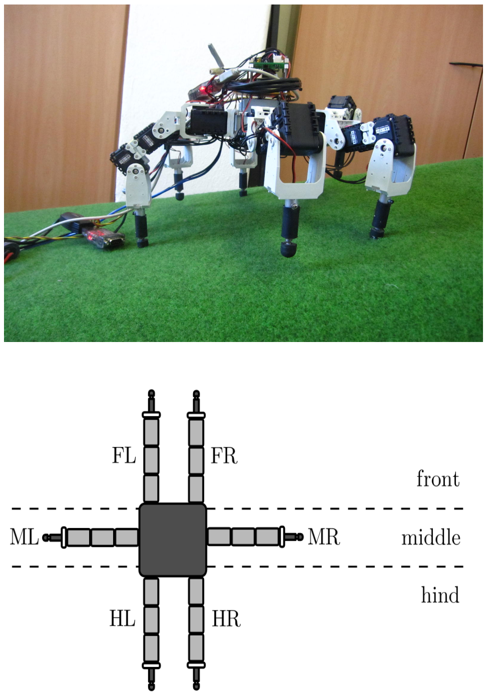


FIGURE 4.1.6. Top: The test platform OSCAR based on Bioloid kit. Bottom: the robot geometry.

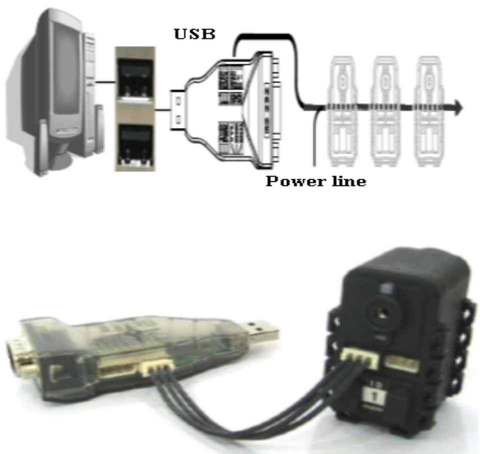


FIGURE 4.1.7. The servo motors are connected to the controller via USB2Dynamixel [**144**].

4.2. OSCAR's Leg

OSCAR's leg is modeled based on a typical insect leg. It is necessary to understand the structure of the insect's leg as well as how the insect

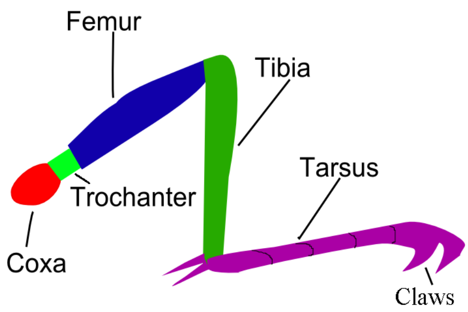


FIGURE 4.2.1. Schematic view of an insect leg. Segments: Coxa, Trochanter, Femur, and Tibia. Foot consists of Tarsus's segments and Claws [**145**].

moves its leg, and also which components cooperate to achieve this task.

4.2.1. Insect's Leg.

The legs of all insects have the same basic parts: coxa, trochanter, femur, tibia, and tarsus. Figure 4.2.1 shows a schematic view of an insect leg.

The basic insect leg's parts can differ considerably [**145**].

- The Coxa: is the basal segment of the insect's leg. It is a ball joint that connects the leg with the body. It enables the insect's leg to move forward and backward.
- The Trochanter: is a leg's segment between 'Coxa and 'Femur'. It is usually small and serves as a joint between the 'Coxa' and the 'Femur' and it enables the insect's leg to move up and down.
- The Femur: this segment is used for running and jumping. It is longer and thicker than the other segments.
- The Tibia: is a leg segment between Femur and Tarsus. Between Femur and Tibia, there is a joint. It helps in the extension and flexion of the insect's leg.
- The Tarsus: is the foot of the insect leg and consists of one to five segments.
- The Claws: are located at the end of the Tarsus. They are used to assist the insect in holding onto the substrate.

4.2.2. Construction of OSCAR's Leg.

OSCAR's leg (Figure 4.2.2) consists of three joints and each joint presents one DoF:

- Alpha joint (α): It is the closest to the robot's body. It is equivalent to the Thoraco-Coxal joint in the insect. This joint enables the leg to move forward and backward.

- Beta joint (β): It is the middle joint. It is equivalent to the Coxa-Trochanteral joint in the insect. It enables the leg to lift up and down.
- Gamma joint (γ): It is the furthest joint. It is equivalent to the Femur-Tibia joint in the insect. This joint enables the insect's leg to make the extension and flexion.

In OSCAR's leg, the tarsus part is ignored. Through the mentioned joints above, OSCAR's leg has the ability to move freely in three dimensions.

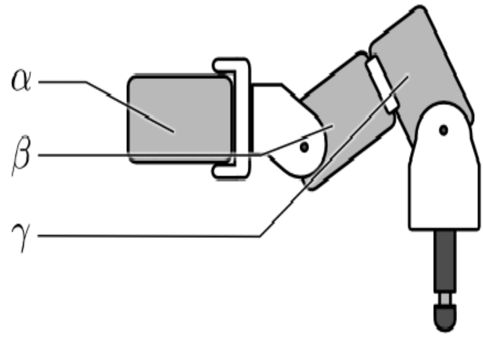


FIGURE 4.2.2. Schematic structure of OSCAR's leg with α-, β- and γ-joints

For a better understanding of OSCAR's leg motion and the robot walking, it is very important to understand some biological locomotion and the types of gait pattern that is used by insects. The following section shows briefly the general concepts of locomotion by legged walkers.

4.3. Concept of Locomotion

At first glance, the insects' walking seems to be very simple and effortless. In fact the locomotion is really complex. The locomotion needs accurate coordination between the insect's different muscle groups that enable it to walk correctly and in a stable way. Therefore, the synchronization of acting between the legs is necessary to achieve the coordination between the six legs. Thus, an interaction between the contralateral and ipsilateral legs of stick insects is used during walking [**41**]. These interactions between insect legs ensure robust and adaptive walking on flat and rough terrains.

In robotic applications, adaptive walking on rough terrain for a legged robot with six or more legs requires also a good coordination between all legs of the robot. E.g. a six-legged robot has at least 18 joints, three per leg. Additionally, in some cases, the body has some joints. Therefore, the robot has to control more than 18 degrees of freedom (DoF). Thus, the important question is: How will these DoF

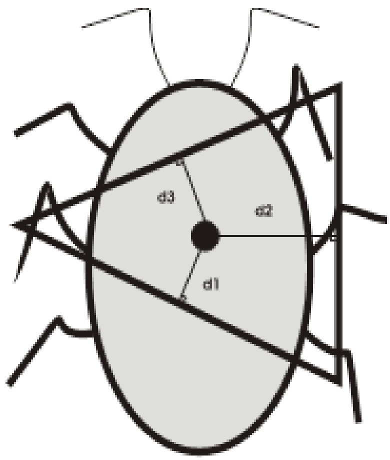


FIGURE 4.3.1. The image shows a picture of the tripod gait formed by the legs of the six-legged insect. The stability margin is the minimum distance from the center of mass to the edge of the triangle of support. Static instability occurs when the centre of mass falls outside the base of support [**58**].

be controlled? Especially in case the robot walks on rugged unpredicted terrain.

The control of locomotion by animals with six or more legs on rough environment is achieved freely and without any effort. They can avoid different obstacles and achieve fascinating walking with different maneuvers without losing their static stability.

Therefore, a legged insect is considered as a good example to learn from its behavior and its walking pattern in order to establish a good walking machine. Two types of insects have been studied extensively and used as one of the most important inspiration resources for walking machine designers. These species were the American cockroach (periplaneta Americana) and the stick insect (Carausius morosus) [**24**]. The cockroach was studied extensively by Pearson [**115**], and the walking of the stick insect has been studied extensively by Cruse [**35, 36**]. The stick insects additionally show a fascinating behavior by maintaining static stability while walking on rough terrain. The following subsection clarifies briefly the static stability of legged walkers.

4.3.1. Static Stability.

An important advantage of six-legged walking is the static stability. Normally the insect's center of mass lies in the polygon of support spanned between the ground touching feet. Thus, a six-legged walker's gait is a static stable movement. It is not allowed to accept three ipsilateral swinging legs. Figure 4.3.1 illustrates a triangular of support spanned by three ground-touching legs.

In addition to the ability to guarantee static stable walking, insects can achieve different gait patterns during their walking.

4.3.2. Insect Walking Patterns.

Walking insects as well as other legged walkers use different walking patterns. These walking patterns do not seem to be predefined but generic. Fixed patterns as well as mixed patterns can be observed in nature. The mixed patterns seem to be smooth transitions from one definable pattern into another. In biology, a walking pattern is classified by the number of feet that are touching the ground simultaneously. As an example the tetrapod gait describes a walking pattern where four feet touch the ground, the tripod gait describes the pattern with three feet touching the ground simultaneously. A slow gait in which just one foot swings can be described in the same way as a pentapod. The three types of insect walking patterns pentapod, tetrapod, and tripod gait are used for different velocities as shown in Figure 4.3.2. While the pentapod represents the slowest walking pattern, the tetrapod is the next faster gait and the tripod is used for the maximal velocity. A pattern mix can be observed while the walker changes smoothly from one pattern to the next one.

Control of walking to achieve static stability and gait patterns walking are the main challenge that face walking machine designers. Thus, robot designers try to emulate the six-legged insects in their behavior, walking patterns, and parts of their control systems without focusing on detailed implementations in their structure or trying to implement their anatomy.

Cruse introduced his theory for the underlying coordination and the emergent gait pattern in stick insects [**37**]. He introduced a decentralized concept to control the individual legs and leg coordination with each other based on six rules (see section 3.5.4). These rules act locally between the legs to produce coordinated stable walking.

Our control system that is implemented on our hexapod robot OSCAR is based on the biological findings that are introduced by Cruse. The following section will introduce the control of OSCAR's individual leg and then the coordination between all of OSCAR's legs.

4.3.3. Control of the OSCAR's Individual Leg.

The cyclic movement of OSCAR's legs consists of two phases: the stance phase and the swing phase. In these phases, OSCAR's leg moves between two extreme positions: the Anterior Extreme Position (AEP) and the Posterior Extreme Position (PEP). They are two points in the three-dimensional fixed coordinate system. Between these points, the distal end of OSCAR's leg moves. Figure 4.3.3 shows OSCAR's leg moving between these predefined extreme positions.

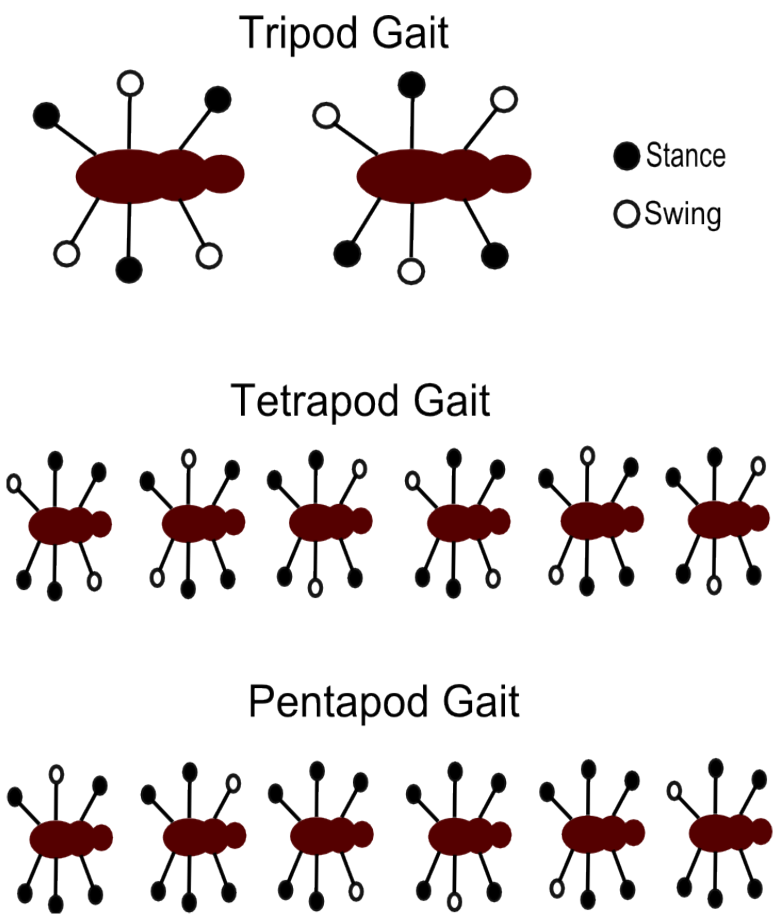


FIGURE 4.3.2. The three basic patterns of insect walking. (top) Tripod gait, (middle) Tetrapod gait, (bottom) Pentapod gait [**13**].

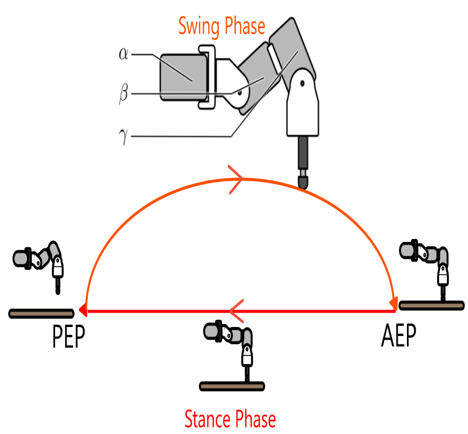


FIGURE 4.3.3. Schematic representation of the stance and the swing trajectory of the foot point between PEP and AEP.

The extreme positions play an important role in defining the currently active workspace of the leg. The distance between AEP and PEP defines the length of the stance and the swing phases. Changing this distance leads to a change in the type of gait pattern and speed of the robot.

4.3.3.1. *Stance Phase.* In this phase, the leg is on the ground, it supports a part of the body weight and propels the robot's body forwards while the α-servo motor moves the leg backwards. Since the distal end of the leg is in contact with the ground on a fixed point, the exerted force from the α-servo motor pushes the robot forward. The distal end of the leg moves in the coordinate system of the leg from position AEP to PEP. During the stance phase, β- and γ-servo motors do not play any role in the leg movement. They just keep contact with the ground during the stance phase and also support the robot's body. The length of the stance phase plays an important role in changing the speed of the robot and also in adapting the robot walking. The stance phase finishes as soon as the leg reaches its PEP.

4.3.3.2. *Swing Phase.* In this phase, the leg moves forward without touching the ground or, in other words, the leg protracts forward. All the leg's joints α, β and γ contribute to execute the swing phase. At the position PEP, the servo motors β and γ achieve elevation and flexion. The leg is lifted up from the ground and it is ready to move forward. The α-joint moves the leg towards the position AEP. At AEP, the β- and γ-joints move the leg down by executing depression and extension. The leg moves down until it detects or finds a foothold. As soon as the leg detects the ground, the swing phase finishes and the stance phase begins anew. As soon as the leg reaches the end of the stance phase in the PEP, the leg switches to the swing phase again. This alternating cycle is integrated by each of OSCAR's leg. This leg's rhythm is similar to the insect leg's movement [**65**].

But, there are some constraints that each leg has to follow before switching from the stance into the swing phase. Otherwise the robot OSCAR will fall down or lose balance during walking especially on rough terrain. To ensure a stable walking, the robot's six legs have to be coordinated. Walking in complex environments requires even more coordination than on flat terrain. The robot's legs can face different obstacles such as objects, slots in the ground, or even larger gaps. In those situations, a stable gait based on a reliable coordination is even more important. The following section will show how OSACR coordinates its legs to achieve stable walking.

4.3.4. Coordination of OSCAR's Legs.

OSCAR coordinates the motion between its legs in a decentralized concept. Each leg uses the same local coordination rule and communicates

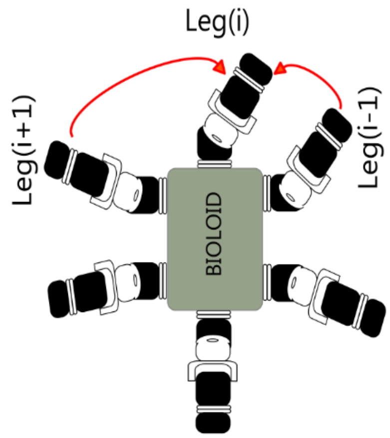


FIGURE 4.3.4. Coordination rule: Leg (i) expected from the neighboring legs (i-1) and (i+1) a ground detection signal before switching into the swing phase.

with its local neighbors. The rule does not permit the leg to switch into the swing phase unless both its neighboring legs are touching the ground. If this condition is not fulfilled, the leg stays in the retraction state (stance phase). As soon as its neighboring legs send a contact ground signal and the leg reaches its PEP, the leg switches into the swing phase and protracts again. This rule is implemented based on Cruse's rule that was introduced in 1990 [**37**] (see section 3.5.4). Figure 4.3.4 shows how the middle left leg communicates with its local neighboring legs before switching from the stance into the swing phase.

Cruse's coordination rule is implemented on the OSCAR robot. The following pseudo code describes how the OSCAR robot's legs coordinate their movements [**54**]. In work, that introduced in [**54, 76**], the robot's leg detects the ground based on switch sensor fixed on the bottom of the robot's leg. While in our work, the leg detects the ground based on the current consumption in the middle joint of robot's leg.

```
if (leg[i-1].detectGround() && leg[i+1].detectGround()) {
    leg[i].swingPhase();
}
else {
    leg[i].stancePhase();
}
```

FIGURE 4.3.5. Pseudo code of the legs coordination

In the pseudo code, the leg [i] switches into the swing phase as soon as the condition is true. The leg [i] switches into swing phase when both neighboring legs (leg [i+1] and leg [i-1]) touch the ground at the same time. In case the condition is false, the leg [i] still retracts back, executing the stance phase. The symbol i presents the leg's number. It is between 0 and 5. The coordination rule for each leg has an influence on the leg's own rhythm. The stance-swing rhythm and the local coordination rule lead to emergent stable walking. The following section describes the emergent walking by OSCAR.

4.3.5. OSCAR Emergent Walking.

The walking emerges through the leg's movement rhythm for each leg and the local coordination rule. In the stance phase, the leg touches the ground while it retracts back. The leg supports the body and carries the weight of the robot. The forward movement of OSCAR is achieved when the leg retracts back until it reaches its PEP. Now, the leg is ready to switch into the swing phase. This condition alone, however, is not sufficient. The leg has now fulfilled the first of two preconditions. The second precondition is a ground contact signal from both neighboring legs. Unless the two preconditions are true at the same time, the leg remains in the stance phase and retracts back. As soon as the two preconditions are true, the leg switches its state into the swing phase and moves forward to the AEP. Then the leg moves down until it touches the ground and the stance phase begins anew. The variation in the stance speed leads to a change in the pattern of the robot gaits, the robot walks in the pentapod gait (one leg swings and the others touch the ground) if the speed of the stance phase is very low. By increasing the speed of the stance phase, the robot changes its gait to tetrapod (two legs swing and the others touch the ground) or to tripod gait which is considered the fastest speed for the robot walking. In the tripod gait, the front and the hind leg swing on one side together with the middle leg from the other side while the remaining legs still touch the ground and execute the stance phase. Due to the leg's rhythm and local coordination between the whole legs, the robot walking emerges based on self-organization.

These leg's rhythm and local coordination are also based on the information received from the internal and external environments. Therefore, our robot is equipped with sufficient sensors to enable it to achieve adaptive and stable walking on rough terrain. The following section sheds light on the more important part of our research. It shows the types of sensory feedback that is used in our approaches to enable OSCAR to achieve adaptive walking on different types of terrains.

4.4. Proprioceptive Sensing

Legged insects are equipped with various sensors that enable them to sense and monitor both their internal and external environments [**24**]. Exteroceptive sensors are used to receive information from the external environments, while proprioceptive sensors are used to monitor the internal environments. During insect walking, proprioceptive information that is obtained from the limbs provides the neural networks with sufficient information about the leg position and its movement as well as forces that are produced in each leg [**3, 114**].

To imitate some of the properties of animal senses, robots need to possess both of these sensors. The robot can achieve adaptive and robust locomotion when it possesses exteroceptive and proprioceptive sensing. Exteroceptive sensors provide the robot's control system with information from the outside world, while proprioceptive sensors monitor the organism's or robot's internal environment to maintain a nearly constant internal environment [**24**].

Our research focuses on the use of proprioceptive sensing to enable legged robots to detect and to achieve adaptive walking on different terrains. This type of sense is used here to provide the control system with signals which indicate the robot's internal states to identify and correct deviations or faults. The current consumption signals in addition to the angular position of the used servo motor are considered the most important parameters that present the internal state of the robot.

In this work, these parameters are used as proprioceptive senors to enable our hexapod robot OSCAR, based on our approaches and organic computing principles, to identify the type of walking substrate and to achieve adaptive walking on different types of terrain.

This work will introduce and discuss five approaches that are based on proprioceptive sensing. The first one sheds light on the applied approach for detecting and executing adaptive walking in accordance with the direction of ground inclination. This approach is based on the evaluation of the local current consumption and angular position in the distal joints of the rear and front legs as proprioceptive sensing. Additionally, the effects of leg disturbance such as hitting an object or leg amputation have been studied for testing the robustness of this approach. The leg disturbance is also based on the proprioceptive sensing

In the second approach, light will be shed on the applied decentralized controller approach to detect slippery ground based on the evaluation of the local current consumption in the coxal joints of the supporting legs as proprioceptive feedback and it also presents the proposed strategies to overcome these challenges. Backward walking is proposed as a reflex reaction once a slippery ground is detected.

In the third approach, the applied decentralized controller approach to detect sandy ground based on the evaluation of the local current consumption and angular position in the middle joint of each leg as proprioceptive feedback will be demonstrated. This approach presents the proposed strategies to achieve adaptive walking on sandy ground.

The fourth approach shows how the robot‘s leg adapts its position to be commensurate with the type of walking surface in order to improve walking on compliant surfaces. This approach is also based on the internal sensing to control the robot legs while they walk on compliant surfaces.

The last approach combines some of the presented approaches to enable the hexapod robot to walk on different complex terrains such as inclined sandy surfaces. The robot can walk on such terrain based totally on the proprioceptive sensing without using any external sensors. For this end, it analyzes the joints’ current consumption and angular position in each servo motor.

The following chapter introduces the first approach that enables our hexapod to walk on an inclined surface based on proprioceptive sensing and organic computing principles.

CHAPTER 5

Inclination Detection and Adaptive Walking on Inclined Surface

5.1. Introduction

Insects can climb different terrains freely, e.g. they can traverse inclined surfaces without losing their statically stable walking. As mentioned previously in chapter 4 (see the section 4.3.1), the statically stable walking for six legged insects is maintained when at least three legs are touching the ground at each moment. The legs that are touching the ground execute the stance phase and the other legs achieve the swing phase. The switching between the insect's legs is based on the coordination rules [**65**]. Besides the coordination between the insect's legs, the insect has to control its body posture to maintain a statically stable walking [**43**]. The control of the body posture has been studied and discussed intensively by Cruse et al. [**35, 43, 44**]. According to the introduced investigation, each leg acts as an independent height controller. The leg controls the distance between the insect's body and the walking substrate [**43**]. Each leg applies a force produced from the torque of different joints of the leg. This force influences the insect's thoracic segments that control the insect's posture based on the applied leg force [**43**]. Another study is introduced by Duysens [**52**] and Akay [**4**]. Their study shows that load information produced by each leg plays an important role to maintain statically stable walking because stable walking on rough terrain requires an effective load distribution between all stance legs. The used approaches by legged animals are considered a good example of transferring them to the legged robots. Therefore, the robotic engineers are trying to benefit from these biological discoveries to develop their legged robots. Legged robots are designed not only to walk on flat ground, but also on inclined ground. However, walking on inclined surfaces requires, besides the statically stable walking, the control of robot's body posture to prevent the robot from overturning. This procedure prevents the robot from falling down which can damage the robot's components or prevents the robot from standing up again while executing its mission.

However, the maintaining of the statically stable walking and control of the body posture of the six-legged robot are considered an important issue and challenging task. This challenging task will be introduced and discussed in this chapter. Several different types of legged robots are designed to achieve stable walking on rough terrain and inclined surfaces, e.g. the LAURON V robot [**124**], a cockroach-like robot [**106**] or AMOS-WD06 [**140**]. A work related to the work introduced here has been conducted on the six-legged Adaptive Suspension Vehicle (ASV) [**117**]. In ASV, the desired force on the robot's trunk is calculated by the robot controller based on the trunk's position and velocity error. This force is distributed to the supporting legs that touch the ground to adapt the robot body's posture. Another similar work to ASV has been conducted on the quadruped robot Warp1 [**121**]. The body posture of Warp1 is controlled by distributing the forces applied onto the trunk by the legs. Its controller calculates the applied forces on the trunk and also calculates the desired force. Then it distributes the forces to the supporting legs. In order to increase the trunk height, the vertically applied force on the legs increases by the same amount. To pitch the trunk forward, more vertical force is applied in the rear leg and less in the front legs [**121**]. Steingrube et al. [**140**] present how their six-legged robot AMOS-WD06 reacts to faced inclined surfaces. AMOS-WD06 is based on a neural circuit as a central pattern generator (CPG). For adaptive walking on inclined surfaces, the motor current consumption is stored as a default value while the robot walks on a flat surface. As soon as the robot faces a tilted plane, the motor current consumption is compared with the stored default current consumption. The produced error signal is used to modify the driving synaptic weight of the learning neuron that leads to a change in the output of this neuron. Based on the output of the learning neuron, the CPG selects the appropriate behavior. The robot choses a slow wave gate that enables the robot to walk with the lowest energy consumption. When AMOS-WD06 encounters a slope again, the inclinometer sensor is triggered leading directly to the selection of the slow wave gait without further learning [**140**].

All introduced works have been conducted by using an inclinometer or acceleration sensor to provide the system with sufficient information about the ground and direction of the inclination.

Our work here is to enable the hexapod robot OSCAR to detect the direction of inclination and to adapt its body's posture based on proprioception sensing and organic computing principles without using more additional sensors.

This work is based on our previous work that is introduced in [**5**]. It is based on the applied decentralized approach for detecting the inclination and the direction of the walking surface. OSCAR detects the inclination and adapts its body posture relative to the direction of inclination based on the evaluation of the consumed current and angular position in the distal joint (γ-joint) in each leg. Additionally, this work shows the effect of the elevator reflex as well as the effect of the leg amputation on the statically stable walking while walking up and downhill. This approach has been implemented in the ORCA architecture and tested on the latest version of the hexapod robot OSCAR, based on the Bioloid robot kit that was introduced previously in chapter 4.

5.2. ORCA Implementation and Current Consumption Measuring

Before explaining our applied approach for adaptive walking on inclined surfaces, it should be clarified how this approach has been implemented in the ORCA architecture as mentioned previously in chapter 3 (see section 3.9 for more information).

In this work, the OCU of each leg has the same structure and the same functionality. It has to monitor the current consumption and angular position in the distal joint (γ-joint). The leg's OCU does not react unless the γ-current consumption exceeds the predefined threshold. As soon as the γ-current consumption exceeds the predefined threshold, the OCU of the considered leg sends a signal to the higher level of the system. The higher level of the system decides that the robot faces an inclined surface and defines the direction of this inclination (uphill or downhill) and decides when the leg's OCU has to change the BCU's parameters of the considered leg. The changed parameters lead to a change in the body posture relevant to the direction and degree of the inclination. Figure 5.2.1 shows the ORCA architecture of one leg.

The approach for inclination detection used here is based on the evaluation of the current consumption in a γ-servo motor. The used servo motor AX-12 offers the properties to read the current consumption, current position, and some other parameters. The current consumption and position values are shown as raw values. A current consumption value of 1024 corresponds to the maximal torque at 1.5A current. The angular position value directly corresponds to the joint position in degrees.

The following section will show how OSCAR detects the direction of inclination based on the analysis of the γ-current consumption.

5.3. Inclined Surface Detection

Normally, inclinometer or acceleration sensors are used to detect the inclination and direction of the traversed terrain. Those increase the complexity of the control system and its fault probability. Furthermore, additional sensors increase the cost of the walking robot. Therefore, the approach introduced here will discuss how the OSCAR robot detects the inclination and its direction and adapts the robot body posture accordingly depending on the proprioception sensing based on analyzing and evaluating the consumed current in each γ-servo motor. As is well known, walking on inclined surfaces does not only cause the robot to overturn, but also leads to different load distributions among

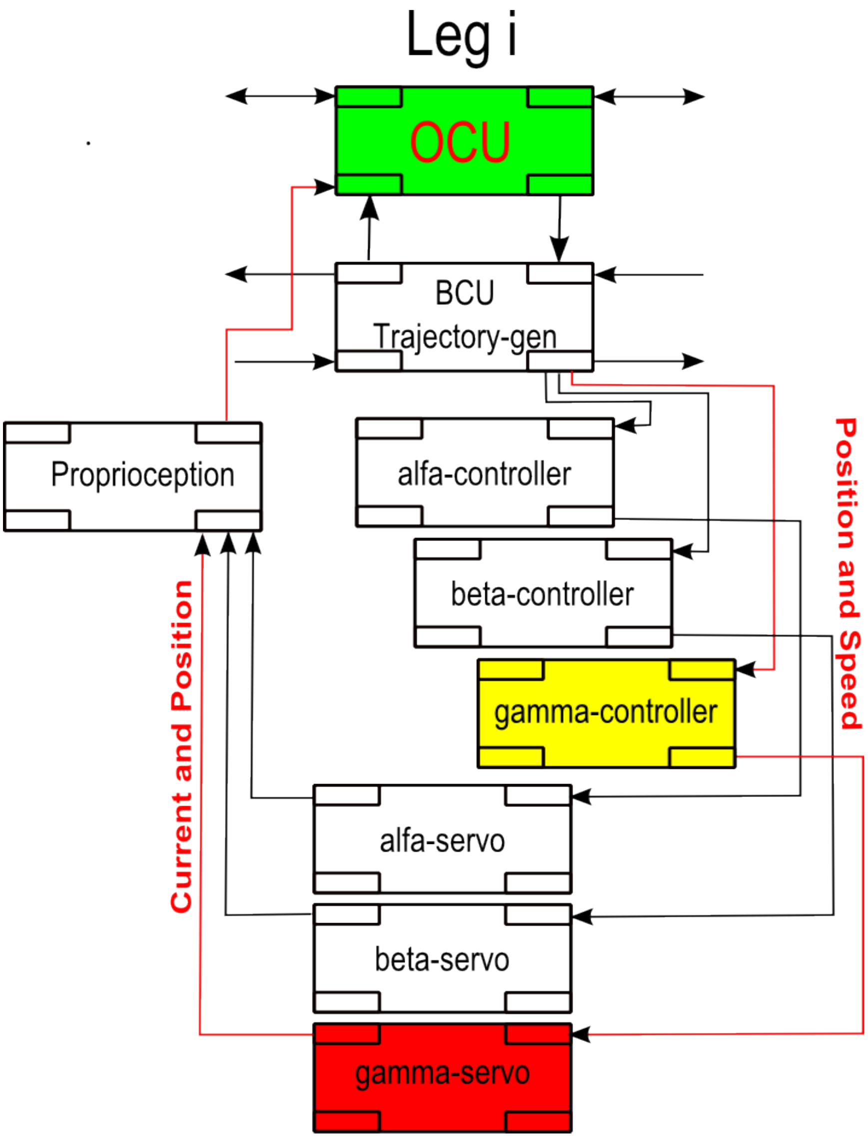


FIGURE 5.2.1. The leg control with ORCA. OCU monitors the current consumption of γ-servo and changes the parameters of β- and γ-controller if the robot detects inclination.

all of the robot's legs. The most affected legs are the front legs in case the robot walks downhill or the hind legs in case the robot walks uphill. For the leg that consists of multi-joints, the most affected joint will be the distal joint when the robot walks on an inclined surface. This is due to the difference in the distribution of the load on the robot legs when the robot walks uphill or downhill. In other words, the weight of the walking robot is not equally distributed on the legs that carry the robot during the stance phase while the robot walks on the inclined surface.

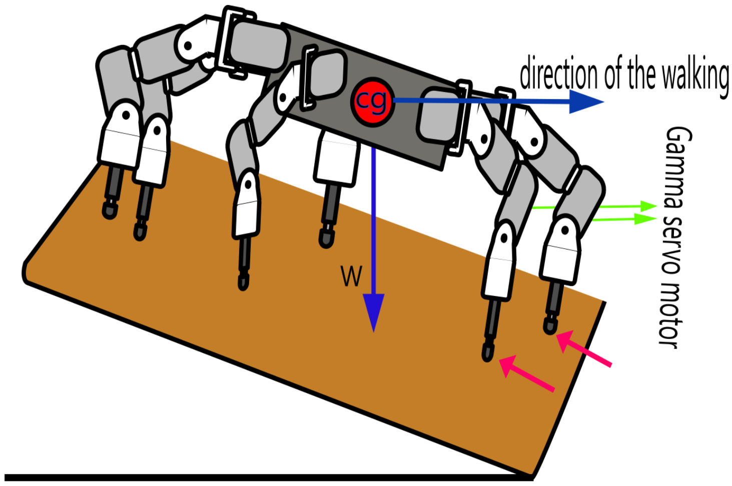


FIGURE 5.3.1. OSCAR's schematic while walking downhill

For the OSCAR robot that is used in this research, the most affected joints while walking on inclined surfaces are the distal joints (γ-joints). As a result, the consumed current in the γ-servo motor in some legs while walking on an inclined surface is higher than for a robot walking on flat ground. E.g. for downhill walking, the front legs carry more load than the hind legs. Therefore, the current consumption in the γ-servo motor in the front legs increases significantly during the stance phase and is higher than the current consumption in the γ-servo motors in the hind legs. Figure 5.3.1 shows the OSCAR's schematic while it walks downhill.

Then, the downward or upward inclination is detected if the current consumption in the γ-servo motor of the considered legs exceeds a predefined threshold during the stance phase for either both front legs or both rear legs, respectively. The following steps show how our algorithm detects inclined surfaces:

(1) Read the current consumption in all γ-servo motors during the stance phase.
(2) Compare the measured values with the predefined stored threshold.
(3) If the current consumption in the γ-servo motor in front legs exceeds the predefined threshold while executing the stance phase, the robot faces downward inclination.
(4) If the current consumption in the γ-servo motors in the hind legs exceeds the predefined threshold while executing the stance phase, the robot faces upward inclination.

The minimum current consumption in the γ-servo motor for downward inclination will be denoted as T_{down}, the minimum γ- current consumption for upward inclination as T_{up}. The predefined thresholds are estimated while the robot walks on a flat and inclined surface by measuring the current consumption by experiments.

Two different thresholds are used for front and rear legs, $T_{down} > T_{up}$. The rear legs have a lower threshold (T_{up}) to detect uphill walking. The front legs have a higher threshold (T_{down}) to detect downhill walking. This prevents the robot from getting confused when it walks on flat ground as sometimes the rear legs that execute their stance phase push the front legs forward which are in the stance phase as well. This action leads to an increase in the current consumption in the front γ-joints while walking on flat ground or during uphill-downhill walking. The inclination test now reads:

$$D_{inc}(t) = \begin{cases} -1 & \text{if } I_\gamma^{FL}(t) > T_{down} \text{ and } I_\gamma^{FR}(t) > T_{down} \\ +1 & \text{if } I_\gamma^{HL}(t) > T_{up} \text{ and } I_\gamma^{HR}(t) > T_{up} \\ 0 & \text{else} \end{cases}$$

Here, -1 indicates downhill and $+1$ uphill walking. After a successful detection of the inclination, the robot adapts its body posture. The following section will discuss how to react to the inclination.

5.4. Control of the Body Posture

This section discusses how the robot adjusts its body posture to maintain statically stable walking as soon as it detects the inclination. The control of the body posture of the robot leads here to move the center of gravity inside the vertical projection of the support feet to avoid an overturning of the robot while walking on an inclined surface. In case the robot detects uphill or downhill inclination, the reaction step will change the front and hind legs' parameters as long as the previously mentioned considered conditions are true. The affected legs that detect the inclination will update their position to expand more and lift up the downward facing part of the robot's body. The legs that are on the opposite side will, similarly, flex more to make the robot's body as horizontal possible. The following steps clarifies how the robot adjusts its posture as soon as it detects the inclination:

(1) Downhill walking: the robot expands the front legs and flexes the hind legs by changing the β- and γ-servo motor position.
(2) Uphill walking: the robot expands the hind legs and flexes the front legs by changing the β- and γ-servo motor position.
(3) Stop the body posture adaptation if the current consumption drops below the predefined threshold.

The leg OCU thus updates the position of the β- and γ-servos for the front and hind legs as follows:

$$P^F(t+1) = P^F(t) - D_{inc}(t) \cdot step$$
$$P^H(t+1) = P^H(t) + D_{inc}(t) \cdot step$$

Here, P^F and P^H is the position of the front and hind legs respectively and *step* is the change in position in degrees.

The decision process is run cyclically. In case the condition becomes false due to a dropping of the consumed current under the threshold, the leg keeps the last position and the robot continues its mission with its new body posture.

As soon as the robot traverses from inclined to flat ground, the leg load is again unequally distributed. The inclination test ensures that the original body posture is regained.

5.5. Reflexes and Amputation

In a natural environment, the robot might face some obstacles or loose one leg while walking on an inclined surface. Therefore, in this chapter, the influence of disturbances, e.g. triggered reflexes and leg amputation, on the proposed inclination detection algorithm and vice versa, will be further analyzed. The following section introduces the elevator reflexes and their effects on OSCAR's walking.

5.5.1. Elevator Reflexes.

Walking insects show fascinating behaviors to avoid unforeseen obstacles. Biological researchers have proved that locusts have an elevator reflex if one of their legs encounters an object while executing the swing phase [**116**].

The elevator reflex has been previously implemented and tested on OSCAR-3 by EL Sayed Auf [**12, 93**]. However, the introduced elevator reflex was implemented while the robot was walking on flat ground showing how the OSCAR leg triggers the elevator reflex as soon as one leg hits an obstacle. EL Sayed Auf's work did not handle the effect of the elevator reflex while walking on inclined surfaces and vice versa.

In this research, the elevator reflex serves as an example of the experiments and will be implemented in the same way as the one introduced in the work [**12, 93**]. However, in this work, we will add some additional adjustments to the elevator reflexes to improve walking on inclined surfaces.

In the previous and the current work, an obstacle is detected by evaluating the α-current consumption during the swing phase. The α-servo motor moves the leg from the PEP to the AEP in the swing phase and vice versa during the stance phase (see subsection 4.3.3). As soon as OSCAR's leg hits an object while executing the swing phase, the current consumption in the α-servo motor rises significantly exceeding the predefined threshold. This stimuli can be used to detect an object which blocks the leg to reach a given target position. The system interrupts the leg's swing phase and triggers the elevator reflex as fast stereotype movements trying to overcome the encountered object. The elevator reflex is divided into three phases: the first one is the first elevation. In this phase the leg retracts and moves up trying to overcome the obstacle. In case it hits the obstacle again, the leg executes the second phase. In this phase, the leg retracts again and moves up higher trying to overcome the faced obstacle. If the leg hits the obstacle anew, the leg attains the extension phase. In this phase, the γ-joint extends the leg.

To achieve the elevator reflex, the leg retracts contrary to its current direction. In El Sayed Auf's approach, the leg moves to the previous PEP position and then executes the first elevator [**12**].

In our applied approach, the leg returns to the position PEP$\pm\delta$ instead of PEP (The symbol δ presents a small displacement behind PEP). The δ value is either positive or negative based on the servo motor position. In this procedure, the leg frees itself from the probably touched obstacle especially when the obstacle comes at the beginning of the swing phase (PEP) (this case is detected per experiments). This is the first improvement on the leg trajectory while executing the elevator reflex. This improvement plays an important role while walking on an inclined surface especially when the robot walks downhill because the weight of the robot and the stance legs push the robot body against the faced obstacle. Then OSCAR's leg is lifted up by the β-joint to a higher position (first elevation or first phase). Then the leg moves forward again. In case the leg hits an object again, the α-joint moves the leg again to PEP$\pm\delta$, the β-joint moves to its maximal position (second elevation or second phase). The leg tries to overcome the faced object again. In case the leg hits the object once again, the third phase (extension) is triggered. The α-joint moves the leg back once again until PEP$\pm\delta$, the γ-joint updates its angular position which causes the leg to expand. The α-joint moves the leg forward again trying to overcome the faced object. When the leg hits the object again, the walking behavior is stopped and a signal is sent to the higher level to change the robot's path. Figure 5.5.1 shows a possible trajectory for the elevator reflex.

Another case that will be discussed here is the effect of the control of body posture on the elevator reflex while walking on an inclined surface. As mentioned previously, the robot's body posture has to be adjusted based on extension and flexion of its front and rear legs according to the detected inclination. For instance, the robot walks uphill, in this case the front legs are flexed and the rear legs are expanded to adjust the robot body posture. Now, if one of the front legs faces an object, the elevator reflex is limited to execute the immediate execution of the second elevation and extension without trying to execute the

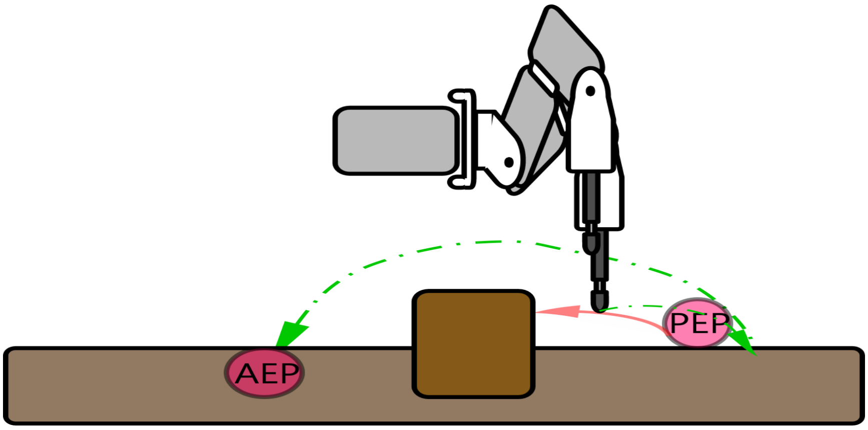


FIGURE 5.5.1. Elevator reflex: the leg hits the object. It retracts back and then moves up trying to overcome the faced object. Here, the leg achieves its swing phase which is presented by the red trajectory. The green trajectory presents the new trajectory of the leg while executing the elevator reflex.

first elevation. However, the rear legs can achieve all reflex phases as if the robot walks on flat ground. The reason behind this is the following: The front legs are flexed and maybe their β-joints exceed the predefined angular position for the first elevation. The hind legs are still able to achieve the first and second elevation and extension. The same scenario is conducted in case the robot walks downhill. The front legs could execute the first and second elevator reflex as well as the extension reflex, while the rear legs are limited to the second elevation and extension. Another modification has been added in this work concerning the rear legs.

As for the rear legs, they have the ability to detect the encountered object by monitoring the current consumption in both α- and γ-joints. If the current consumption in one of these joints rises above the predefined threshold, the elevator reflex in the considered rear leg is executed because in some cases the most affected joint is the γ-servo motor instead of the α-joint especially when the robot walks on an inclined surface. This is due to the position of the rear legs, i.e. γ-servo motor is the most affected at the begin of swing phase when the hited object at the PEP and the leg moves up (this case has been observed by experiment). In this case, the rear legs detect the faced object as soon as the current consumption in the α-servo motor or in the γ-servo motor exceeds the predefined thresholds. These thresholds are defined per experiments. Additionally, the rear legs enlarge their swing phase as soon as they hit an obstacle to ensure the robot overcomes or climbs over the faced object.

5.5.2. Leg Amputation.

The second disturbance that will be discussed here is the leg amputation. This case represents the worst case that the robot faces while walking on rough terrain. Indeed, several researches have been conducted to enable the robot to cope with this case and to achieve adaptive walking with amputated legs, i.e. for OSCAR-2 [**54**], OSCAR-X, [**76**] and DLR-Crawler [**62**]. Yet, it was focused on walking on flat hard ground.

In the context of our work, the impact of leg amputation on the robot walking will be discussed in several different scenarios e.g. while walking on an inclined surface, which will be introduced in this section, a sandy surface, and other terrains that will be demonstrated later in the next chapters.

This section will show the effect of leg amputation on the statically stable walking while the robot walks on inclined surfaces. There are two methods to achieve leg amputation either in software that is used in OSCAR-2 [**54**] or in hardware, where the system has the ability to detach the affected leg and which is used in OSCAR-X [**76**].

In the context of this research, The amputated leg will be simulated based on the software, i.e. a leg will be moved and stiffened on top of the body to minimize its impact on the remaining legs. Figure 5.5.3 shows how the OSCAR's leg is stiffened to simulate the amputated leg. As soon as the robot loses one of its legs, the servo motor speed is decreased and the length of leg's step is shortened to ensure stable walking. This behavior is also observed with walking insects [**133**]. Furthermore, the robot expands its leg's coordination rules based in the same way that is used in OSCAR-3 [**54**]. Figure 5.5.2 shows how OSCAR extends its leg's coordination as soon as one of its leg is amputated.

Additionally, the robot reconfigures its legs' positions as soon as it looses one of its legs. The reconfiguration is based on the direction of inclination and location of the amputated leg.

5.6. Experimental Results

For testing our introduced approach on our hexapod robot, a test ground has been established. This test ground is a board of 240x50 cm with a 15° inclination in an indoor area. To simulate the rough ground, the used board is covered with a rough carpet. The robot is equipped with an inclination sensor as a reference to our adjustment of the body posture. Figure 5.6.1 shows OSCAR's robot while walking uphill.This work has focused on the most common critical situations which may be encountered by the robot during its locomotion on rough terrain. The conducted experiments comprise the uphill and downhill inclination detection without robot body posture adaptation and with robot body posture adaptation. Additionally, two experiments have been conducted to show how the robot handles disturbances like obstacles

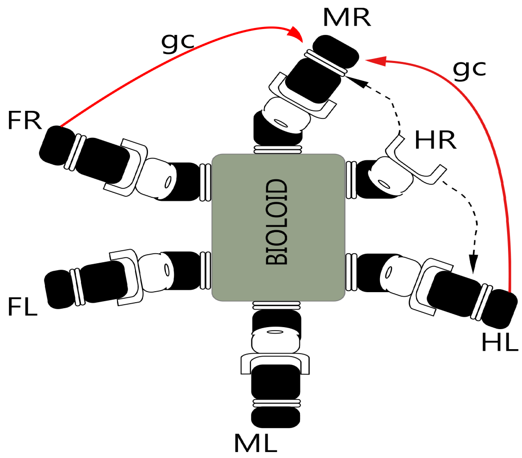


FIGURE 5.5.2. Coordination influences that allow the middle right leg (MR) to swing if the hind right leg (HR) is lost. The middle right leg is allowed to swing if the front right leg (FR) and the hind left leg (HL) are sending a ground contact (gc) signal. The dotted line represents the missing ground contact signal of the hind right leg.

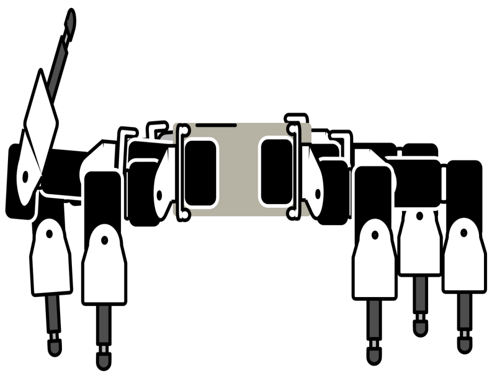


FIGURE 5.5.3. Simulation of the leg ampuation

and leg failures, i.e. leg amputation. The following section discusses the obtained results when the robot walks on uphill and downhill surfaces without robot body posture adaptation.

5.6.1. Uphill-Downhill Detection Without Robot Body Posture Adaptation.

In these experiments, the robot walks on uphill and downhill surfaces without adjusting the body posture. These experiments aim to show how the consumed current in the γ-servo motor (distal joint) increases significantly and in some legs more than the others. In the following

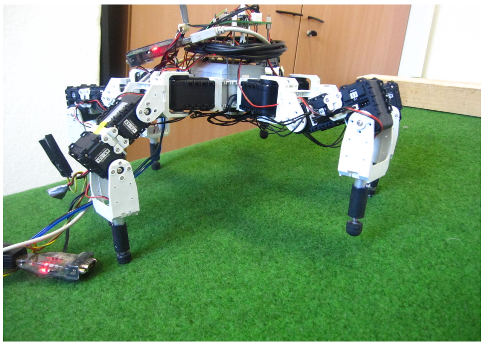


FIGURE 5.6.1. OSCAR walks up. The rear legs are expanded more than the front legs

section, walking uphill without adjusting the robot body posture will be introduced.

5.6.1.1. *Uphill Walking.* Figure 5.6.2 shows the obtained results while the robot walks on an uphill surface. It is clear that the consumed currents in the γ-joints for the hind legs (HR and HL) exceed the predefined threshold for uphill walking as shown in the center of Figure 5.6.2. The first exceeding for the hind right (HR) leg is at data time 11, when the leg executes its stance phase. The γ-current consumption is still above the predefined threshold as long as the leg touches the ground. As shown in this Figure, the current consumption for the HR leg exceeds the threshold at each stance phase. The same scenario is noted in the hind left (HL) leg. Its γ-current consumption exceeds its predefined threshold at each switching to the stance phase. The first exceeding is at data point 46. For uphill walking detection, the γ-current consumption in the rear legs exceeds the threshold at data times 11 and 46 respectively and thus successfully detect the upward inclination during the first stance phase. For the front legs (FR and FL) as shown at the top of Figure 5.6.2, the γ-current consumption does not exceed the predefined threshold for uphill walking. The bottom of Figure 5.6.2 shows the data reference inclination of the robot obtained from a tilt sensor which indicates to the value -15° inclination.

5.6.1.2. *Downhill Walking.* Figure 5.6.3 shows the obtained results while the robot walks downhill. As noted at the top of this Figure, the γ-current consumption for the front legs (FR and FL) rises significantly more than the consumed current in the rear γ-joints (HL and HR). The current consumption of the γ-joint of the front leg (FR) exceeds the threshold at data point 24, then the current consumption exceeds the predefined threshold at each stance phase. The FR curve swings during the stance phase above and under the threshold. This swinging is due to the position of the leg on the ground. At the beginning and the end of the stance phase, the γ-servo motor is more affected compared to the middle of the stance phase. The same scenario is noted in the

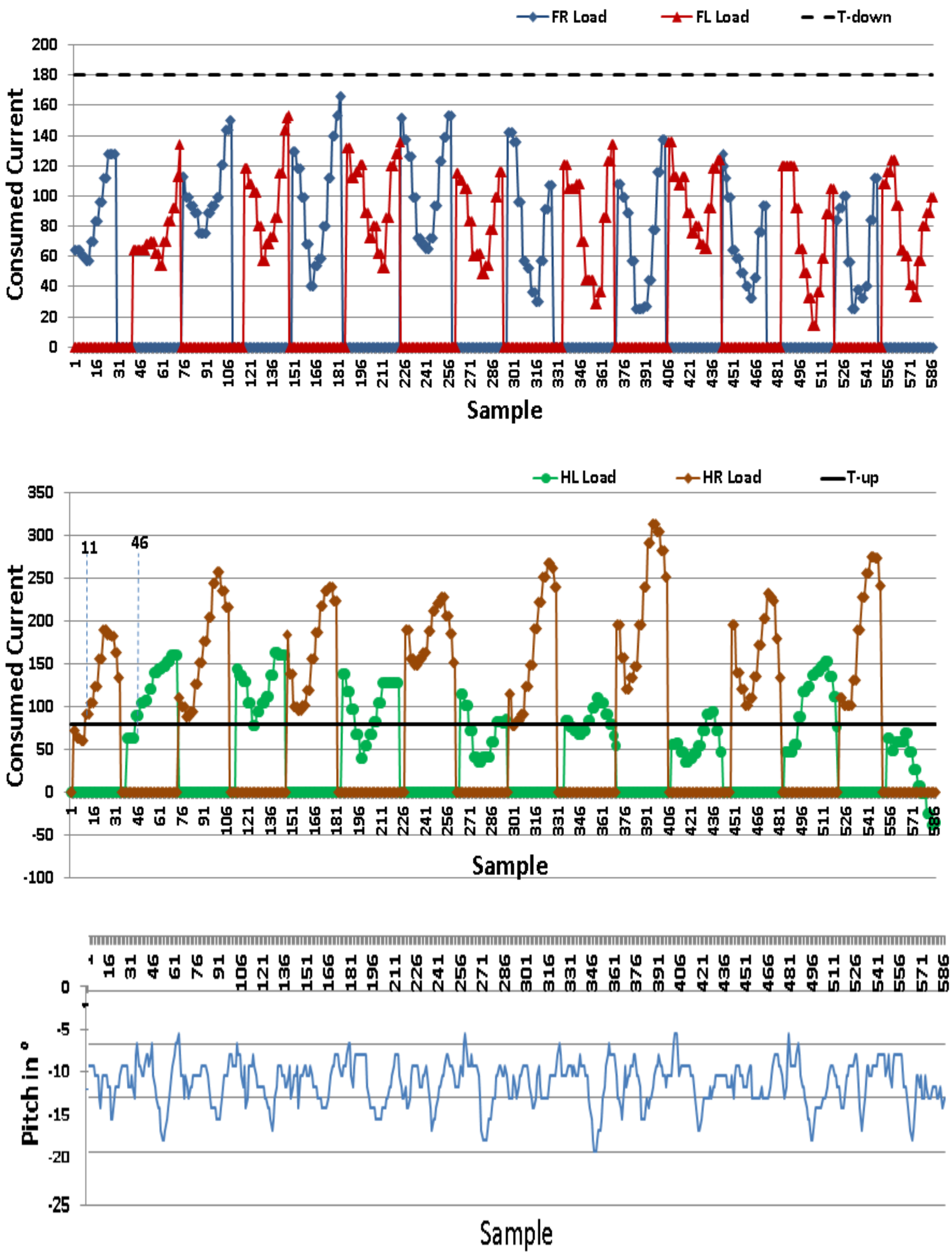


FIGURE 5.6.2. Walking without correction on uphill surface. Top: γ-consumed current in the front legs (FR and FL) during stance phase. Center: γ-consumed current in the rear legs (HR and HL) during stance phase. Bottom: reference inclination of the robot obtained from a tilt sensor.

front left (FL) leg. The first exceeding of the γ-current consumption in the FL leg is at data time 58 as shown at the top of Figure 5.6.3. The exceeding is repeated at each stance phase. For downhill walking, the γ-current consumption of the front legs exceeds the threshold at data times 24 and 58 respectively and thus successfully detects the downward inclination during the first stance phase. The bottom of Figure 5.6.3 shows the degree of inclination obtained from the tilt sensor. Here, it indicates to the value of the inclination 15°. Also as noted in the center of Figure 5.6.3, the current consumption in the γ-joint of the hind right leg (HR) exceeds the threshold for a while several times during its stance phase while the robot walks down. However, that is not enough to recognize the inclination direction unless the current consumption remains above the threshold during the whole stance phase and, additionally, the γ-current consumption in the adjacent leg exceeds its predefined threshold.

5.6.2. Uphill-Downhill Detection With Robot Body Adaptation.

The following experiments show how the robot detects inclination and its direction as well as how the robot adjusts its posture according to the direction of inclination during uphill and downhill walking. In the following experiment, the same test ground that was used previously is used. The following section shows the obtained results while OSCAR walks on an uphill surface.

5.6.2.1. *Uphill Walking With Robot Body Adaptation.* The center of Figure 5.6.4 shows the consumed current in the γ-joints of the hind legs (HR and HL) while the robot walks uphill. As shown in this Figure, the consumed current in the γ-joint of the HR leg exceeds the predefined threshold at data point 11 and increases as long as the leg executes the stance phase. For the HL leg, the consumed current in the γ-joint increases significantly as soon as it touches the ground at data point 60 as shown at the center of Figure 5.6.4.

Here, the system recognizes two sequential crossings of the threshold by the hind legs. In contrast to the front legs (FR and FL), the current consumption in their γ-joints does not exceed the predefined threshold as shown at the top of Figure 5.6.4. In this case, the robot system recognizes that the current consumption in the hind legs is higher than in the front legs. The bottom of Figure 5.6.4 shows how the robot adjusts its body posture from an inclination of -15° to an average inclination of -6°. This signal is obtained from an external additional inclination sensor. It is added to prove the strength of our used approach. This adjustment is finished at data point 240 due the dropping of the current consumption of both hind legs below the threshold. The body posture gets thus nearly horizontal during uphill walking and the

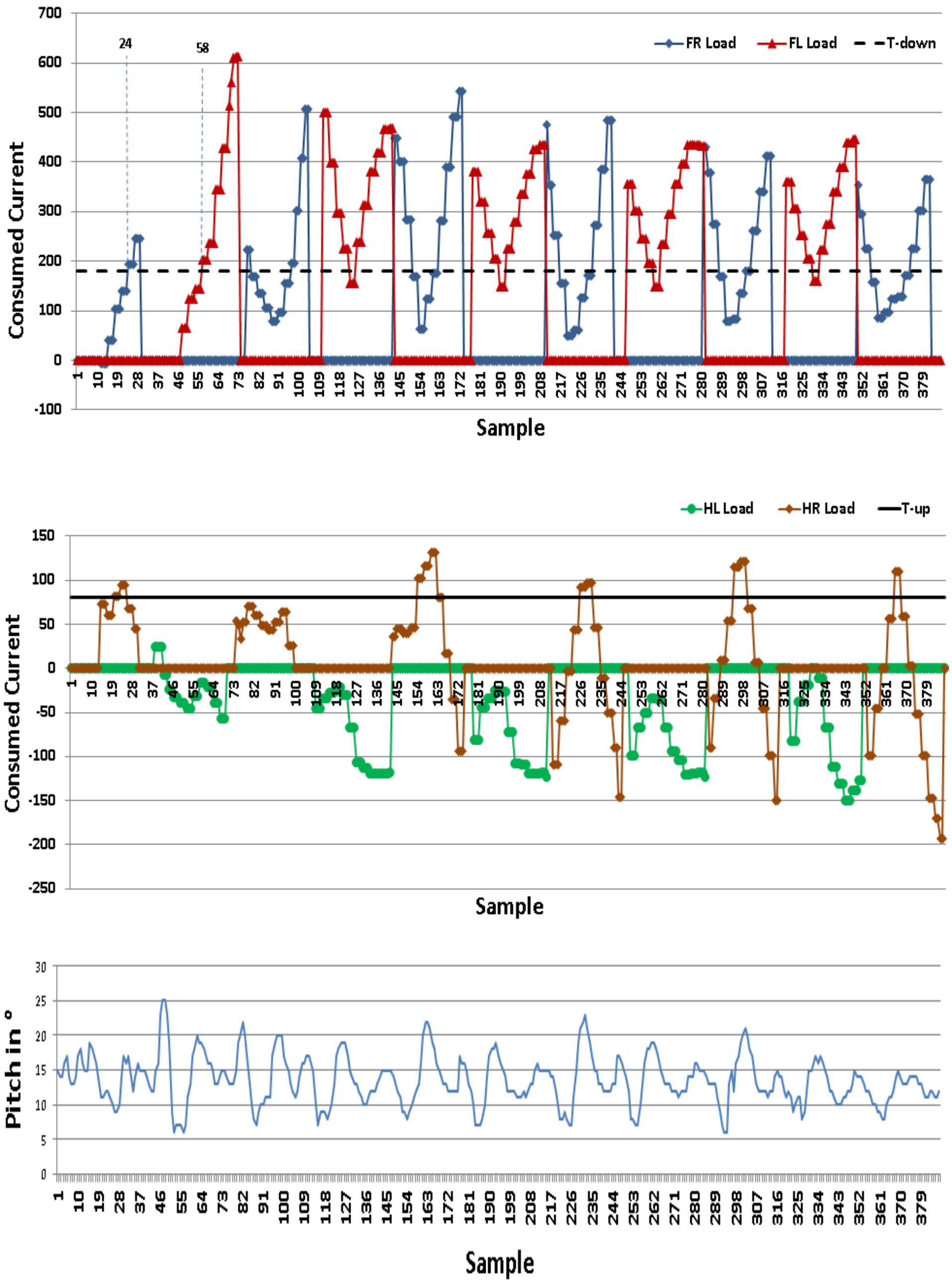


FIGURE 5.6.3. Walking on downhill surface without controlling the robot body posture. Top: consumed current in γ-joints for front legs (FR and FL) during stance phase. Center: consumed current in γ-joints for rear legs (HR and HL). Bottom: reference inclination of the robot obtained from a tilt sensor.

current consumption in the affected joints is decreased significantly as shown in the hind legs in the center of Figure 5.6.4.

5.6.2.2. *Downhill Walking With Robot Body Adaptation.* In this experiment, the robot walks on a downhill surface. Figure 5.6.5 shows the obtained results while walking on a downhill surface. The top of Figure 5.6.5 shows the current consumption in the γ-joints of the front legs (FR and FL). The γ-current consumption in the FR and FL legs exceeds the predefined threshold at data point 22 and 50 respectively. It remains above the threshold during the whole stance phase. The robot system recognizes the downhill walking and starts adjusting its body posture as shown in the bottom of Figure 5.6.5. The bottom curve presents the pitch signal in the direction of walking obtained from an external additional inclination sensor. At data point 135, the current consumption of the FR leg drops below the threshold and the robot stops adjusting its body posture. The body posture is reduced to an average inclination of 6°. The center of Figure 5.6.5 shows the consumed current in the γ-joints of the hind legs. The γ-current consumption of the hind legs sometimes exceeds the predefined threshold while the robot walks down, but this exceeding does not remain until the end of the stance phase. Therefore, this transient exceeding is ignored by the OCU unit. Also, the robot system does not react unless the predefined conditions that have been defined previously are true. The robot walks on a downhill surface with an almost horizontal body posture.

The following section shows the elevator reflex while walking on an inclined surface.

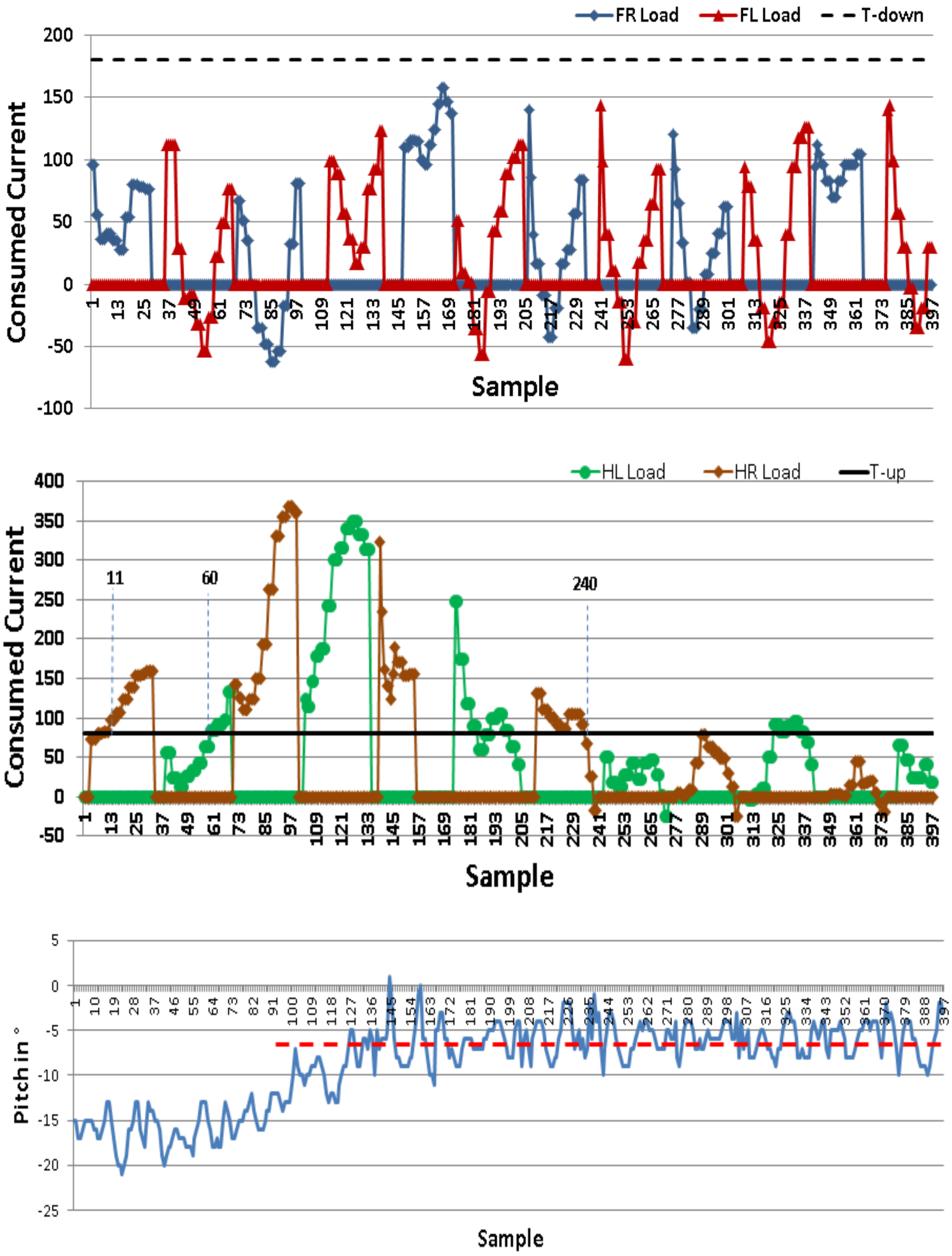


FIGURE 5.6.4. Walking on uphill surface with control of the robot body posture. Top: Consumed current in the γ-joints of the front legs (FR and FL) during stance phase (top). Consumed current in γ-joints for the rear legs (HR and HL) (center). Reference inclination of the robot obtained from tilt sensor (bottom).

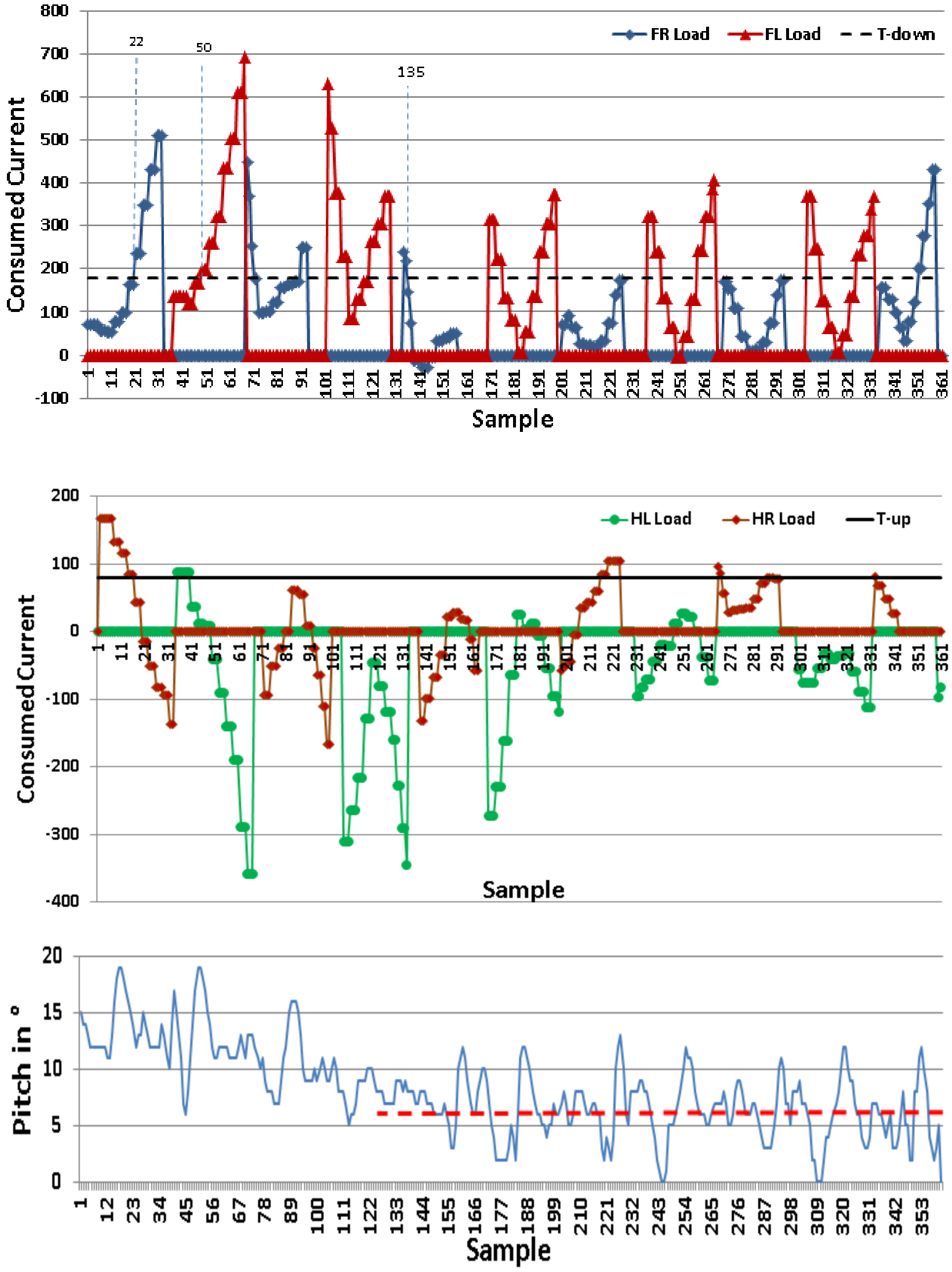


FIGURE 5.6.5. Correction of downhill inclination. Consumed current in the front legs (FR and FL) during stance phase (top). Consumed current in the hind legs (HR and HL) (center). Reference inclination of the robot obtained from tilt sensor (bottom).

5.6.3. Elevator Reflex.

To evaluate the elevator reflex, several experiments have been conducted on the OSCAR robot. Some experiments have been carried out while the robot walked on flat ground showing how the front and the middle legs overcome a faced object. Other experiments have been conducted to show how the hind legs overcome an encountered object. Additionally, an experiment has been conducted to show the effect of this reflex on the robot walking on an inclined surface. Firstly, we will demonstrate how the middle left (ML) leg handles an object.

5.6.3.1. *Elevator Reflex of Middle Left Leg While Walking on Flat Ground.* Figure 5.6.6 shows the obtained result in case the robot walks on a flat surface. The elevator reflex in this experiment is triggered as soon as the middle left leg hits an object and the α-current consumption exceeds the predefined threshold. Figure 5.6.6 represents the angular position and consumed current for the joints of the ML leg during just one swing phase. Here, the ML leg moves in the swing phase from position PEP = 170° to the position AEP = 130°. This Figure is divided into four parts. Part (1) represents a small part of the swing phase. It begins at 0 s. The first elevator reflex is triggered after 1.31 s. The current consumption in the α-joint exceeds the predefined threshold of 39% (red curve) at the α-joint position 146°. The α-joint moves the leg backward to the position PEP+δ (from 146° to 180°). Then, the β-joint lifts up the leg by changing its position from 120° to 90° as shown in part (2). Then, the α-joint moves the leg forward as shown in part (2). The second elevator reflex is triggered 1.93 s after the first trigger when the current consumption in the α-joint exceeds the threshold again at 3.23 s. The α-joint achieves the same reaction as mentioned for the first elevator, it moves back to position 180° (PEP+ δ), the β-joint moves the leg higher by changing its position from 90° to 70°, the γ-joint is still at the same position 120° as shown in part (3). After two last triggers, the β-joint reaches its maximal position. Then, the α-joint moves the leg to its given position. In this moment, any collision with an object leads to the triggering of the extension reflex. The current consumption in the α-joint rises again at time 5.28 s. The α-joint moves the leg back again, then just the γ-joint changes it position from 120° to 180° leading to an extension reflex as shown in part (4) . The first reflex begins at 1.31 s, the last reflex finishes at time 6.54 s. After three reflexes, the leg is lifted and expanded to overcome the faced object. In case of detecting a probable obstacle again after elevation and extension, the system stops the actual swing phase and puts the leg down at its last PEP. In this case, a signal is sent to a higher level in the system to change this behavior.

5.6.3.2. *Elevator Reflex of Hind Right Leg While Walking on Flat Ground.* Figure 5.6.7 shows the obtained results from the HR leg while the robot walks on a flat surface and the leg faces an object. The

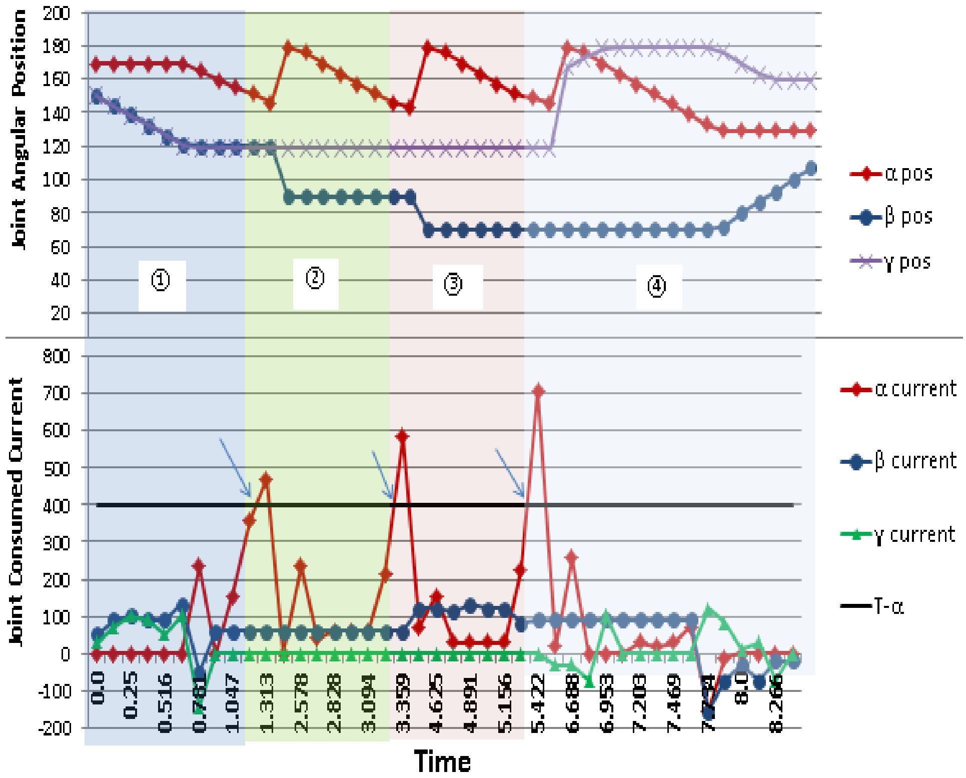


FIGURE 5.6.6. α-, β- and γ-joint angular position and current consumption over time in seconds showing elevation and extension reflexes triggered in quick succession in the ML leg.

depicted results present the angular position and consumed current for all leg's joints over time and during just the swing phases. The stance phases are not depicted because they are not relevant to the elevator reflex. Figure 5.6.7 has been divided into 4 parts. Part (I) represents the normal leg's trajectory during the swing-stance phase. The leg begins at PEP (160°) and finishes its swing phase as soon as it detects the ground at AEP (200°) relative to the α-joint. In this part, the β- and γ-joints lift up the leg by changing their angular position from angular position 140° to 180°. As soon as they reach their required position, they stop moving. Then, the α-joint moves the leg forward by changing its angular position from position 160° to 200° as shown in the first part (1). Then, the β- and γ-joints move the leg down by changing their angular position from position 180° to 140° again to enable the leg to detect the ground. The first leg's step is done without hitting any object. The leg executes the normal swing phase without changing its trajectory. Part (2) represents the next swing phase. At the beginning of this swing phase, the β- and γ-joints lift up the leg by updating their angular positions. But due to the existence an object that hinders the leg elevation, the elevator reflex in this case is triggered at time 3.73 s due to a significant increase in the consumed current in theγ-joint. The γ-current consumption exceeds its threshold (the higher threshold) as shown in part (2) in the bottom of Figure 5.6.7. Now, all the leg's joints update their angular position to overcome the encountered obstacle. The γ-joint changes its position to release the leg from the touched obstacle. Then, the α-joint moves the leg to the position 150° (PEP - δ), then, the β-joint lifts up the leg by updating its angular position from 170° to 210° as shown in the top of Figure 5.6.7 in part (2). The leg starts a new swing phase with a new position. At time 5.82 s, the current consumption in α-joint increases significantly exceeding its threshold (low threshold). All the leg's joints adjust their position to lift up the leg more as shown in part (3). The β-joint moves from position 210° to 230° to reach its maximal position. The leg is lifted to the maximal position. At time 8 s, the current consumption in the α-joints rises again exceeding the threshold. The α-joint moves the leg back to 150° (PEP - δ) and simultaneously, the γ-joint updates its position to 120°, which represents the maximal extension of the leg as shown in part (4). The leg is lifted and expanded to its maximal position. The α-joint moves the leg forward overcoming the encountered object and stops at position 220° (AEP + 2 x δ) as shown in part (4). The β- and γ-joints move the leg down until it touches the ground and finishes its swing phase.

In this case, the α- and γ-joints contribute to detecting the faced object based on the measurement of the current consumption. Additionally, the step's length is increased by 20° to enable the rear legs to overcome or climb over an encountered object. The predefined thresholds for the α- and β-joints are defined per experiments. They present the maximum current consumption when the robot walks without hitting any object.

5.6.3.3. *Elevator Reflex While Walking on an Inclined Surface.* Figure 5.6.8 shows the experimental results while the robot walks on a downhill surface and its FL leg hits an object. The FL leg is disturbed by touching an obstacle during its swing phase. The current consumption in the α-joint is monitored and analyzed. The bottom of Figure 5.6.8 shows the consumed current in the leg's joints. At data point 1184, the current consumption in the α-joint increases significantly exceeding its threshold. The leg faces an object. The elevator reflex is triggered to correct the leg trajectory. The leg overcomes the faced object and the robot continues its forward walking on the inclined surface.

Pitch and yaw signals are obtained from the reference sensor and depicted at the top of Figure 5.6.8. The pitch signal presents the inclination in the walking direction (x), while the yaw signal presents the

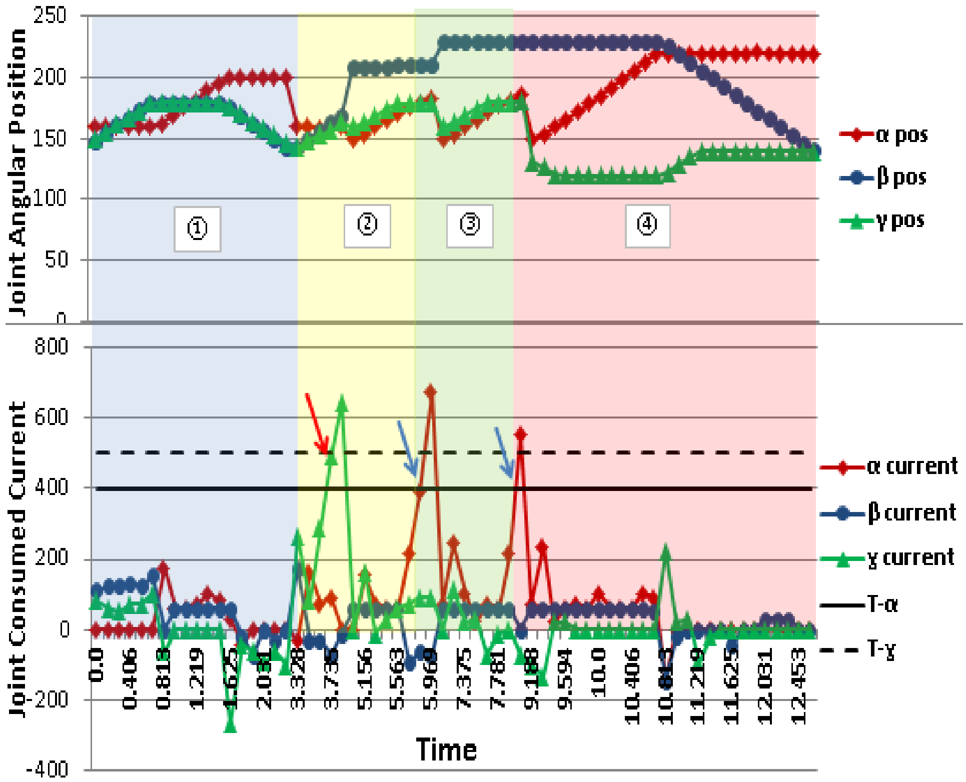


FIGURE 5.6.7. α-, β-, and γ-joint angular position in degree and current consumption in % over time in seconds showing elevation and extension reflexes triggered in quick succession by the HR leg.

inclination in the lateral direction (y). They show that the walking robot is not disturbed by the elevator reflex and the static stable walking is not affected.

Additionally, in this case, the front left leg has the ability to achieve the first and second elevator reflex and, also, the extension reflex due to its sufficient extension while walking on a downhill surface. But in case of walking on an uphill surface, this leg has the ability to execute just the second elevator reflex and the extension reflex due to its sufficient flexion.

5.6.4. Leg Amputation.

To evaluate the effect of leg amputation on the robot walking on an inclined surface, two experiments have been conducted. These two experiments show the effect of losing the middle left (ML) leg while the robot walks on uphill and downhill surfaces.

5.6.4.1. *Walking on an Uphill Surface with Amputated Leg.* In case the robot walks uphill and no more signal is received from the ML leg, the robot folds the affected leg by moving it on the top of the body

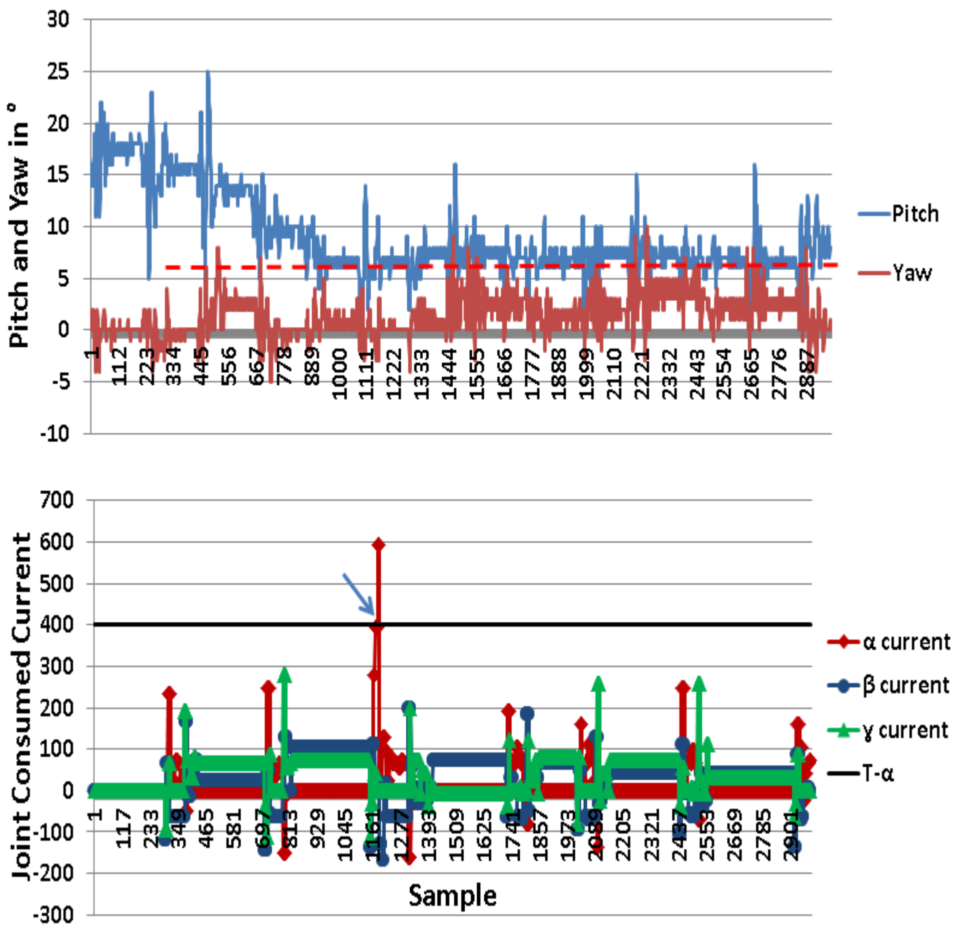


FIGURE 5.6.8. Top: corresponding pitch and yaw readings from the reference sensor while the robot walks on downhill surface. Bottom: α-, β-, γ- current consumption for the FL leg during just the swing phase. The FL leg hits an object and executes the elevator reflex .

and the robot continues its walking with the amputated leg. Due to the leg amputation, a big gap between the legs emerges. To reduce this gap, the robot reconfigures its legs depending on the direction of the inclination and the position of the amputated leg. In case of uphill walking, the neighboring legs of the amputated leg shift their PEP and AEP toward the amputated leg to balance the robot weight during the uphill walking. Additionally, the right rear leg shifts its PEP and AEP in clockwise direction to enable the robot to walk stable while walking uphill and to reduce the robot's tilting. Figure 5.6.9 shows the experimental results. The top of Figure presents the signal obtained from the reference inclination sensor. The bottom diagram shows the

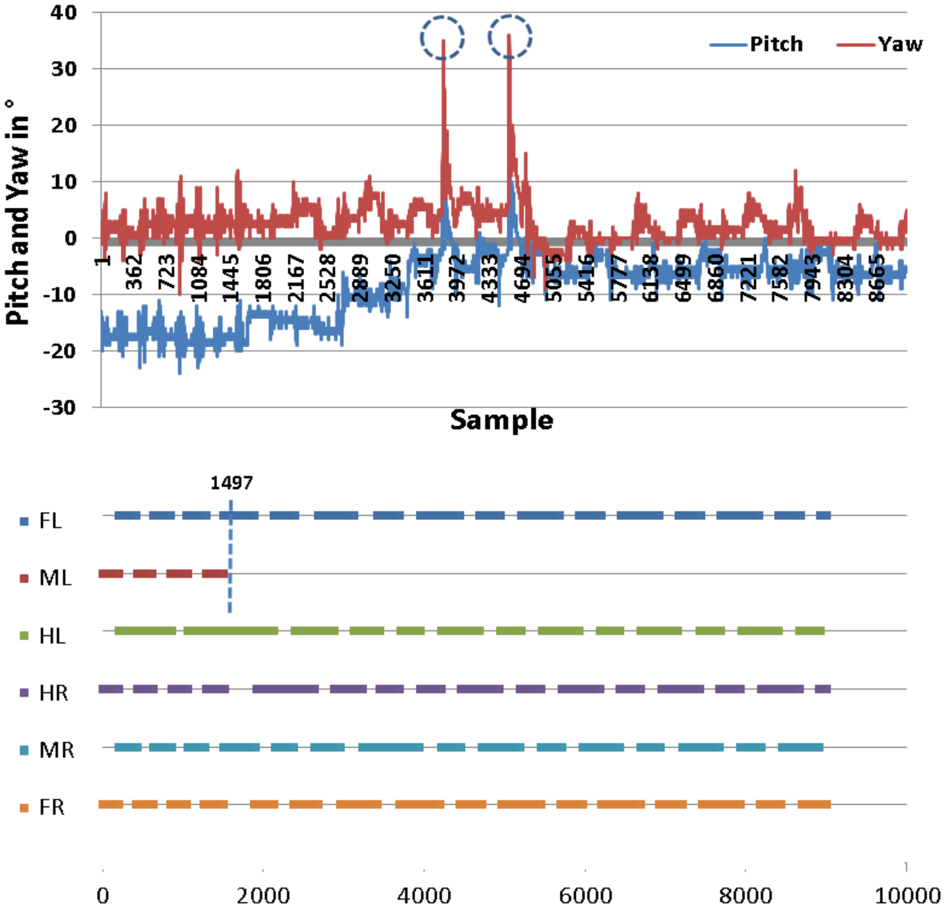


FIGURE 5.6.9. Correction of an uphill inclination with amputated leg (top). Leg amputation is at data time 1497. The bottom diagram is the step pattern with amputated ML leg. Colored bars present the stance phases and the spaces between them are the swing phases.

step pattern and the time of amputation. The middle left (ML) leg is amputated at data time 1497. As shown in the top diagram, the pitch value of the reference sensor indicates that the robot detects the direction of inclination and adjusts its body posture to -6 after the leg amputation. However, the robot tilts twice briefly forward as a result of the loss of one of its middle legs (see circles on the top of Figure 5.6.9). Nevertheless, the robot continues its forward walking on the uphill surface maintaining its statically stable walking despite of the loss of one of its legs.

5.6.4.2. *Walking on a Downhill Surface with Amputated Leg.* The same experiment has been conducted, but here the robot walks on a downhill surface. In this case, the legs' reconfiguration is changed slightly because the direction of the inclination is changed. As soon as the ML leg is amputated, the neighboring legs support the the amputated leg by shifting their position toward the amputated leg. Additionally, the FR leg shifts its PEP and AEP in counter-clockwise direction to ensure stable walking while the robot walks downhill and to prevent the robot from a probable forward overturn. Figure 5.6.10 shows the signals obtained from the reference tilt sensor and the step pattern while walking down. The ML leg is amputated at data time 1436. As shown at the top of Figure 5.6.10, the robot detects successfully the direction of inclination and adjusts its body posture. The robot continues its forward walking on the inclined surface with the amputated leg, but it tilts slightly four times without losing its statically stable walking.

As a result of the leg amputation, the robot walks in a curve. The reason is: One side has two legs and the other one has three legs. The side with three legs is more effective than the other side in pushing the robot forward.

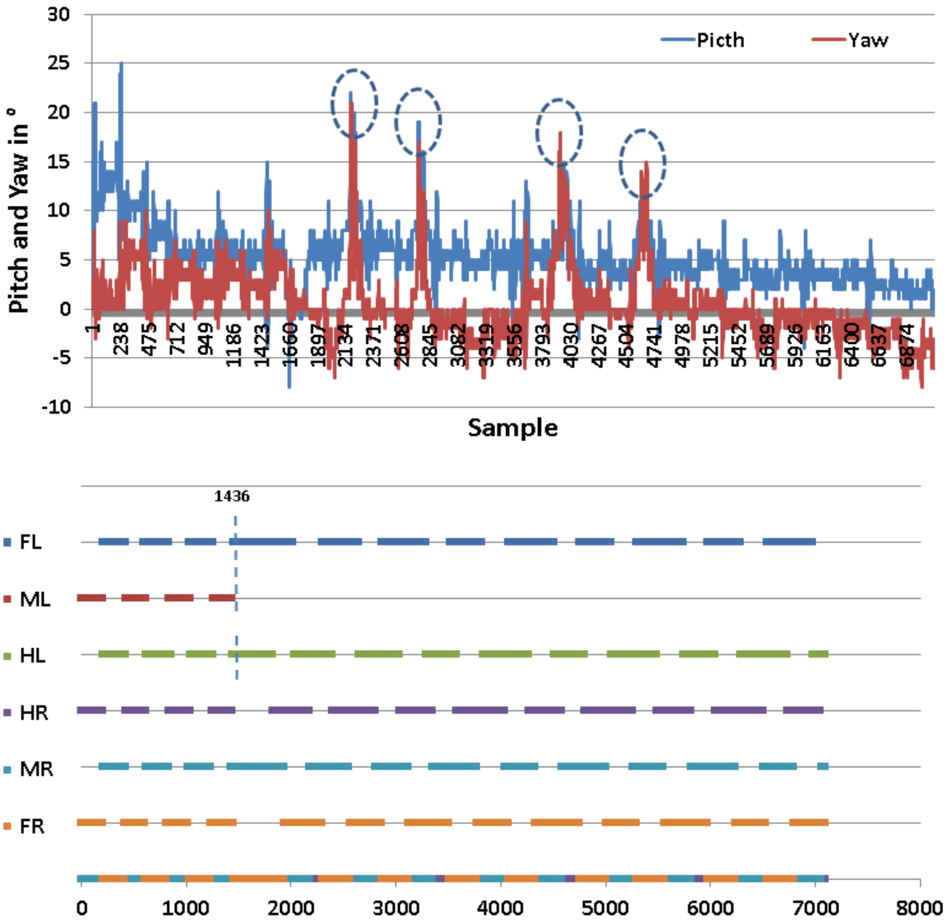


FIGURE 5.6.10. Correction of downhill inclination with amputated leg. Leg amputation is at data time 1436 for downhill walking. The bottom diagram is the step pattern with amputated ML leg. Colored bars present the stance phases and the spaces between them are the swing phases.

5.7. Conclusion

The introduced approach has enabled the robot to detect the direction of inclination and to adjust the robot body posture according to the direction of inclination and the degree of the inclination. Several experiments have been conducted to evaluate the used approach. This approach is focused on the most important critical situations that the robot faces while it walks on rough terrain. These two critical situations include walking on downhill and uphill surfaces. Ground inclination is detected by monitoring the current consumption in the distal joint (γ-joint) in the front and rear legs during the stance phase. The presented approach enables the robot to adapt its body posture and to continue its mission without using a tilt sensor.

For walking on an uphill surface, the front legs flex more and the hind legs expand more to adjust the robot's body posture and to maintain the statically stable walking. For walking on a downhill surface, the front legs expand more and the hind legs flex more. Additionally, the effect of leg disturbances on the robot walking, i.e. small obstacles and leg failure, has been studied in the context of this chapter. The elevator reflex is triggered as soon as the robot's leg hits an object hindering it to reach its required position during the swing phase. The faced object is detected by monitoring the current consumption in the α-joint for the front and middle legs and in the α- or γ-joint for the rear legs.

Finally, the leg amputation has been demonstrated in this chapter. The effect of the amputation of the ML leg has been discussed and presented. The initial results show that our approach also works under such disturbance.

In the next chapter, another approach will be discussed. It shows how our OSCAR detects slippery surfaces and achieves backward walking as soon as it faces such terrain.

CHAPTER 6

Slippery Ground Detection and Backward Walking

6.1. Introduction

Legged insects have the ability to sense the walking terrain. Furthermore, they are able to adapt their walking gait pattern quickly to overcome faced problems that hinder their forward walking. Moreover, they are able to react quickly to unpredicted obstacles such as large gaps by executing a search movement **[51]** or object avoiding by performing the elevator reflex **[116]**. In some critical cases, they resort to the escape behavior by executing backward walking to avoid the encountered obstacle through rapid turning **[57]**. Additionally, several researches have proved that a six-legged insect has the ability to traverse a slippery surface and to adapt its step gait pattern on it **[66, 55, 68]**. Based on Epstein and Graham experiments that have been conducted on surface smeared with oil, it has been observed that the insect's step pattern can be adapted while walking on such terrain according to the degree of the oil viscosity. If the oil viscosity is low, the insect walks with a tripod gait or it achieves a tetrapod gait in case the oil viscosity is high **[55]**. Additionally, it has been observed that its legs achieve significantly different velocities on slippery surfaces, and the legs do not retract at the same rate while walking on the slippery surface **[55]**. Furthermore, the insects' legs that are on the slippery substrate operate with a reduced power output while walking **[55]**.

This observed behavior of the legged insects makes the robotic engineers willing to benefit from this biological behavior to cope with the highly challenging task of detecting and performing adaptive walking.

This chapter will cope with this challenging task and presents our decentralized controller approach for detecting and overcoming the faced slippery surface.

But before describing our used approach, some related works that have been conducted on legged robots will be demonstrated.

Some related researches on the bipedal robot have been carried out to achieve stable walking on slippery ground **[25, 152]**. In **[152]**, an approach to detect slippery surfaces was introduced and the used method to continue the walking on slippery ground was also demonstrated. The introduced approach is based on the use of acceleration sensors which are fixed on each foot of the biped robot. If the biped's leg achieves the stance phase and the ground is not slippery, the acceleration signal is zero. In case the walking surface is slippery, the signal of the acceleration sensor is higher than zero.

Other research has been conducted on the quadruped robot such as TITAN-VIII [**142**]. H. Takemura el at have found a method to detect the slippery surface. They proposed two new strategies to achieve slip-adaptive walking on a slippery surface, the slippery surface is detected depending on a method similar to the one mentioned above in the bipedal robot by using acceleration sensors attached on the top of each foot.

The proposed two new strategies are: a slip-adaptive walk based on slip reflex and a slip-adaptive walk based on force control in the slip leg. The first one modifies the walking pattern via a Central Pattern Generator (CPG). The CPG changes the walking parameters (stride and walking cycle). The second method is based on a force control in the slip leg by adding more force to the slip leg for a short time.

As noted in the previously conducted works, all the mentioned works are based on the use of additional sensors for detecting the type of walking surface, such as acceleration sensors for detecting slippery surfaces.

This chapter will shed light on our applied decentralized controller approach for detecting slippery surfaces and also present the proposed strategy to overcome this challenge. This work is based on our work presented in [**6**]. The novelty of our approach is based on the estimation of the local current consumption in the coxal-joint (α-joint) of each robot's leg during the stance phase. Once the robot detects a slippery surface, a backward walking is proposed as a reflex reaction to prevent the robot from walking on the slippery surface. This work has been implemented in the ORCA architecture and tested on the hexapod robot OSCAR based on the low-cost Bioloid robot. The following Figure 6.1.1 shows the OSCAR robot while traversing from a rough surface to a slippery surface.

6.2. ORCA Implementation for Slippery Ground Detection

Our approach for detecting a slippery surface has been implemented in the ORCA architecture. In this work, the OCU of each leg has the same structure and the same functionality. Each OCU unit for each leg monitors the current consumption in the α-joint while the leg achieves the stance phase. The leg's OCU does not react unless the α-current consumption shows unusual signals during the stance phase. As soon as the α-current consumption drops under the predefined threshold during the entire stance phase, the OCU of the considered leg intervenes. It sends a signal to the higher level of the system. The higher level of the system does not react unless it receives more signals from other leg's OCUs that also detect the slippery surface insuring that the robot

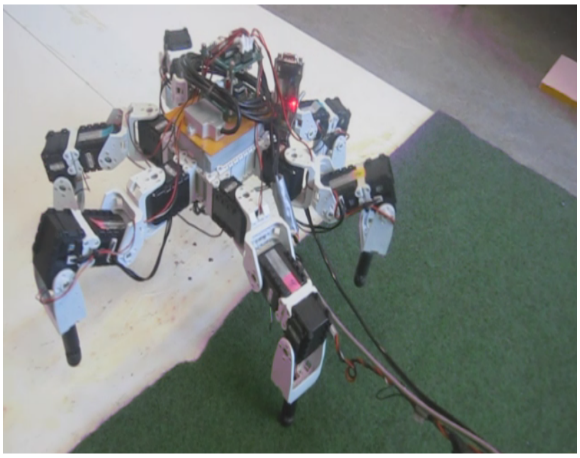


FIGURE 6.1.1. OSCAR walks on slippery surface. Green part represents the rough terrain, the white part represents the slippery surface

walks really on the slippery surface. The higher level decides when the leg's OCUs have to change the BCU's parameters of the considered leg. As soon as the higher system decides that the robot faces a slippery surface, each OCU unit of each leg changes the PEP and AEP of the leg's BCU to achieve backward walking. This procedure leads the robot to change its direction from forward to backward. Figure 6.2.1 shows the ORCA architecture for one leg.

The following section discusses our approach for slippery ground detection based on the analysis of the current consumption in the α-servo motor.

6.3. Slippery Ground Detection

A commonly used method for detecting a slippery ground is using accelerometer sensors that are attached to the top of each robot's foot. The disadvantage of adding these sensors is an increase in the cost of the robot, its fault probability and that the control system becomes more complicated. Therefore, our introduced approach will show how the robot can detect a slippery ground without adding more extra sensors based on estimating the consumed current in each α-servo motor while the leg executes the stance phase.

Generally, if the robot walks on rough terrain, the friction force of the supporting foot with the walking surface is efficient while the leg achieves the stance phase. In this case, the supporting foot is still static during the stance phase. An efficient friction with the substrate produces a large driving force in the supporting leg, which helps to push the robot body forward. On the contrary, in case of low friction between the supporting foot and the substrate, a small driving force

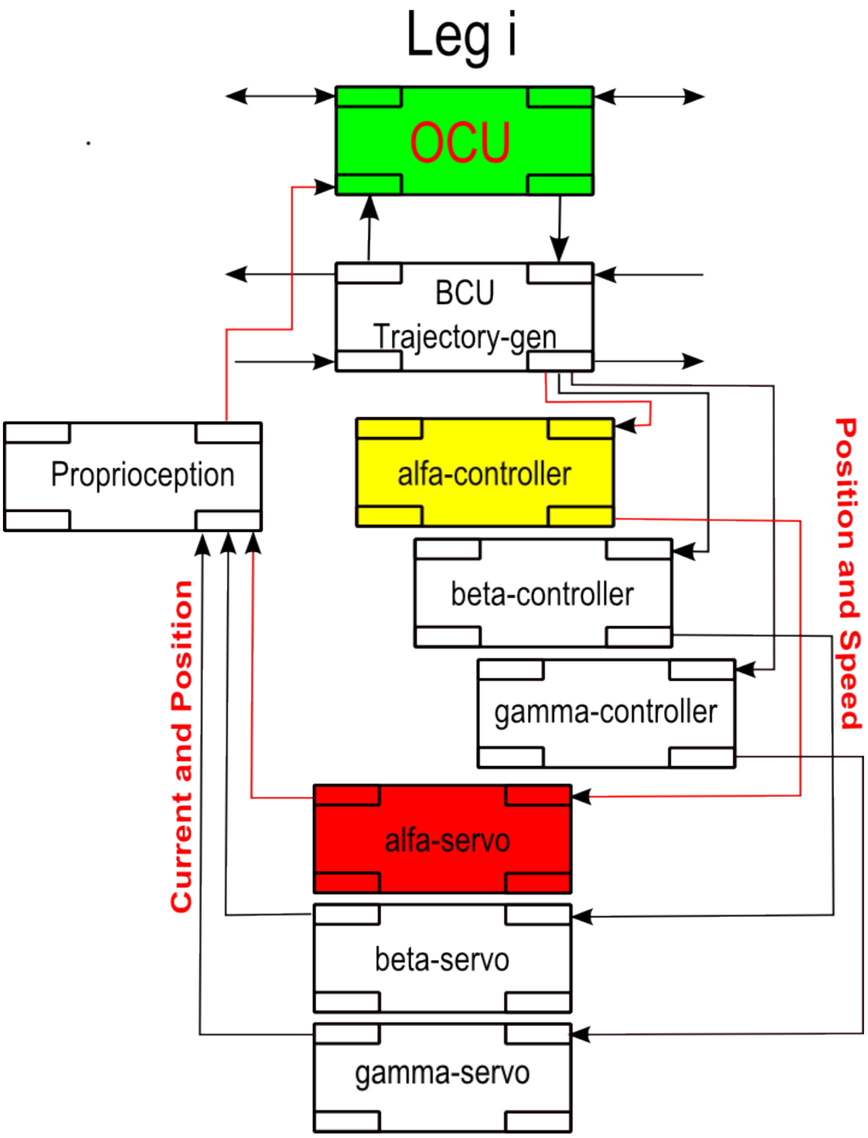


FIGURE 6.2.1. Leg control with ORCA, OCU monitors the current consumption of α-servo and changes the parameters of α-controller.

in the supporting leg is produced. The supporting foot slips on the walking surface.

However, walking on rough terrain that has a high friction coefficient causes a high current consumption in the considered servo motors of the supporting legs. This is due to the large produced driving force. If the friction level is low, the contrary is the case.

Our introduced approach in this chapter depends on the proprioception sensing by measuring the consumed current in each α-servo motor while executing the stance phase. Due to the difference in the friction coefficient between rough and slippery surface, the consumed current in the α-servo motor of the supporting leg is also different. While walking on rough terrain, the consumed current in the α-servo motor during the stance phase is higher than while walking on a slippery surface.

Each leg's OCU monitors the current consumption in the α-servo motor during the stance phase. Once the current consumption in the α-servo motor drops under the predefined threshold, the OCU sends a signal to the higher system level that this leg is slipping. But the higher system level does not react unless it receives more signals from the other legs insuring that the robot faces a slippery ground. The following conditions clarifies how one leg detects a slippery surface. The slippery ground detection for a single leg $D^L_{slip}(t)$ can now be defined as:

$$D^L_{slip}(t) = \begin{cases} true & \text{if } I^L_\alpha(t) < T_{slip} \\ false & \text{else} \end{cases}$$

Here, the detection evaluates to $true$, if slippery ground is detected. $I^L_\alpha(t)$ presents the current consumption in the α-joint while the leg achieves the stance phase. T_{slip} presents the minimum current consumption for α-joint during the stance phase while walking on a rough surface. This threshold is defined experimentally.

However, one leg is not sufficient to define that the robot faces slippery ground. The front and middle legs have to detect the slippery surface until the robot system decides that OSCAR really faces slippery ground. The slippery ground detection for the robot can be defined as:

$$D_{slip}(t) = \begin{cases} true & \text{if } D^{FL}_{slip}(t) \text{ and } D^{FR}_{slip}(t) \text{ and } D^{ML}_{slip}(t) \\ & \text{and } D^{MR}_{slip}(t) \\ false & \text{else} \end{cases}$$

The following section discusses how our hexapod robot OSCAR reacts to the detected slippery surface.

6.4. Reaction to Slippery Ground Detection (Backward Walking)

As soon as our hexapod robot detects a slippery surface, it immediately reacts by reversing its forward walking into backward walking. The OCU unit in each leg exchanges the leg's extreme positions in each leg's BCU. The previous PEP becomes the new AEP and the previous AEP becomes the new PEP. In this way, the robot reverses its forward walking into backward walking.

Figure 6.4.1 shows how the leg's extreme positions will be switched to change the robot walking direction.

In this approach, just the front and the middle legs touch and detect the slippery surface. The legs that touch the slippery surface lose

FIGURE 6.4.1. The direction of leg trajectory during the swing and the stance phase in case of forward and backward walking. The red curve represents the leg trajectory in forward direction. The blue curve represents the leg trajectory in backward direction.

their driving forces. While the rear legs are still on the rough surface, they are retaining the normal driving forces. This has a big advantage if the robot's walking direction is reversed from forward to backward as soon as the robot detects the slippery surface. The rear legs help the robot to achieve backward walking and to move the robot to the rough surface again. After the retreat to the back, if all legs traverse the rough terrain, the robot system changes the walking path. This reaction prevents the robot from walking on a slippery surface that maybe leads to unpredictable results and stops the robot from continuing its mission.

The introduced approach is tested and evaluated on our hexapod robot OSCAR. The following section will show the experimental results.

6.5. Experimental Results

To evaluate our previously mentioned approach, two experiments have been conducted. In the first one, the robot walks on a slippery surface and then on a rough surface. In this experiment, the α-current consumption is monitored and recorded. The results show the difference in the current consumption in the α-servo motor while walking on rough and slippery surfaces. The second experiment will show how OSCAR detects a slippery surface and reverses its forward walking into backward walking and it also shows the step gait pattern.

6.5.1. Walking on a Rough and Slippery Surface.

In this experiment, the test ground is a board of 200x100 cm in an indoor area. This board is divided into two parts. The first part is covered with rough carpet to simulate the rough ground. The second part is smeared with oil to simulate slippery ground.

At first, the robot walks on the rough carpet and then on the slippery ground. The consumed current during the stance phase in each α-servo motor is monitored.

Figure 6.5.1 shows a comparison between the α-current consumption in both cases, rough and slippery surface. The presented results in this Figure are the α-current consumption during a number of successive stance phases.

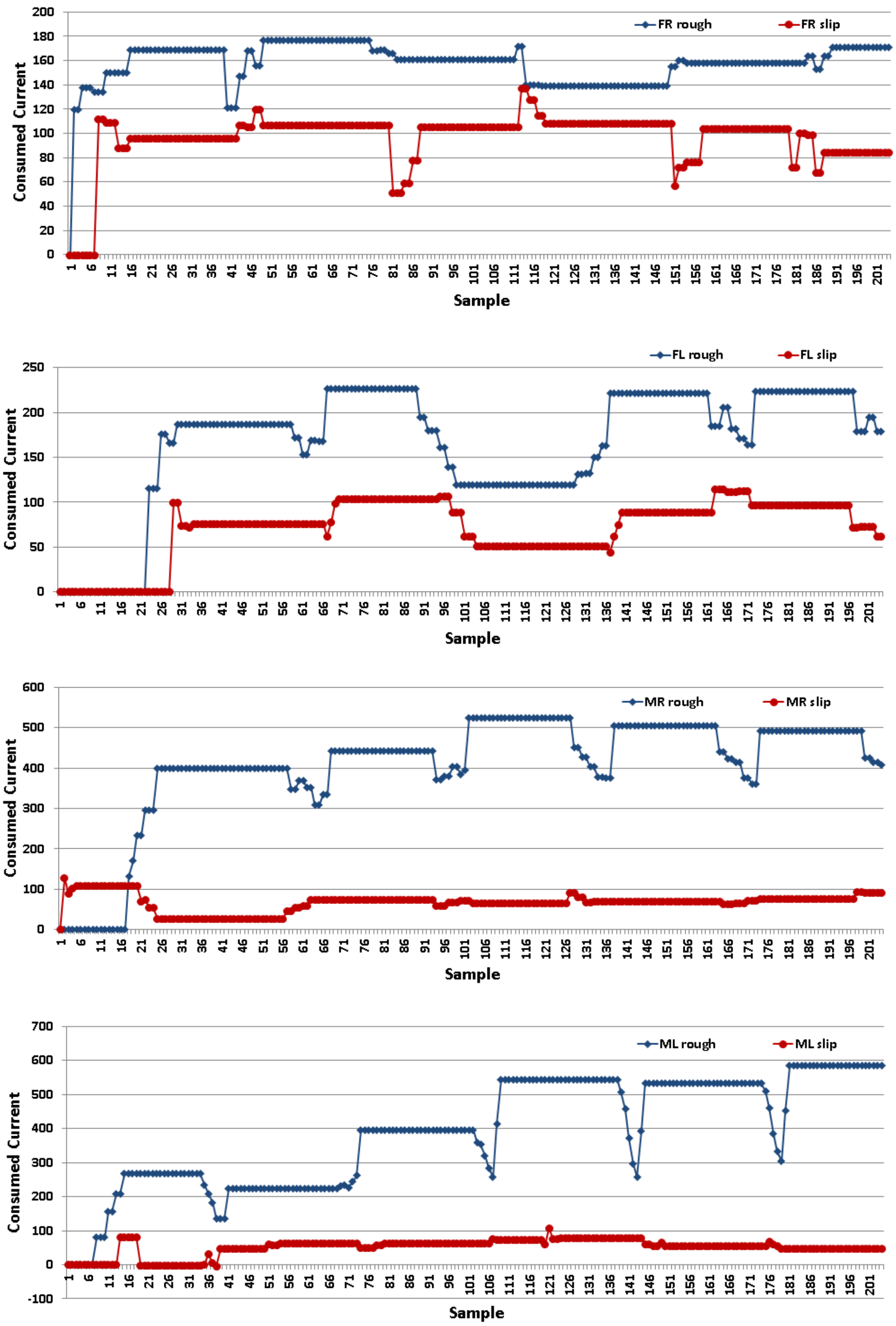


FIGURE 6.5.1. Current consumption in α-servo motor during walking on rough and slippery surface. The above curve presents walking on a rough surface and the low curve presents walking on slippery surface.

As noted in Figure 6.5.1, it is clear that the α-current consumption while walking on the rough ground is higher than the consumed current while walking on the slippery ground.

6.5.2. Slippery Ground Detection and Reaction.

The second experiment shows how our low-cost robot OSCAR detects the slippery ground and reverses its direction immediately from forward walking to backward walking as soon as the robot detects the slippery surface. The test ground of this experiment has the same establishment that is mentioned in the above experiment.

At first, the robot walks on a rough carpet and then on slippery ground. The α-current consumption is monitored while the legs execute the stance phase. The α-current consumption is compared with a predefined threshold that is defined in the first experiment.

Figure 6.5.2 shows the current consumption during the stance phases in the front and middle legs while the robot traverses on rough and slippery surfaces. The depicted curves in this Figure present a number of successive stance phases for the front and middle legs.

When the first leg touches the slippery surface, the FR leg is at data-point 119, its current consumption (red) drops under the predefined threshold, the current consumption remains below the predefined threshold while the leg moves on slippery surface. The leg's OCU sends a signal to the higher level to inform it that this leg faces a slippery surface. The robot continues its forward walking.

The FL leg touches a slippery ground at data point 160, its current consumption (blue) drops under the predefined threshold, the current consumption remains below the predefined threshold while the leg touches the slippery surface. Also, the leg's OCU sends a signal to the higher level that this leg also faces slippery surface. Now both front legs touch the slippery ground and the middle and rear legs are on rough ground. The robot is still continuing its forward walking. At data point 206, the MR leg detects the slippery surface and its α-current consumption (yellow) drops under the predefined threshold. Its OCU unit sends a signal to the higher level. The current consumption (green) in the ML leg drops under the predefined threshold at data point 244. At this moment, all front and middle legs touch the slippery surface and also their consumed currents drop under the threshold. Now, the higher level has received four signals from the front and middle legs ensuring that the robot really faces slippery ground. Now, the higher level of the control system decides that the walking surface is slippery and the robot has to react immediately. The robot changes its current walking direction and achieves backward walking. Since the slipped legs have lost their driving force, the hind legs that did not touch the slippery surface still retain their driving force. The rear legs help the robot to execute backward walking and return it to the rough ground again.

Figure 6.5.2 shows how the α-current consumption in the front and middle legs increases anew due to executing backward walking when the robot walks again on the rough ground. Figure 6.5.2 shows how the current consumption in the α-servo motors for the front and middle legs rises again at data point 274 and beyond while the robot achieves backward walking. The robot traverses again to the rough surface. As also noted, the consumed current in the middle legs exceed the predefined threshold firstly, then the current consumption in the front legs. The reason for this is the backward walking. The robot changes the order of its legs. The hind legs become front legs and vice versa. The middle legs remain unchanged. Therefore, the middle legs traverse the rough ground before the front legs.

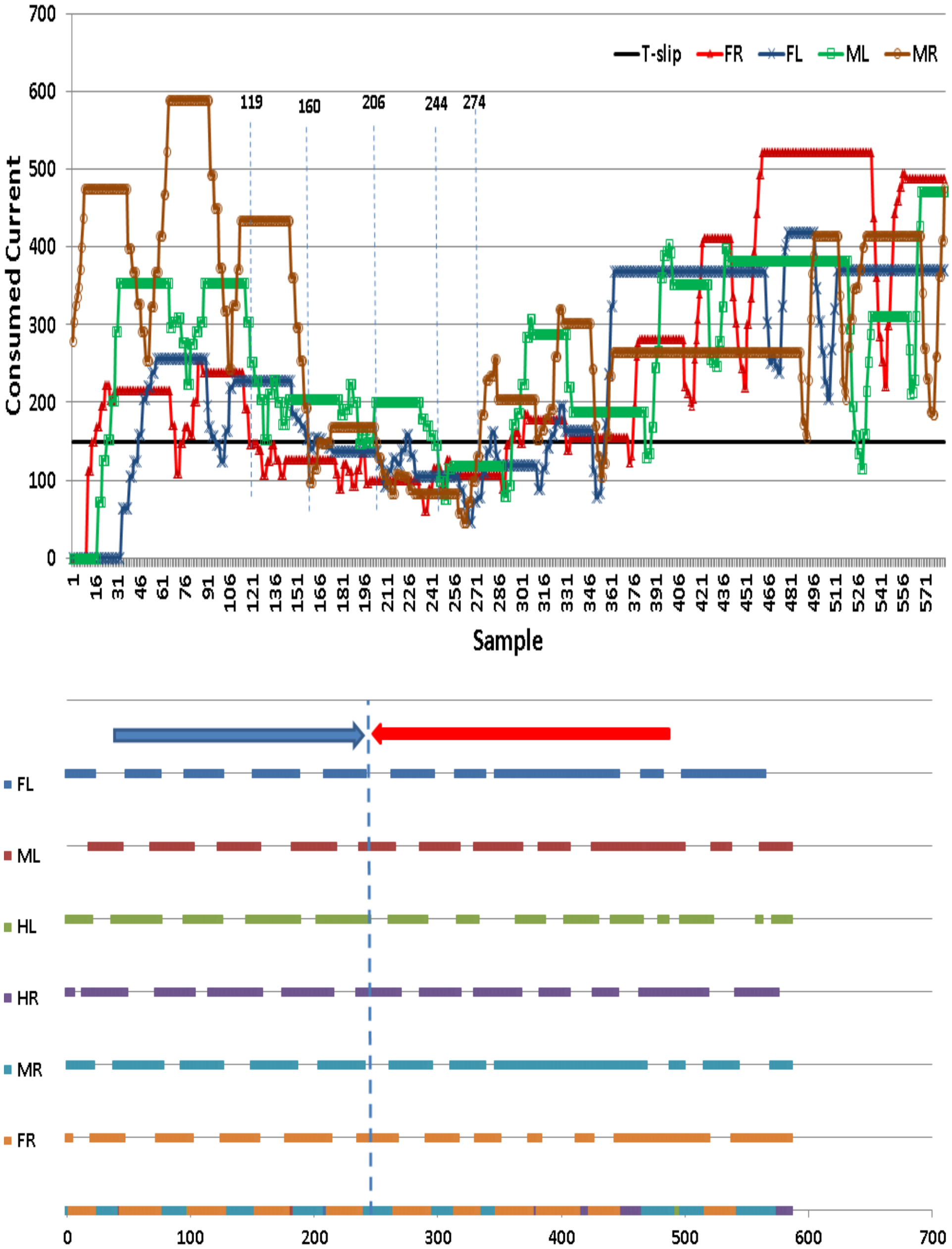


FIGURE 6.5.2. α-current consumption in front and middle legs during the stance phases while the robot traverses from rough to slippery ground (Top). These curves present just the obtained samples of the current consumption during the stance phases. The bottom diagram shows the step gait pattern during robot walking on rough and slippery ground. The colored bars present the stance phases and the space between them presents the swing phases.

6.6. Conclusion

This chapter demonstrated the used approach for slippery ground detection. This approach is based on monitoring the consumed current in the coxal joint (α-joint) for the front and middle legs during the stance phase. The robot reverses its walking direction as soon as the current consumption in the front and middle legs drops below the predefined threshold.

This chapter demonstrated two experiments to evaluate our introduced approach. The first one shows the difference between the current consumptions while walking on rough and slippery surface. The second one shows how our hexapod robot detects a slippery surface by monitoring and analyzing the consumed current in the α-servo motor during the stance phase. The robot reacts to the detected slippery surface by executing backward walking. This reaction prevents the robot from walking on a slippery surface that may result in the termination of the forward walking or to the failure of the robot mission.

This approach is based only on current consumption analyzing without using any external sensor that increases the cost of the robot and makes the control system more complex. Moreover, this approach can be used as alternative sensing in the event of a malfunction of the used hardware sensors to enable the robot to define the type of the walking substrate.

CHAPTER 7

Sandy Ground Detection and Adaptive Walking

7.1. Introduction

In this chapter, we demonstrate our approach for sandy ground detection and adaptive walking on sandy ground. Walking on sandy ground is one of the most important and difficult tasks for all kinds of robots. But, legged robots are still more efficient than other robots when they walk on a sandy surface. They have the ability to change their gait pattern and achieve adaptive walking on such a type of terrain. Firstly, the robot has to define the type of walking surface in order to achieve the proper adaptive walking. Therefore, several researches have been conducted on wheeled and legged robots to enable them to achieve adaptive locomotion on different terrains. Some related researches have been conducted on wheeled vehicles to classify the type of the terrain to be traversed based on the analysis of the vibration of the Z-acceleration signal [**149, 50, 28**]. Weiss et al. [**149**] have presented an approach for classifying the terrain types based on support vector machines (SVM). Their approach is based on using accelerometers to measure the vertical vibration to the ground surface (Z-acceleration). Due to the difference in the terrain types, different vibration signals are produced which are used to enable the learned model to classify the feature of the ground. Another similar work has been conducted on wheeled vehicles introduced by DuPont et al. [**50**]. They introduced an algorithm based on eigenspace and neural networks to classify the terrain. The feature of the terrain is extracted from the vehicle vibration. Another work has been introduced by Brooks and Iagnemma [**28**]. Their approach is based on a linear discriminal analysis that enables the rover vehicle to classify the terrain types online during a traverse. The vibration of the rover structure is measured by an accelerometer. The vibration signals are used in the training phase and online classification.

A close work to our research has been introduced by Lewinger and Quinn [**88**]. They introduced an approach to enable a hexapod robot to achieve emergent gaits based on neurobiological mechanisms. Their work depends on sensory information such as joint angle and joint load to generate emergent gaits and to execute elevator and search reflexes while walking on rough terrain. Their work, however, does not handle complex cases such as walking on sandy or slippery surfaces. Another

FIGURE 7.1.1. OSCAR walks on sandy ground

related work has been introduced by Li et al. [89]. Their SandBot robot has six legs. Each leg is such half-circle similar to the C character. It has the ability to achieve adaptive walking on granular media thanks to the robot speed.
In a work closer to what we report here, Giguere et al. [61] introduced their six-legged amphibious robot which has the ability to identify the type of walking surface based on internal sensor information (vertical acceleration, motor current consumption, and angle of a particular leg). Sensor information is used to let the robot identify the environment depending on a probabilistic Bayesian classifier.

This chapter demonstrates our decentralized controller approach to detect sandy ground by using proprioception sensing. This work is based on our previous work that is introduced in [6]. The novelty of our approach to detect sandy ground is based on monitoring and analyzing the locally consumed current and the angular position in the middle joint of each of OSCAR's leg as proprioceptive sensing. The robot adapts its walking on sandy ground as soon as it is detected. Our approach is also based on an organic computing architecture and was tested on the latest version of the hexapod robot OSCAR, based on the Bioloid robot kit as shown in Figure 7.1.1.

7.2. ORCA Implementation

Our approach to detect sandy ground was implemented in the ORCA architecture. The OCU unit monitors the current consumption and angular position in the β-joint at the end of the swing phase while the leg moves down. In other words, the OCU monitors the current consumption and angular position just when the leg moves down and ignores the rest of the swing phase and the whole stance phase. The OCU unit reacts as soon as it observes that the consumed current and the angular position of the middle joint show a deviation from the predefined threshold. In this work, each leg detects sandy ground independently. As soon as the leg's OCU detects sandy ground, it sends a signal to the higher level of the control system. It decides that the robot faces a sandy surface if it receives at least two signals from the legs that contribute to the detection of sandy ground. The legs that contribute to the detection of sandy ground are the front and the middle legs. As soon as the robot decides that the walking surface is a sandy surface, each leg's OCU changes the parameters of its leg's BCU to enable the robot to achieve adaptive walking on the sandy surface. Figure 7.2.1 shows the ORCA control architecture for OSCAR's leg.

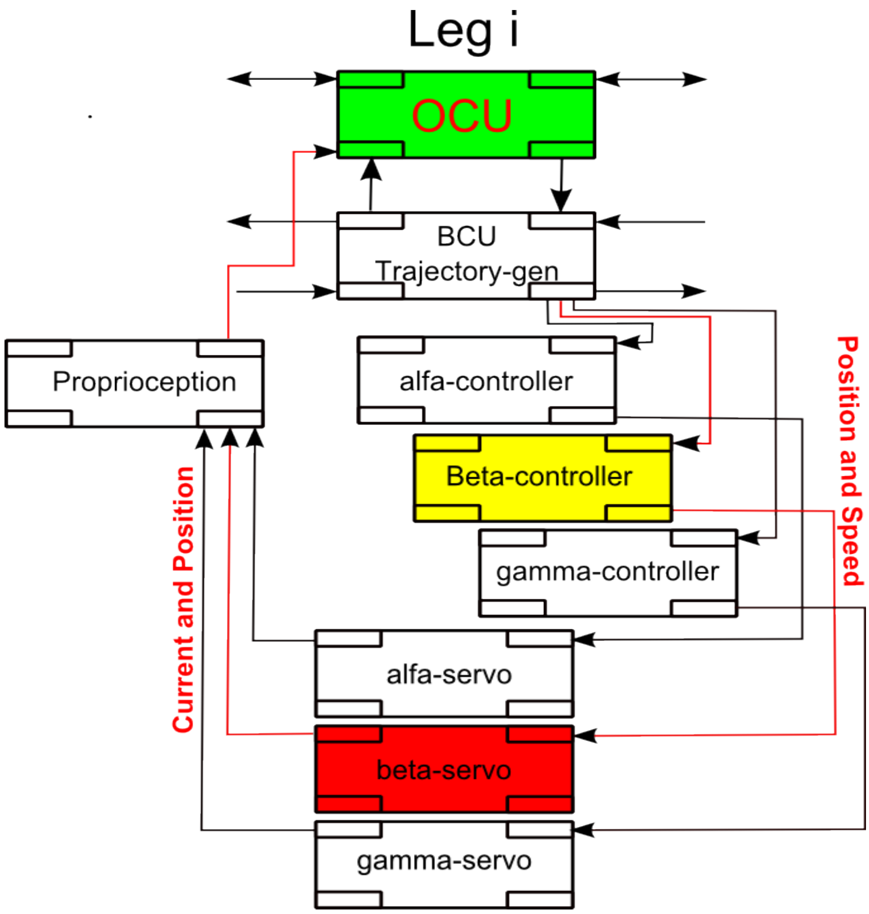


FIGURE 7.2.1. ORCA control architecture for OSCAR's leg (for sandy ground detection). OCU monitors the current consumption of β-servo and changes the parameters of β-controller.

7.3. Sandy Ground Detection

A common method to detect sandy ground is to use force sensors fixed on the bottom of the robot's feet. Our approach introduced here is based on the evaluation of the consumed current and angular position of the middle joint without adding any external sensor. The consumed current is used to identify the stiffness of the walking surface. This approach enables our hexapod robot to identify the type of walking surface.

In case the robot walks on hard ground, the β-current consumption rises significantly exceeding rapidly the lower threshold, T_{min}, as soon as the leg touches the ground. At this moment, the OCU unit records the angular position of the β-servo motor and starts monitoring the angular position changes in the β-servo motor. The leg tries to continue its downward movement and pushes up the robot's entire body upwards if it does not stops moving down after touching the ground. Therefore, the higher threshold, T_{max}, is predefined to prevent the robot's leg from pushing up the robot body. Now, if the consumed current exceeds T_{min} as well as T_{max} and the angular position change ΔP_{β}^{L} does not deviate significantly from D, the robot assumes that it walks on hard ground.

But in case the robot faces soft or sandy ground, the β-consumed current exceeds T_{min} as soon as the leg touches the ground. The current consumption in this case increases gradually due to the leg sinking into sandy ground. Additionally, the angular position deviates significantly from the predefined value D. Now, if the β-current consumption exceeds T_{min} and there are significant changes in the β-angular position, the robot's leg faces sandy ground. The ground type recognizing test reads:

$$D_{sand}^{L}(t) = \begin{cases} true & \text{if } T_{min} < I_{\beta}^{L}(t) \leq T_{max} \text{ and} \\ & \Delta P_{\beta}^{L}(t) \geq D \\ false & \text{else} \end{cases}$$

Here, ΔP_{β}^{L} is the change in the angular position after the consumed current exceeds T_{min}. I_{β} presents the β-current consumption while the leg moves down. T_{min} represents the β-current consumption while moving the leg down without contact to any object. D represents the amount of change in the servo position, after which the consumed current is assumed to rise above T_{max} on hard ground. T_{min}, T_{max}, and D are defined by experiments.

As soon as the leg detects the sandy surface, its OCU unit sends a signal to the higher system that the leg faces a sandy surface. But the higher system level does not react unless it receives at least another signal from other legs that cooperate in the sandy ground detection. The robot detects a sandy surface when at least two legs detect the sandy surface. This procedure ensures that the robot really faces soft or sandy ground. The cooperating legs are the front and middle legs. The robot detects the sandy surface based on the following condition:

$$D_{sand}(t) = \begin{cases} true & \text{if } (D^{FL}_{sand}(t) \text{ and } D^{FR}_{sand}(t)) \text{ or } (D^{FL}_{sand}(t) \text{ and } D^{ML}_{sand}(t)) \text{ or } (D^{FL}_{sand}(t) \\ & \text{and } D^{MR}_{sand}(t)) \text{ or } (D^{FR}_{sand}(t) \text{ and } D^{ML}_{sand}(t)) \text{ or } (D^{FR}_{sand}(t) \text{ and } D^{MR}_{sand}(t)) \\ & \text{or } (D^{ML}_{sand}(t) \text{ and } D^{MR}_{sand}(t)) \\ false & \text{else} \end{cases}$$

7.4. Adaptive Walking on Sandy Ground

The robot adapts its behavior while walking on a sandy surface through changing some parameters. First, the robot decreases its forward walking speed by reducing the speed of all servo motors to 40%. This procedure has big advantages during the swing and the stance phases. In the swing phase, the leg comes into the sand more gently. This obviously reduces the leg penetration and makes the sand beneath the leg compact enough to stop the leg from sinking further into sand. During the stance phase, it makes the sand behind the supporting leg compact enough providing the supporting leg with sufficient driving force that pushes the robot body forward. Secondly, the robot increases the step height of each leg when the leg starts its swing phase. This procedure facilitates the leg‘s movement while walking on a sandy surface, because walking on a sandy surface makes the legs sink into the sand, and furthermore, the surface is uneven due to sandy hills. Another adaptation is decreasing the length of the leg's step. The length of each leg's step is shortend to reduce the motor load during the stance phase while walking on a sandy surface. However, the reduction of the overall servo motor speed as well as the increase of each leg's height and the shortening of the step length lead to a decrease of the robot's forward walking speed. The robot now walks with low speed compared to the walking on flat hard ground. In some cases, the robot walks in a curve due to the low friction between the supporting legs and the sandy surface. Therefore, as soon as the robot detects a sandy surface, the slippery surface detection is deactivated to prevent the system from confusing the sandy and slippery surface because in both cases, the friction coefficient during the stance phase is low. Another additional adaptation will be discussed in the next chapter. It will show how the robot legs adapt their position while walking on complaint surfaces such as sandy or gravel surfaces.

7.5. Experimental Results

To evaluate our introduced approach, two experiments have been conducted. The first one shows the obtained results while walking on hard ground. It shows the current consumption in the middle joint. The second one is conducted while the robot changes from a hard to a sandy surface. It shows how OSCAR detects a sandy surface and reacts immediately by executing adaptive walking on the sandy surface.

7.5.1. Walking on Hard Ground.

In this experiment, the robot walks on hard flat ground. The β-angular position and consumed current are monitored at the end of the swing phase while the leg moves down. Figure 7.5.1 shows the obtained samples of the β-angular position and the β-current consumption at the end of the swing phases, i.e. while the leg moves down. The stance phases are not depicted in this figure, because they are not relevant for sandy ground detection.

As noted from Figure 7.5.1, the β-current consumption does not exceed the low predefined threshold (T_{min}) while walking on hard ground. In some cases, the current consumption exceeds T_{min} as shown in the FR leg at data point 387 and in the ML leg at data point 492. These cases occur when the leg touches the ground at its maximum position. The system does not react unless there are also angular position changes. In this experiment, the leg switches from the swing into the stance phase in both cases. Either the β-current consumption exceeds T_{max} or the leg reaches its maximum position. The β-current consumption in the FL and MR legs is monitored while the β-servo motor moves from position 160° to 120°. In the FR and ML legs, the β-current consumption is monitored while the β-servo motor moves from position 140° to 180° as shown in Figure 7.5.1. Now the robot system does not detect any sandy surface. It assumes that the walking surface is hard ground.

7.5.2. Walking on Sandy Surface.

In this experiment, our hexapod OSCAR walks firstly on hard ground and then it changes over to sandy ground. The setup of the test ground consists of a rough hard board of 200x100 cm. It presents the hard surface. At the end of this board, a sand basin of 200x100 cm is established. The used sand is dry and fine. It is similar to sand that is found in sandy deserts. Figure 7.5.2 shows the obtained results. It shows the β-angular position and current consumption while the robot walks on a hard and then on a sandy surface. The depicted data presents the samples of the β-current consumption and angular position at the end of each swing phase, i.e. while the leg moves down. For more clarification, the consumed current and angular position of the β-servo motor in the stance phase are not depicted in this figure because they are not relevant for the sandy ground detection. As shown in Figure 7.5.2, the β-current consumption rises significantly exceeding its low threshold as soon as the leg touches a sandy surface because the leg sinks into the sandy surface. The leg sinks into the sand unless the current consumption exceeds T_{max}. At the top of Figure 7.5.2, the β-current consumption in the FR leg exceeds the low predefined threshold T_{min}

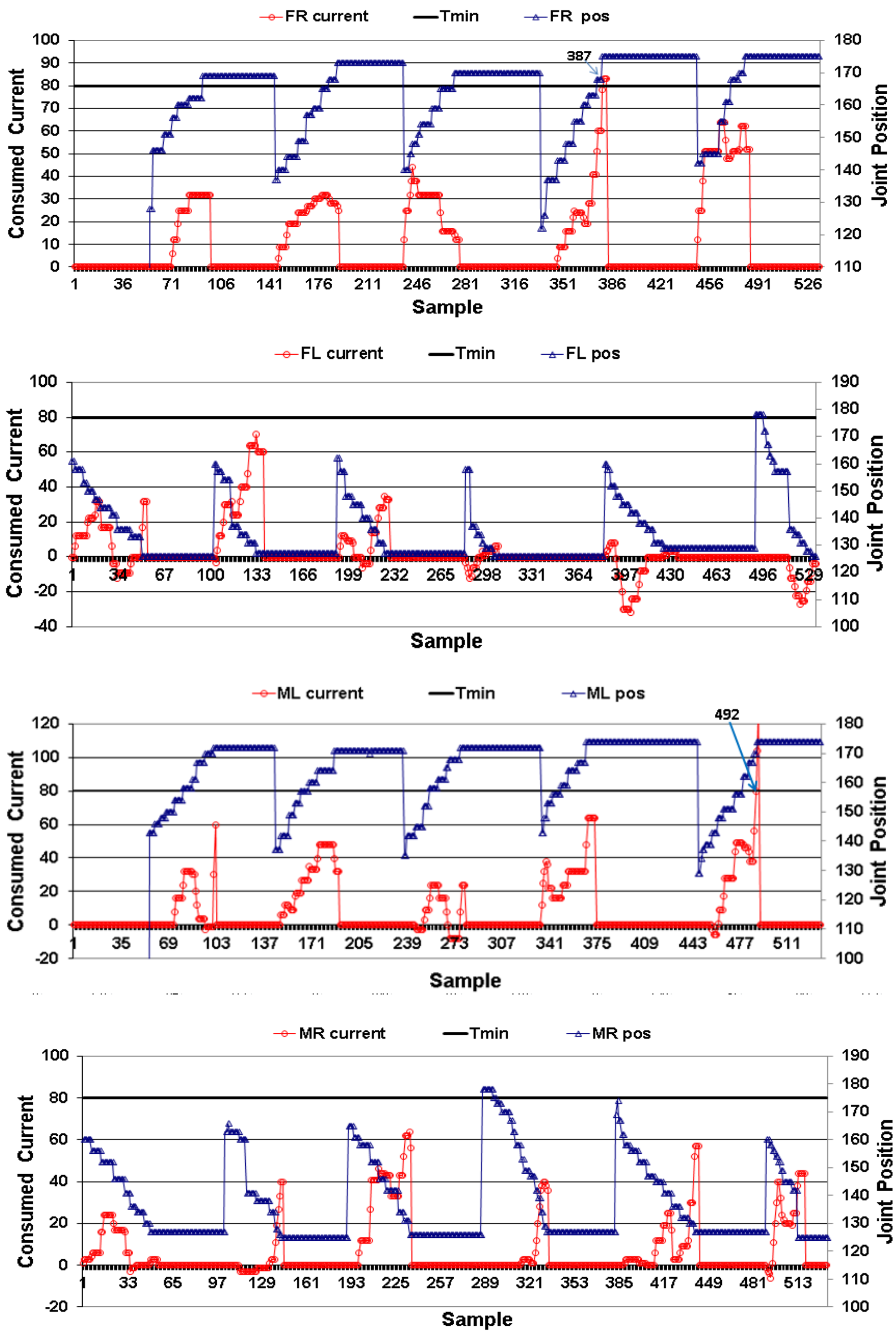


FIGURE 7.5.1. Measured β-angular position and current consumption in the front and the middle legs while walking on hard ground. Each pulse presents the obtained samples at the end of the swing phase while the leg moves down.

at data time 250 and at β-angular position 149° while the leg moves down. The leg continues its downward movment until the β-current consumption exceeds T_{max} or the leg reaches its maximum position. In this case, the β-current consumption exceeds T_{max} at data time 281 and at β-angular position 173° that stops the leg's movement. The leg stops moving down at position 173° before reaching its maximal β-servo motor position 180°. The leg switches from the swing to the stance phase. Here, the FR leg has detected the sandy surface due to the current consumption and angular position changes. The same scenario will be repeated in each of OSCAR's leg as soon as they touch the sandy surface. In the middle of Figure 7.5.2, the β-current consumption in the FL leg exceeds the T_{min} at data time 291 and at angular position 153°. The FL leg touches the ground and continues moving down. It detects the sandy surface at data time 300 and angular position 141°. Now, both FR and FL legs detect sandy ground. At data point 300, the robot decides that the walking surface is a sandy surface and it has to react immediately to the walking surface.

As noted from the curves of the FL leg for walking on a sandy surface, the leg stops moving down before reaching its maximal position although the β-current consumption does not exceed T_{max}. This is due to the robot's adaptation on the sandy surface after the sandy ground detection. The robot tries to adapt its walking behavior on sandy ground.

The robot now decreases the servo motor speed from 80 which presents 54° degrees/second to 50 which presents 30° degrees/second as shown in the green line in Figure 7.5.2. The robot also shortens the length of the steps and increases the step height of each leg. Figure 7.5.2 in the bottom shows how the β-current consumption in all legs increases significantly as soon as the robot faces a sandy surface. The same scenario and results are in the middle and rear legs. Figure 7.5.2 shows that the robot has the ability to walk on sandy surfaces, but at slower speed.

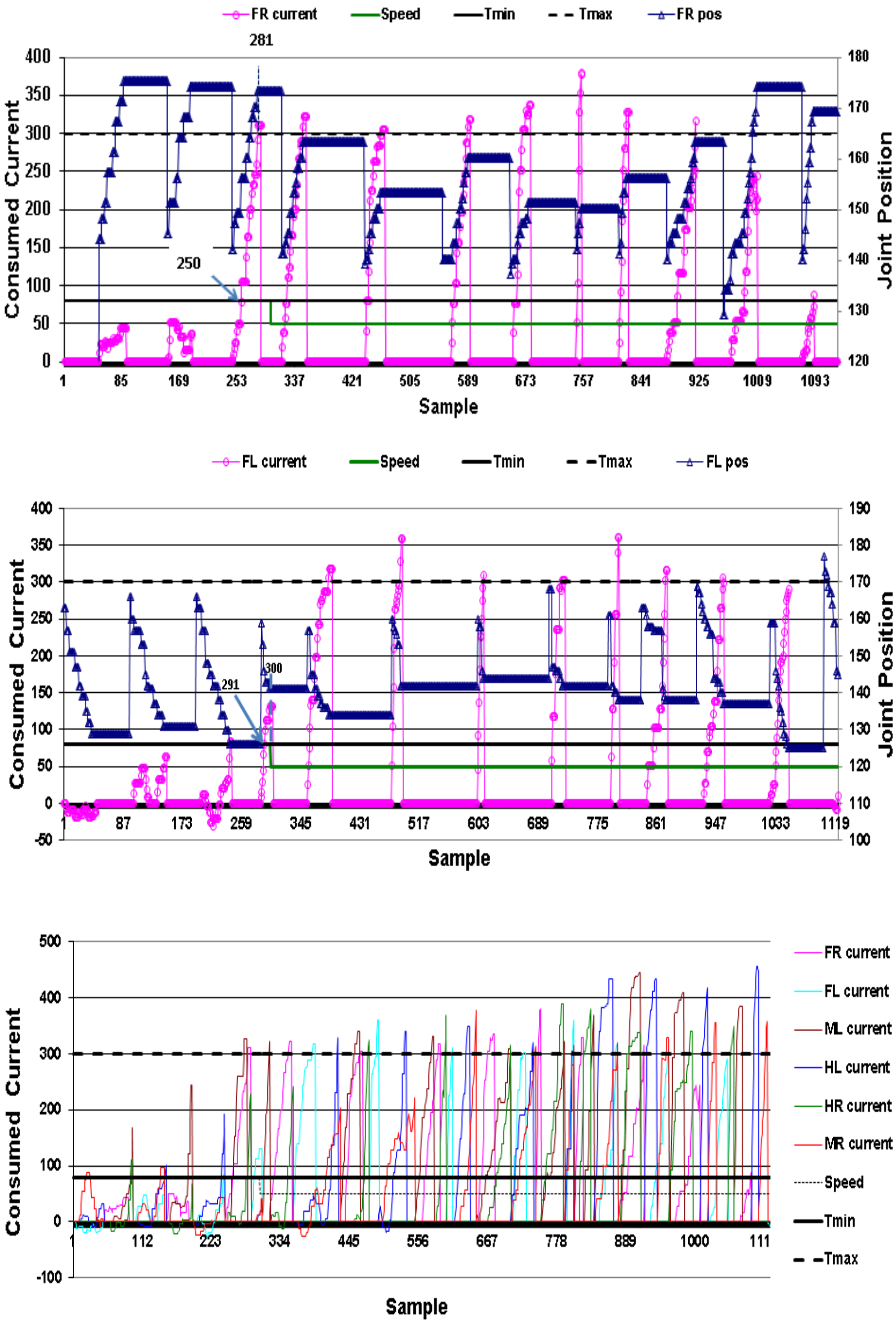


FIGURE 7.5.2. Measured β-angular position and current consumption in front left and right legs while walking on sandy surface. The bottom Figure shows how the current consumption changes in all robot's legs. Each pulse presents the obtained samples at the end of the swing phase while the leg moves down.

As noted from the curves for walking on sandy and hard ground, the β-angular position, while walking on hard ground, reaches its maximal position due to the absence of any obstacle hindering the leg movement. But while walking on the sandy surface, the leg detects or touches the surface earlier. The reason is: the robot legs sink into sand and that leads to decrease the robot body height. Therefore, the robot leg touches the sandy surface early compared with the robot walking on hard ground. The leg stops moving down if the consumed current exceeds T_{max}. The question is how much the angular position has changed after touching the ground. Obviously, as shown in Figure 7.5.2, the angular change in the β-joint after touching sandy ground is bigger than when the same leg touches hard ground.

This legged robot is designed to walk on rough terrain. Therefore, another case has been tested in this work. This case is: The robot leg faces an object beneath it while it is moving down. In this case, the β-current consumption rises rapidly exceeding the T_{max} without significant changes in the angular position. Figure 7.5.3 shows the obtained results when the FR leg faces an object at the end of the swing phase while it moves down. Figure 7.5.3 also shows the β-angular position and current consumption. At data point 276, the β-current consumption suddenly exceeds the high predefined threshold T_{max} without significant changes in the β-servo motor position. The robot system recognizes this case as either hard ground or a transient object under the leg. In this case, the leg stops moving down and begins its next stance phase.

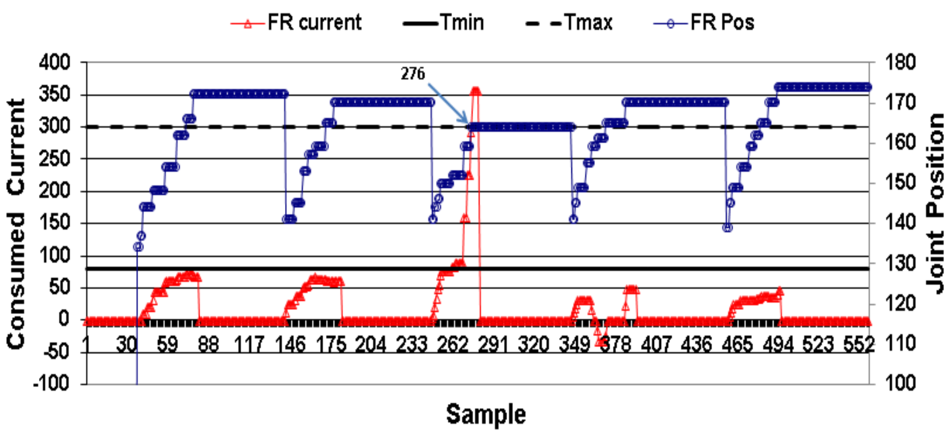


FIGURE 7.5.3. Walking on hard ground with transient object beneath the robot's leg.

7.6. Conclusion

This work has focused on enabling OSCAR to identify the hardness and softness of the walking terrain through monitoring and analyzing the angular position and current consumption in each β-servo motor at the end of the swing phase while the leg moves down. As soon as the robot detects a sandy surface, it adapts its forward walking by changing some parameters. It decreases the overall servo motor speed. This procedure improves the leg's movement in the stance and the swing phases. Additionally, the control system shortens the length and increases the height of the leg's step. This enables the robot to walk on sandy ground better, but it also leads to a very slow walking speed.

Based on our approach, the robot continues its mission without external sensors, which increases the complexity of the control system and the cost of the robot. Moreover, this approach can be used as alternative sensing in the event of a malfunction of the used hardware sensors.

In the next chapter, it will be discussed how the robot's legs adapt their position while walking on soft grounds such as sandy and gravel surfaces. This adaptation is considered a further adaptation that will improve the walking on sandy surfaces.

CHAPTER 8

Adaptive Control of Leg Position

8.1. Introduction

Walking on compliant surfaces such as sandy or gravel ground is considered an important issue. However, walking on sandy or gravel ground makes the robot's legs sink into the ground and walking straight forward becomes very difficult. In this case, the robot's legs have to span a varying distance between the robot's body and the walking substrate while walking on uneven terrain. Additionally, the robot has to coordinate its legs' movements to maintain its stable walking. This behavior is obviously copied from legged insects. Legged insects have the ability to control the movement of their six legs, which has been investigated by Graham and Cruse [**64, 65, 37**]. Also, legged insects have to control their legs positions while walking on uneven terrain, which has been studied intensively by Cruse et al. [**42**].

These biological concepts encourage engineers to transfer them into legged robots to solve the challenging task of controlling the robot's leg position while walking on uneven terrain and maintaining the coordination between the legs.

A related work has been accomplished on the hexapod robot that is introduced by Celaya and Porta [**32**]. Their robot adapts the height of each foot to the elevation of the ground below it based on a force sensor signal. When the force signal drops below a given threshold, the leg is lowered a bit. In a work closer to our work, Palankar and Palmer III [**113**] describe an approach to control the robot's leg position based on a force threshold controller. Their robot's leg adapts its position also based on a force sensor signal.

In this work, an approach will be demonstrated to improve the robot's walking on compliant surfaces. This approach enables each robot's leg to adapt its position to be commensurate with the type of the walking surface while walking on highly and weakly compliant surfaces. When the substrate compliance is high (sandy ground), the leg has to correct its position close to the point of contact with the ground. When the substrate compliance is low (gravel ground), the correction of the leg position is minor.

The other mentioned works mostly rely on additional sensors like force sensors while our approach depends on the evaluation of the consumed current and angular position in the leg's joints as proprioception sensing.

This work is implemented in the ORCA architecture and was tested on the latest version of the hexapod robot OSCAR (see Fig. 8.1.1). This work is based on our previous work that is presented in [**7**].

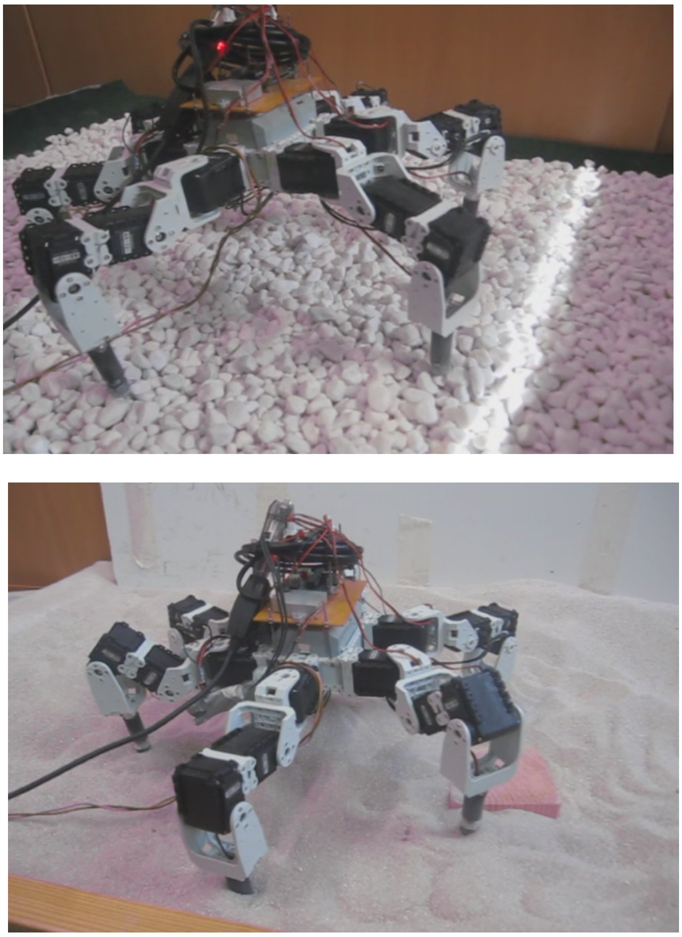


FIGURE 8.1.1. Test platform hexapod robot OSCAR while walking on gravel ground (top) and sandy ground (bottom).

8.2. ORCA Implementation

Our approach to control the leg position was implemented in the ORCA architecture. The entrusted task to the organic control unit (OCU) here has the same function that is entrusted to the OCU unit for sandy ground detection that is discussed in the previous chapter (Chapter 7). But here, there is an additional task. Each leg's OCU unit monitors the current consumption and angular position in the β-joint at the end of the swing phase while the leg moves down. The OCU unit of each leg records the β-angular position when the β-current consumption exceeds the minimum predefined threshold due to touching the ground while the leg moves down. This angular position presents the point of contact with the ground. The OCU unit does not react until the current consumption exceeds the maximum current consumption or the leg reaches its maximum span. As soon as the current consumption in the middle joint (β-joint) exceeds the maximum threshold or the leg reaches its maximum span, the leg stops moving down, then the OCU unit sends a signal to the BCU unit that is responsible for achieving the swing phase enforcing it to correct the leg's position. The BCU unit moves the leg to the point of contact with the ground that is given by the OCU unit. Each leg corrects its position independently from other legs according to the height and type of the walking substrate. The leg stops moving up when it reaches the point of contact with the ground.

8.3. Control of Leg Position

Our previous work that is mentioned in the previous chapter (Chapter 7) was focused on analyzing the current consumption and angular position in the β-servo motor to detect the type of walking terrain. In this section, we will show how the OSCAR robot controls its legs' positions to improve its walking on complaint surfaces.

The used approach to control the leg position is very similar to what has been mentioned in previous Chapter 7. To clarify this approach, the previous approach should briefly be mentioned to show the difference between the two approaches.

In our previous work, two cases have been discussed: the robot walks either on hard ground or soft ground. In case the robot walks on hard ground, the leg defines that the substrate is hard as soon as the current consumption in the β-servo motor rises above the maximum predefined threshold (T_{max}) without significant changes in the angular position of the β-servo motor. While in case of walking on soft ground such as sandy ground, the leg sinks into the sandy surface and the current consumption in the β-servo motor rises gradually exceeding the minimum threshold (T_{min}). The leg stops sinking into the sandy ground as soon as the current consumption in the β-servo motor exceeds the maximum threshold or the leg reaches its maximum span. Then the leg switches into the stance phase without correcting its position. Due to the leg sinking into the sand, the robot faces a great difficulties while walking forward. Therefore, our approach is introduced to cope with this problem.

In this approach, the control of the leg position is conducted while the robot walks on a compliant surface. In this case, the current consumption and the changes of the angular position in the β-servo motor is monitored at the end of the swing phase while the leg moves down. As soon as the β-current consumption exceeds the minimum predefined threshold T_{min}, the OCU records the angular position of the β-servo motor. This position represents the point of contact with the ground. The leg continues moving down until the current consumption in β-joint exceeds the maximum threshold T_{max} or the leg reaches its maximum span. The leg sinks into the ground depending on the nature of the substrate compliance. As soon as the leg stops moving down, the OCU sends a signal to the leg's BCU that adapts the leg's position and returns it to the point of contact with the ground. The leg adapts its position depending on the following conditions:

if $I^L_\beta(t) \leq T_{min}$ **then:** The leg moves down without touching any object and continues its movement down until it reaches its maximal span.

if $T_{min} < I^L_\beta(t) \leq T_{max}$ **then:** The leg touches the ground and the OCU records the β-angular position. The leg still moves down until the β-current consumption exceeds T_{max} or the leg reaches its maximum span. Then the leg has to correct its position. It moves to the point of contact with the ground that is recorded by the OCU unit.

if $T_{max} < I^L_\beta(t)$ **and** $\Delta P^L_\beta(t) < D$ **then:** The leg stops moving down. It detects hard ground and has to move back slightly to the point of contact with the ground to compensate the slight displacement in the angular position.

ΔP^L_β, D, $I^L_\beta(t)$, T_{max} and T_{min} are defined in the previous Chapter 7. They have the same values.

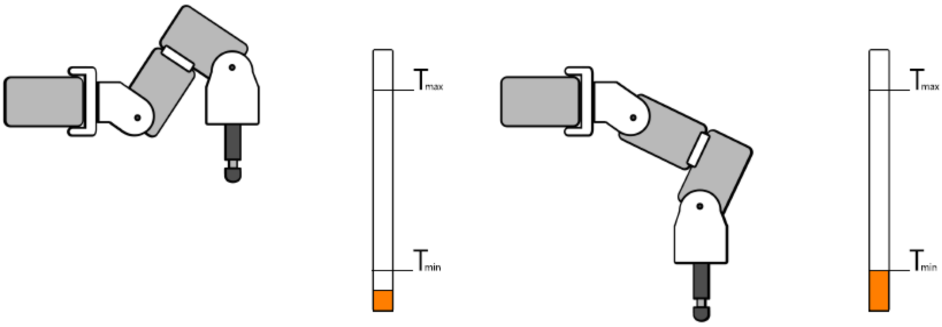
Figure 8.3.1 shows how the current consumption and angular position will change according to the properties of the walking surface.

8.4. Experimental Results

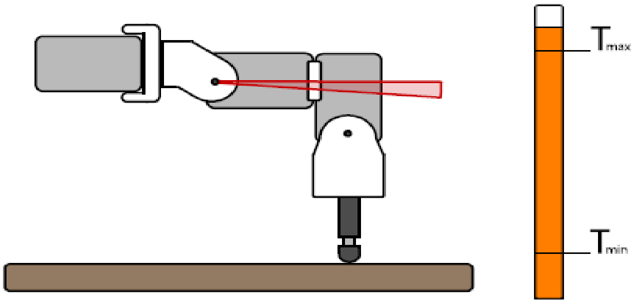
To evaluate our approach, several test grounds have been established. The robot has been tested while walking on gravel and sandy ground. The following experiments show the obtained results when the robot walks on weakly and highly compliant surfaces like gravel and sandy surfaces.

8.4.1. Walking on a Gravel Surface.

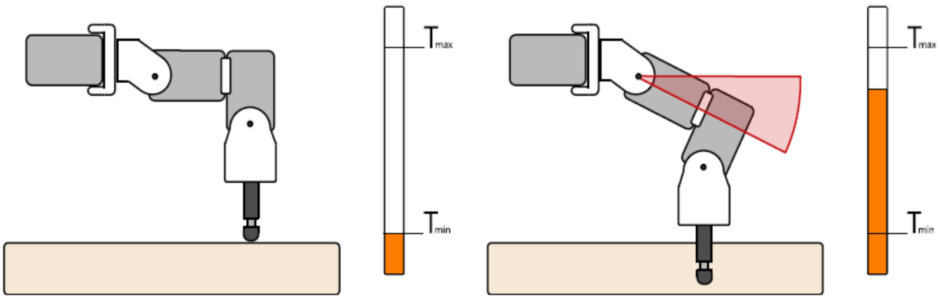
In this experiment, the robot walks on a ground covered with small stones with a diameter of about 1.5 cm. The test ground is an indoor



(A) The robot leg moves down without touching any object. The β-current consumption in this case does not exceed the T_{min}.



(B) The leg touches hard ground. The β-current consumption rises rapidly exceeding its T_{max} with small changes in the β-angular position when the leg touches the hard surface.



(C) The leg touches compliant surface. The β-current consumption rises exceeding T_{min} with significant changes in the β-angular position.

FIGURE 8.3.1. How OSCAR's leg defines the type of the walking substrate.

area of 200x100 cm. Figure 8.4.1 shows the obtained results from the front and middle legs. The depicted curves in this figure present the obtained samples of the current consumption and the angular position of the β-servo motor at the end of the swing phases while the leg moves down. It presents successive swing phases, where each pulse presents the end of the swing phase. The stance phase is not depicted in this figure because it is not relevant for the control of the leg position.

The β-current consumption in the FR and ML legs is monitored while moving the β-servo motor from position 140° to 180°. In the FL and MR legs, the β-current consumption is monitored while the β-servo motor moves from 160° to 120°. Fig. 8.4.1 shows how the β-angular position is corrected as soon as the leg stops moving down while walking on gravel ground.

For better clarity, we will analyze a part of the FR leg's curve that is marked with the green rectangle (from data point 923 to data point 1001). At data point 923, the β-current consumption exceeds T_{min} at the β-angular position 149° while the leg moves down (the first arrow indicates the point of contact with the ground). The leg continues moving down. At data point 961, the β-current consumption exceeds T_{max} at the angular position 177°. The leg sinks into the gravel ground and, in some cases, lifts up the robot's body height. Now, the FR leg corrects its position by moving the β-servo motor from position 177° to 151° that presents nearly the point of contact with the ground (at the second arrow). The same scenario repeats itself at the end of each swing phase and the robot body height varyies depending on the type of the walking surface. Fig 8.4.1 shows the same result for FL. The leg returns to the point of contact with the ground as soon as the leg stops moving down. The same scenario is repeated in all other of robot's legs as shown clearly in Figure 8.4.1.

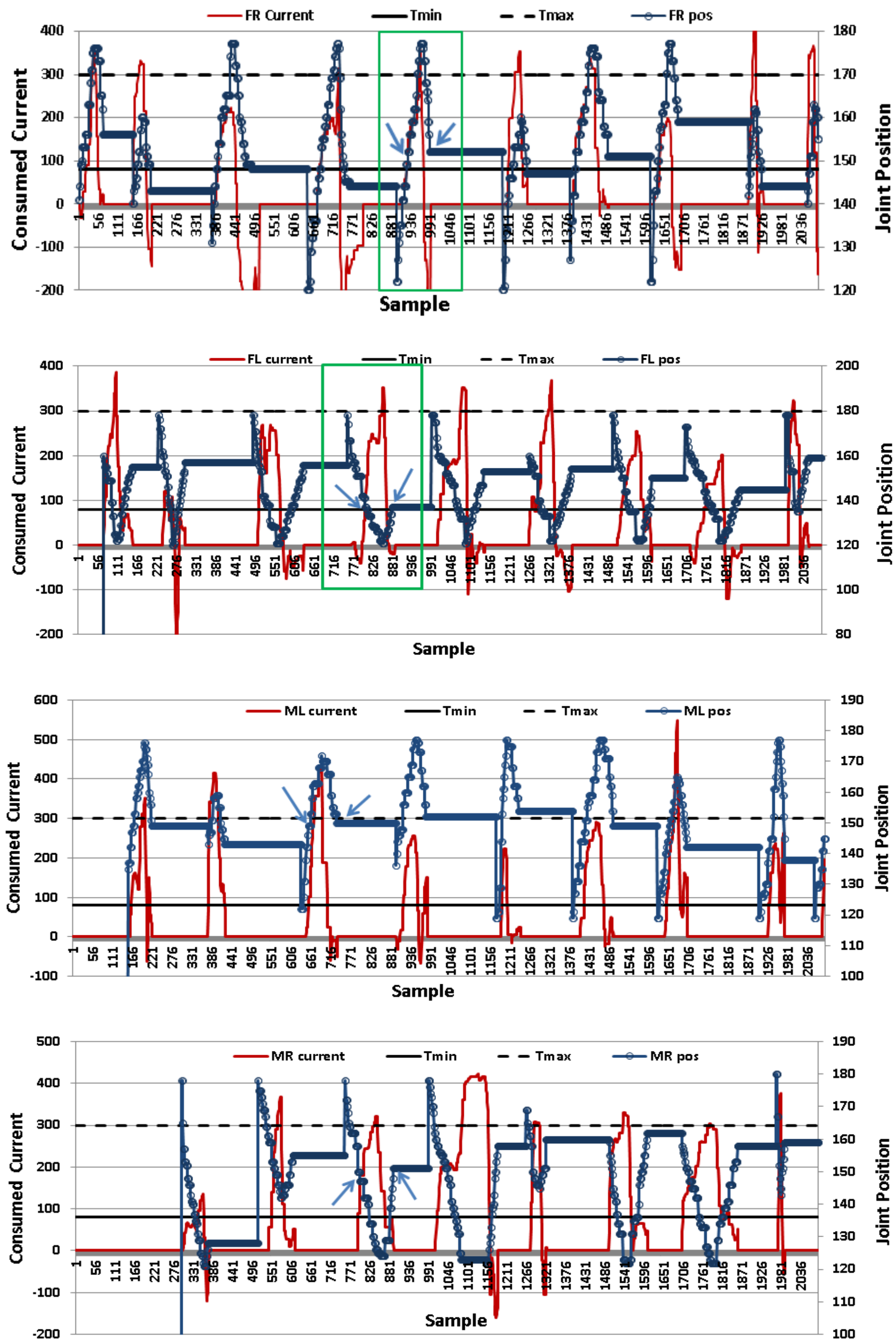


FIGURE 8.4.1. Current consumption and angular position of the β-servo motor in the robots' legs at the end of the swing phases while the leg moves down while walking on gravel ground.

8.4.2. Walking on a Sandy Ground.

The setup of the test ground consists of a sandy basin of 200x100 cm. The sand is dry and fine, similar to the sand that is found in sandy deserts and to that used previously for the sandy ground detection. Figure 8.4.2 shows the β-current consumption and the angular position of the front and middle legs while OSCAR walks on a sandy surface. It is clear that the β-current consumption exceeds T_{min} and rises gradually with significant changes in the β-angular position until it exceeds T_{max}. In order to clarify the behavior of each leg while the robot walks on a sandy surface, a part of the FR leg curve (from data point 238 to data point 318) will be analyzed, it is marked with the green rectangle. The FR leg touches the ground at data point 238 due to the exceeding of T_{min} at the β-angular position 146°. At data point 262, the β-current consumption exceeds T_{max} at the angular position 173° and the leg stops moving down. Now, the leg begins to correct its position and it returns to the point of contact with the ground. The leg moves up until the β-servo motor reaches 147°. The same scenario repeats itself for each leg at each end of the swing phase while the leg moves down while walking on sandy ground.

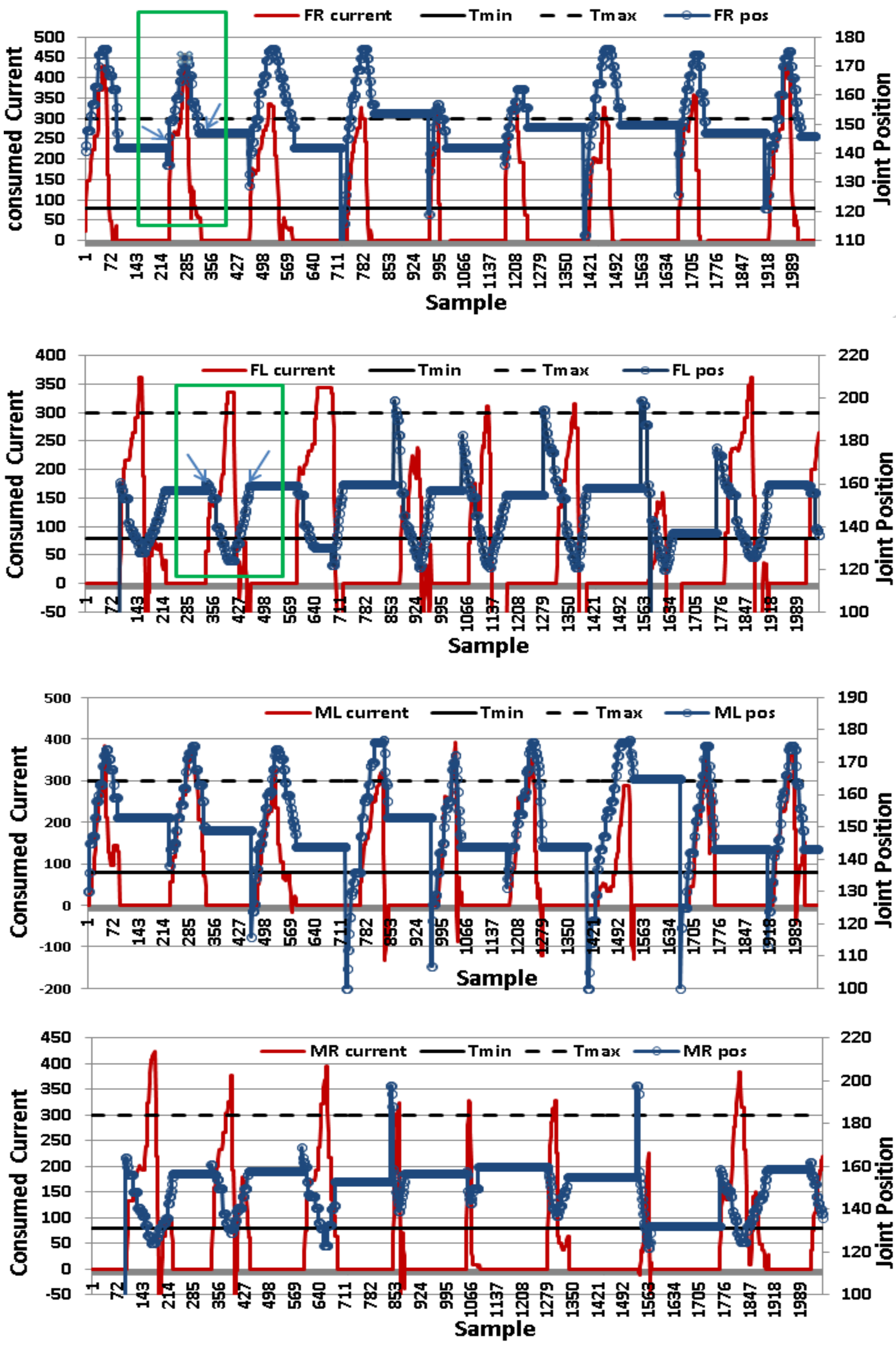


FIGURE 8.4.2. Current consumption and angular position in the β-servo motor in the front legs (FR and FL) and middle legs (ML and MR). Each pulse presents a downward motion of the leg at the end of the swing phase while walking on sandy ground.

The previous results present just the obtained results for the front and middle legs. The rear legs have the same results, therefore, they are not presented here.

As a result of the application of this approach, the center of mass swings up and down. Moving the center of mass to the bottom side in conjunction with the execution of the stance phase helps the supporting legs that achieve the stance phase to push the robot body forward more effectively than with the center of mass still being up. The center of mass works, in this case, as a pendulum that lets the robot walk faster and in a more efficient way on highly compliant surfaces. Additionally, this approach prevents the leg from sinking into sandy ground too deeply and improves the robot's walking on highly compliant surfaces.

8.5. Walking on Compliant Surfaces with an Amputated Leg

The amputation of the robot leg while the robot walks on inclined hard ground has been introduced in Chapter 5. In this Chapter, leg amputation while walking on compliant ground will be discussed . The amputated leg will be simulated here in the same way that is used in Chapter 5 (see section 5.5.2) based on software. As soon as the robot loses one of its legs, the robot reconfigures its legs on the basis of the location of the amputated leg to maintain statically stable walking. The neighboring legs of the amputated leg shift their position toward the amputated leg to compensate the resulting gap.

Figure 8.5.1 shows an experiment where the ML leg is amputated while the robot walks on sandy ground. Figure 8.5.1 (top) shows data obtained from a tilt sensor attached to the robot for reference purposes only. The pitch signal presents the inclination in the walking direction and the yaw signal presents the inclination in a lateral direction. Figure 8.5.1 (bottom) shows the step pattern. At data time 650, the ML leg is amputated and the leg loses contact with the ground. The robot reconfigures its legs and continues walking. The robot‘s tilting obviously shifts leftwards as shown at the top of Figure 8.5.1, but the robot does not lose its statically stable walking. Furthermore, due to the leg amputation, the robot walks in a curved path. The same experiment has been conducted while the robot walked on gravel ground. The obtained results are presented in Figure 8.5.2. The ML leg is amputated at data time 603 as shown in the dashed vertical line at the top of Figure 8.5.2. As noted from Figure 8.5.2, the degree of the tilting increases obviously, but the robot continues its forward walking without losing its stability. However, robot walking on compliant surfaces with an amputated leg is very difficult without using the adaptive control of the leg position.

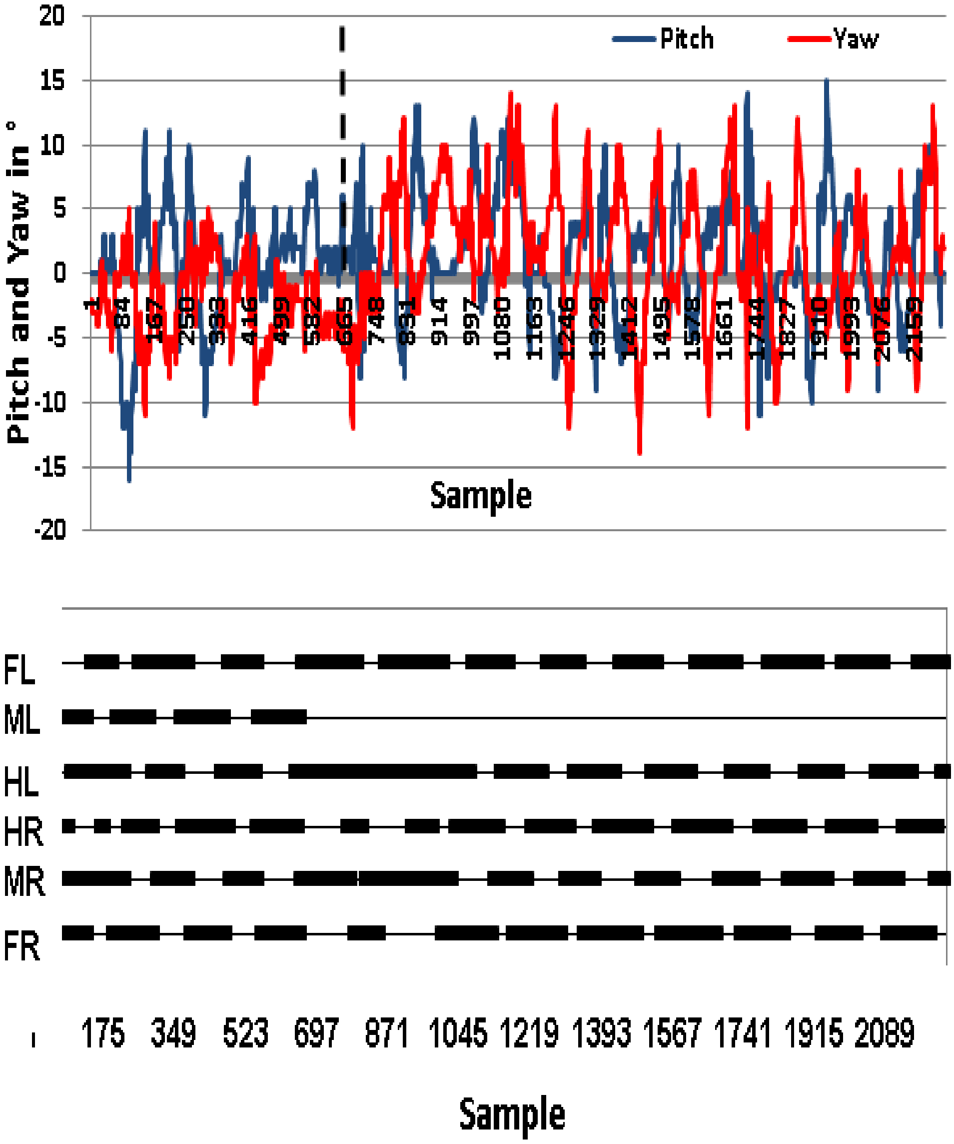


FIGURE 8.5.1. Top: Corresponding pitch and yaw readings from a tilt sensor while walking on a sandy surface with ML leg amputation. Bottom: Step pattern while the robot walks on sandy surface and the ML leg is amputated. The black bars present the stance phase and the spaces between the black bars present the swing phase.

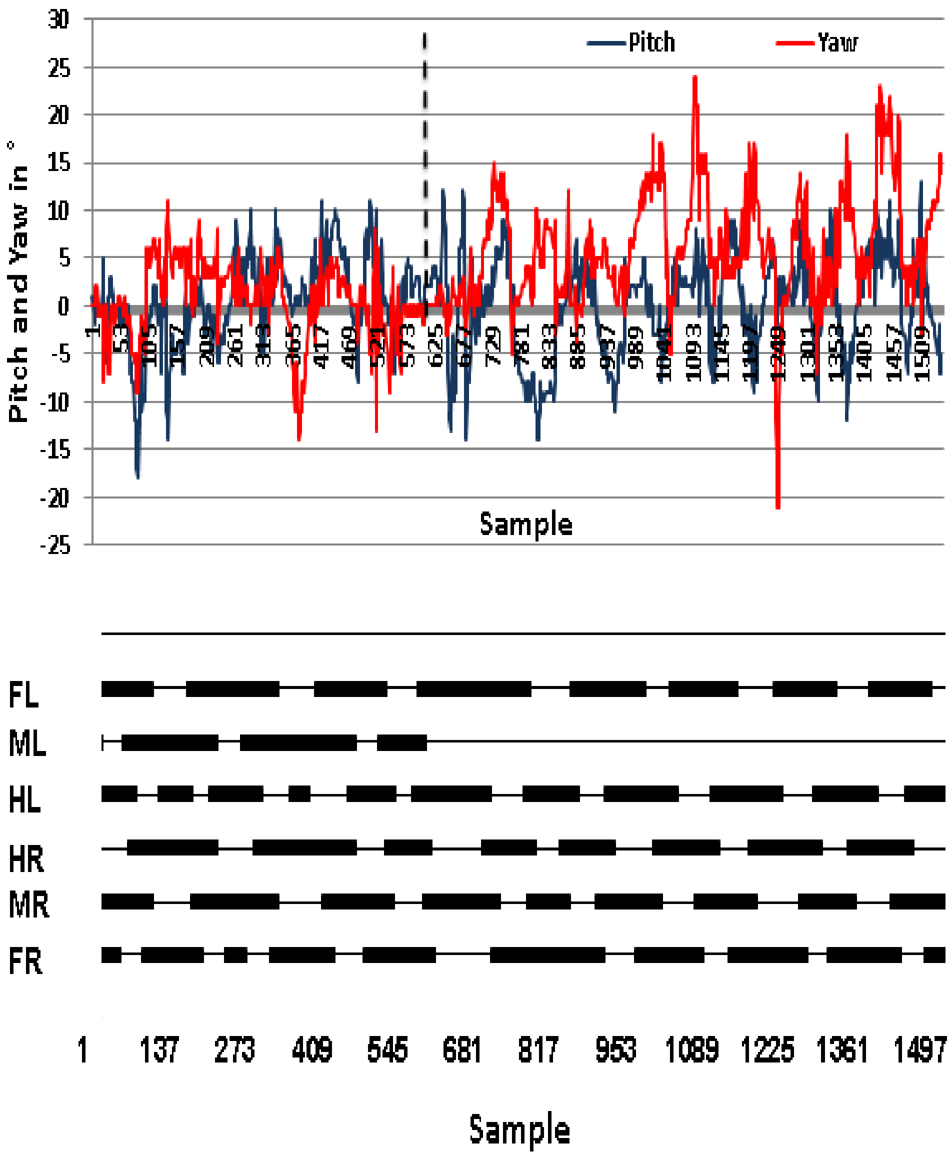


FIGURE 8.5.2. Top: Corresponding pitch and yaw readings from a tilt sensor while walking on a gravel surface with ML leg amputation. Bottom: Step pattern while the robot walks on gravel surface and the ML leg is amputated. The black bars present the stance phase and the spaces between the black bars present the swing phase.

8.6. Conclusion

This chapter has discussed our introduced approach to improve the locomotion behavior of the hexapod robot OSCAR on compliant surfaces by adapting the leg's position. Each leg adapts its position independent of the other legs, based on monitoring the current consumption and angular position in the $\beta-$joint at the end of the swing phase while it is moving down. Two experiments have been conducted to evaluate this approach, walking on gravel and sandy surfaces. Additionally, this chapter demonstrates the effect of a leg failure on the robot walking while it walks on compliant surfaces.

The presented approach improves the robot locomotion on compliant surfaces without using additional sensors. The impact of this approach on the robot walking is very obvious especially when the robot walks on a sandy surface, where the robot can walk faster as if this approach was not used.

The following chapter will show the combination of our three approaches that are mentioned previously to enable our hexapod robot to walk on one of the most difficult terrains, which is walking on sandy inclined surfaces.

CHAPTER 9

Adaptive Walking on Sandy Inclined Ground

9.1. Introduction

Walking on inclined sandy surfaces is considered one of the most difficult challenges the walking robot is facing. In this case, the robot has to cope with the inclination and the compliant surface at the same time. The robot has to adapt its locomotion on inclined surfaces by adjusting its body posture to prevent the robot from overturning. Furthermore, the robot has to detect the properties of the walking surface to achieve adaptive walking on such difficult terrain. This is not an easy task to accomplish unless the control system has a robust algorithm that is able to cope with this challenge.

Our aim in this chapter is to show our results of the combination of the detection and reaction mechanisms to different terrain types as described in the previous chapters. The combination includes the last three approaches that have been put forward previously. Several experiments have been conducted to test the ability of OSCAR to achieve adaptive walking on one of the toughest terrains, which is an inclined sandy surface. In these experiments, our hexapod robot walks downhill and uphill on a sandy surface.

All these approaches are based only on monitoring the current consumption and angular position of the β and γ servo motors of each leg. For the inclination detection, the current consumption in the $\gamma-$servo motor will be monitored during the stance phase. For sandy ground detection, the β-current consumption and its angular position will be monitored during the swing phase while the leg is moving down at the end of this phase. Figure 9.1.1 shows the joints of OSCAR's leg that contribute to determining the nature of the walking substrate in these experiments.

9.2. Walking on Inclined Sandy Surface

In this chapter, two cases have been studied in detail. Either the robot faces downhill sandy surface or uphill sandy surface. The OSCAR robot decides the type of gait pattern to achieve adaptive walking based on the type of the walking terrain.

9.2.1. Downhill Sandy Surface.

The OSCAR robot achieves adaptive walking based on the use of the combination of our introduced approaches that were discussed in detail

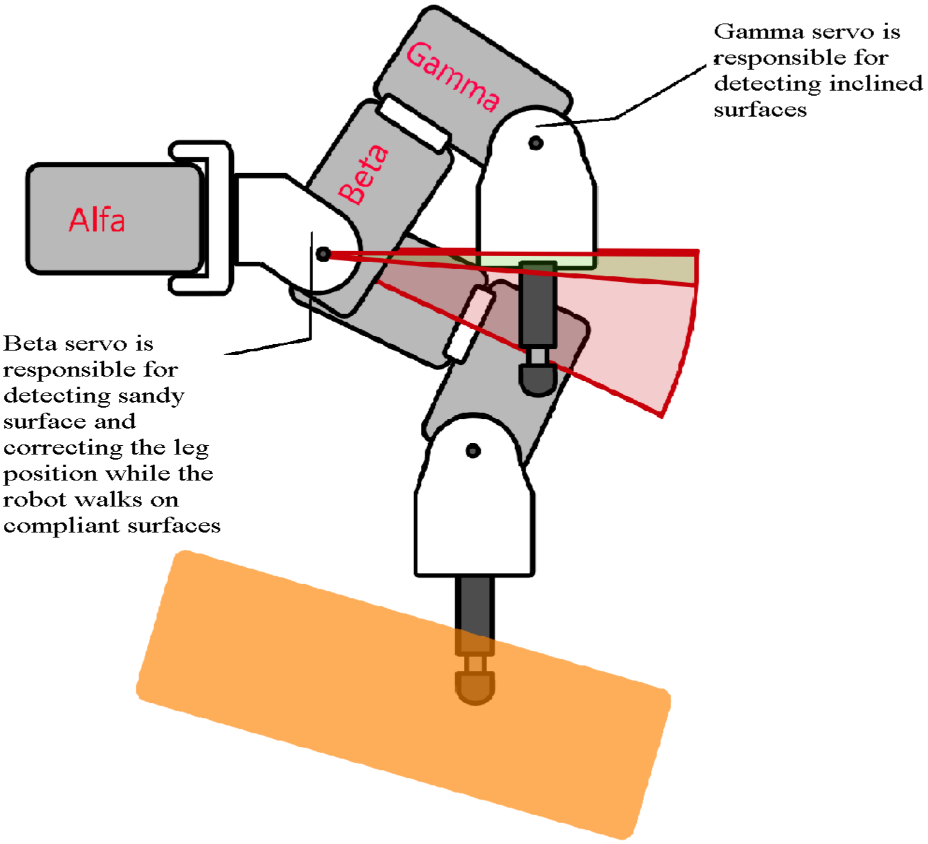


FIGURE 9.1.1. OSCAR leg's schematic

in the previous chapters. This combination comprise the following approaches:

- Inclination Detection and Adaptive Walking on an Inclined Surface
- Sandy Surface Detection and Adaptive Walking on a Sandy Surface
- Control of the Leg Position on a Compliant Surface

9.2.2. Uphill Sandy Surface.

Walking on uphill sandy surface is a challenging task and is considered one of the most difficult terrains that a walking robot can face. This type of terrain has been tested in context of this research, but it was noted that walking on sandy uphill surface needs a special strategy to cope with this case. This section shows our introduced approach to enable a hexapod robot to walk on such complex terrain.

Our introduced approach is based on the combination between our previously introduced approaches, which are a sandy ground detection approach and a control of the leg position approach that are presented

FIGURE 9.2.1. OSCAR walks on sandy uphill surface

previously. The control of the body posture due to the inclination detection was ignored in this approach for reasons that will be mentioned later. The sandy ground detection approach enables the OSCAR robot to detect the sandy surface, while the control of leg position approach enables each leg to control the distance between the robot body and the walking substrate. The two approaches are used together while the robot walks uphill. This combination improves the robot walking on sandy uphill surface, but it is still insufficient to enable the robot to walk forward effectively. In addition to the mentioned combination, a new suggested strategy has been added to enable the hexapod robot to walk on uphill sandy surface. The suggested strategy is based on the synchronization between the moved legs that are touching the ground. This synchronization provides the robot's legs with the sufficient driving force during uphill walking. This approach was implemented in ORCA architecture and tested on a low-cost version of the OSCAR walking robot. This work is based on our previous work that is introduced in [**8**]. Figure 9.2.1 shows OSCAR robot walks on sandy uphill surface.

As soon as the robot detects sandy surface, OSCAR executes some new reactions that help the hexapod robot to achieve adaptive and effective walking while it walks on uphill sandy surface.

9.2.2.1. *Reaction to Uphill Sandy Ground.* Walking on compliant surfaces such as sand, leads to the activation of the control of the leg position regardless of the sandy ground detection. Each leg controls the distance between the robot's body and the surface independent from the other legs. This work has been presented in detail in our previous work and is active either if the robot walks on flat or inclined ground. In case of sandy ground detection, the robot system changes some parameters to achieve adaptive and effective walking:

• **Decreasing the height of the robot body:** The robot body is decreased to be near to the walking substrate. This procedure guaranties that the robot does not overturn while walking on uphill/downhill surface. Additionally, it increases the efficiency of the robot legs during the stance phase. The reason is: the produced torque in the α-joint is greater than the applied torque on it. This leads to pushing the robot forward effectively on sandy surface.

• **Changing the overall servo motor speed:** The robot increases the overall servo motor speed while it walks on uphill sandy surface to reduce the time of legs synchronization and to compensate the time during the climbing on sandy surface due to difficultly of the terrain. In case downhill walking, the robot decreases the overall servo motor speed to prevent the robot from overturn while it walks downhill.

• **Synchronization between the legs:** This is considered the most important procedure. Walking on sandy surface in general and on uphill sandy surface in particular needs a special strategy. In the normal case, as soon as the leg reaches its AEP and touches the ground it executes its stance phase without waiting for the other stance legs that reach their aimed position (AEP) later. This leads to a weak driving force while walking on sandy surface due to the low friction coefficient between the leg and the surface. The low friction coefficient is obvious while walking on uphill sandy surface. As a result, there is not a united driving force that pushes the robot forward effectively while walking on sandy or sandy uphill surface. Additionally, in this case the α-joint is loaded and the robot sometimes stops moving forward. To solve this problem, the robot has to synchronize its stance legs in case of facing such a terrain to produce a unified driving force that pushes the robot forwards. Our suggested approach for legs synchronization is activated as soon as the robot detects sandy surface. This approach work as following: When one leg reaches its AEP and is ready to achieve the stance phase, it has to wait for the other two legs at this point. As soon as three legs are at their AEP and they are ready to execute their stance phase, they retract backward together producing a unified driving force of the stance legs, which leads to push the robot forward on sandy flat and uphill/downhill surface. In this method, OSCAR robot keeps the Cruse's rule for robot legs coordination. The leg does not switch into swing phase unless its two neighboring legs are touching the ground, but at the begin of stance phase the leg does not retracts rabidly, it waits the other legs due to the legs synchronization. In other words, the robot uses Cruse's rule for swing phase and synchronization approach for stance phase.

9.3. ORCA Implementation

In this case, the OCU unit for each leg has to monitor the current consumption and angular position in the distal and middle joints (β-

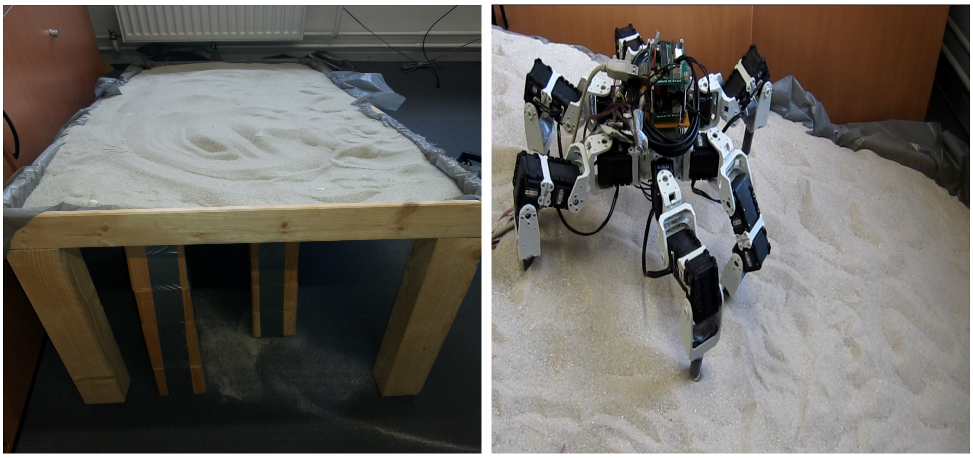


FIGURE 9.4.1. Test ground (left). Our hexapod robot while walking down on a sandy surface with 15° inclination (right).

and γ-joints) at the same time. For walking on inclined surfaces, the OCU unit monitors the γ-current consumption while the leg executes the stance phase. For walking on sandy surfaces, the OCU unit monitors the current consumption and angular position of the β-joint while the leg moves down at the end of the swing phase. As soon as the OCU unit detects that the current consumption and angular position deviate from the predefined value, it sends a signal to the higher level. The higher control system decides that the OCU unit of each leg has to react. For more information see the previous chapters (Chapters 7, 8 and 9). As soon as the robot detects the type of walking surface, each OCU unit changes the parameter of the BCU units of the same leg. For instance, when the robot walks downhill, the front and rear legs change their span, the front legs expand and the rear legs flex to make the robot body nearly horizontal. In case of a sandy surface, the OCU unit changes the BCU parameters of the leg to enable it to correct its position.

9.4. Experimental Results

For the experimental evaluation, a test ground was established, which consists of a sandy basin of 180x70 cm indoor area with 15° inclination. Figure 9.4.1 shows the used test ground.

The sand is dry and fine, similar to the sand that was used in the previous experiment. The following experiments focus on downhill and uphill walking. For inclination and sandy ground detection, the current consumption and angular position in the β and γ-servo motors are monitored while the robot walks on downhill or uphill sandy surface.

9.4.1. Downhill Sandy Surface.

To clarify how the robot detects the inclination direction and adapts its body posture while it is walking, Figure 9.4.2 shows the obtained results from the front and rear legs.

The γ-current consumption is monitored during the stance phase, therefore, the depicted results only present the γ-current consumption during the stance phases. The swing phases are not depicted because they are not relevant to the inclination detection. The γ-current consumption of the front legs (FR, FL) exceeds the threshold at data points 13 and 53 respectively and thus detects the downward inclination during the first stance phase. At data point 53 (see dashed line in Figure 9.4.2), the robot detects the downhill inclination and begins to correct its body posture. The body inclination is reduced from 15° to an average of 8° at data point 146 as shown in the bottom of Figure 9.4.2, which shows reference tilt sensor values. Here, the pitch value represents the inclination in the walking direction, the yaw value presents the inclination in lateral direction.

Several differences are visible in comparison to the previous experiments that are conducted on hard inclined surfaces. In our previous work, the robot body was corrected to 6°, while in this experiment, the robot does not correct its body posture to more than 8°. This is due to the robot's leg sinking into the sand, which leads to more inclination in the robot's body posture toward the direction of inclination. The center curve of Figure 9.4.2 shows the γ-current consumption in the hind legs, which does not exceed the predefined threshold while walking downhill on a sandy surface. Due to the combination of the control of the leg position and body posture adaptation, the robot tends to tilt slightly forward for a short time as shown in the bottom of Figure 9.4.2, which shows some pulses in a pitch curve. They are due to the control of the leg position on a compliant surface. Each leg tries to control its position while walking on a sandy surface during the swing phase, while during the stance phase, the leg has to move down again to maintain the body posture adjustment while walking downhill. Therefore, these pulses emerge. However, this emergent motion also improves the robot walking on sandy surfaces. The robot body motion here is similar to a camel's body motion while it walks in deserts.

To clarify how the robot detects the sandy ground and how each leg controls its position while walking on a sandy inclined surface, Figure 9.4.3 shows the obtained results from the front legs (FR and FL). The red curve represents the β-current consumption and the blue curve represents the β-angular position while the leg moves down at the end of the swing phase. The results depicted here present only the obtained samples during the swing phase while the leg moves down, the stance phases are not depicted in this figure because they are not relevant to detect sandy ground and to the control of the leg position.

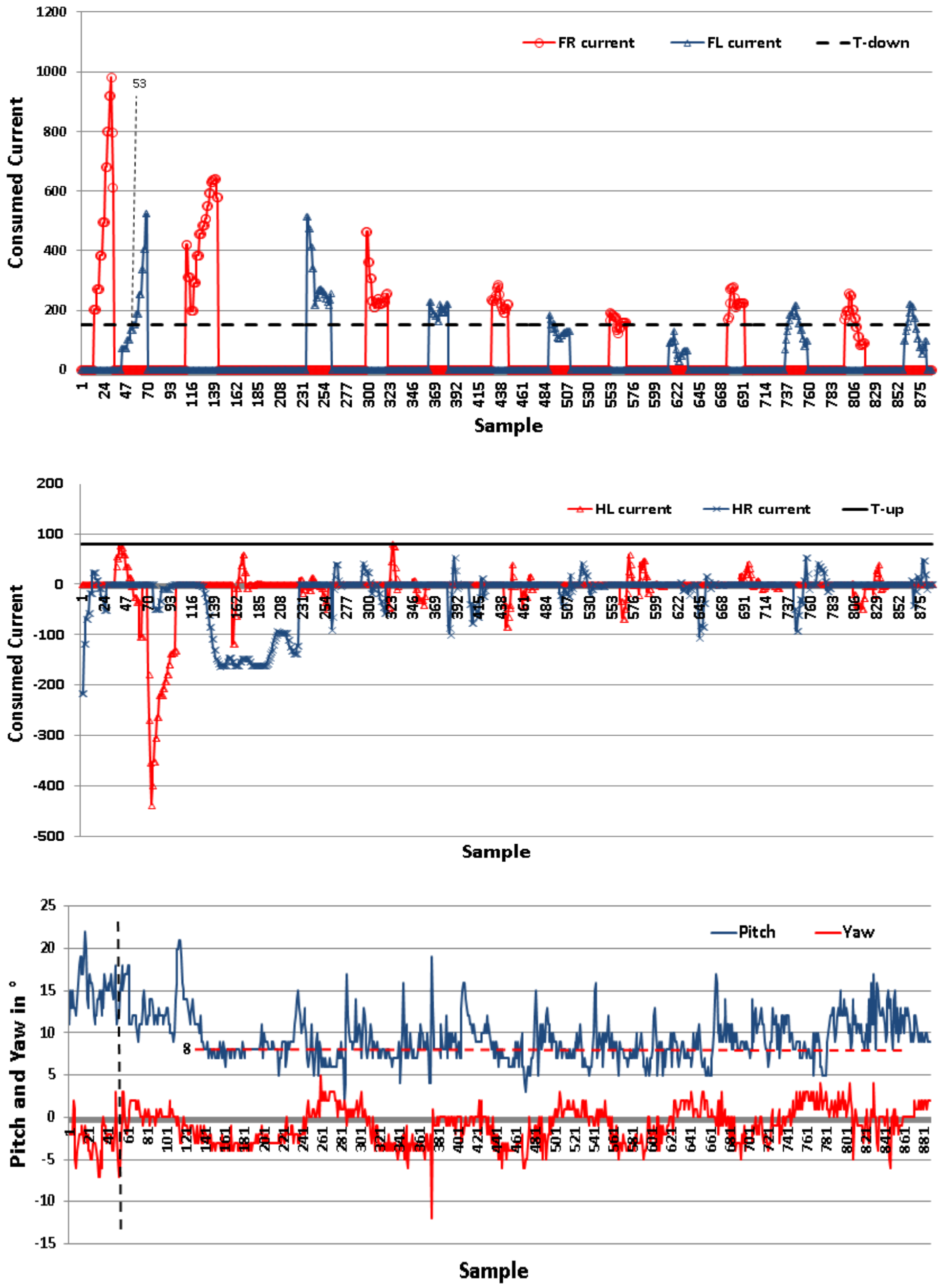


FIGURE 9.4.2. Correction of downhill inclination. Top: Consumed current in the front legs (FL, FR) during the stance phase. Center: Consumed current in the rear legs (HL, HR) during the stance phase. The thresholds T-up and T-down present the predefined threshold for uphill and downhill respectively. Here, each pulse represents one stance phase. Bottom: Reference inclination of the robot obtained from a tilt sensor.

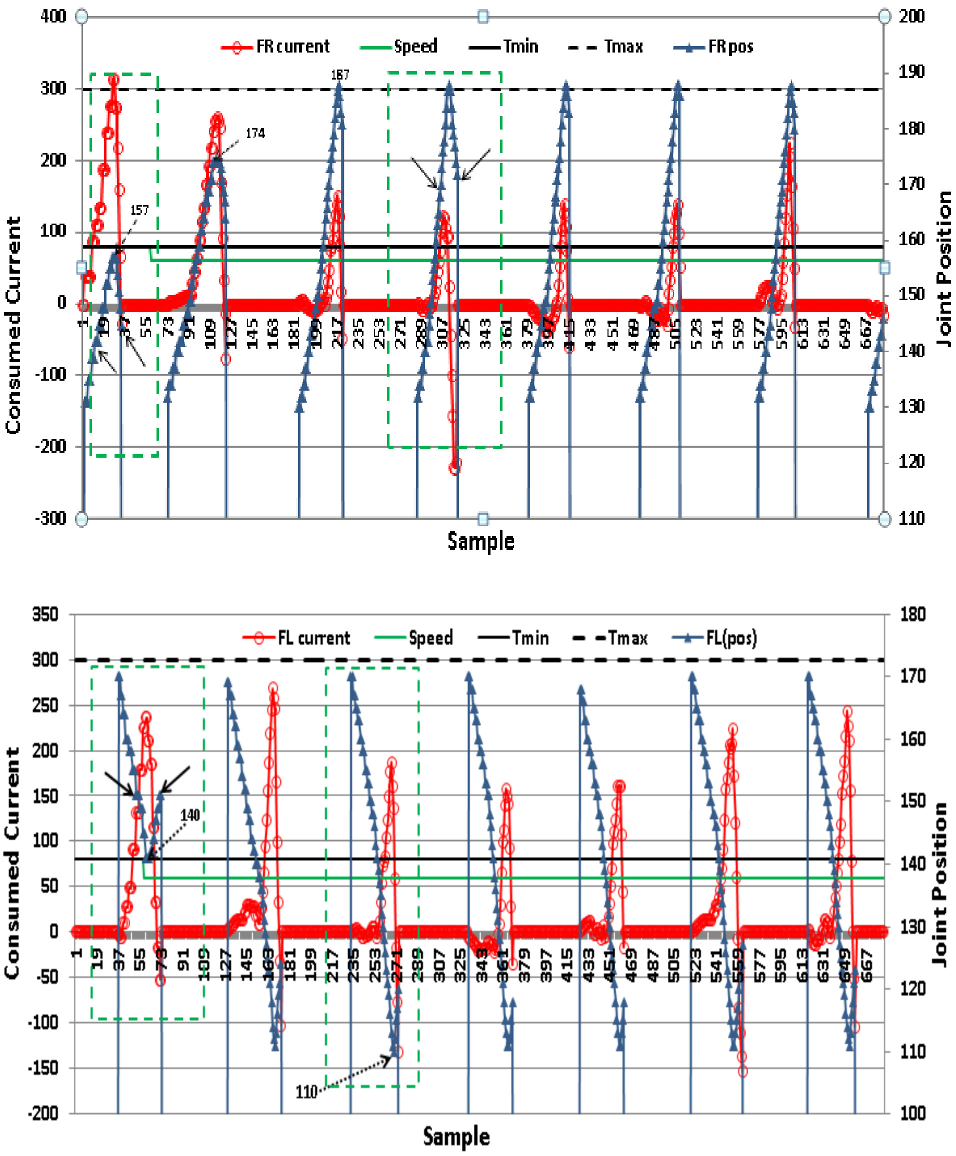


FIGURE 9.4.3. Current consumption and angular position in the β-servo motor in the front legs (FR and FL) at the end of the swing phase while walking on an downhill sandy surface. Each pulse presents the obtained sample at the end of the swing phase while the leg moves down. Thresholds T_{max} and T_{min} are shown dashed and solid black respectively.

For more clarification, we will analyse the first step of the FR leg that is surrounded with the green rectangle at the top of Figure 9.4.3. The FR leg touches the surface at data point 8 and β-angular position 139° (the first arrow). The first arrow represents the β-angular position of the point of contact with the ground. The second arrow represents the correction of the β-angular position to the point of contact with the ground. The β-current consumption rises gradually until it exceeds T_{max} at angular position 157°. The leg stops moving down. Here, the leg has detected the sandy surface due to the obvious angular position changes in addition to the rising of the current consumption above the predefined threshold.

On a compliant surface, the control of the leg position is activated, which will correct the leg position and return it to the point of contact with the ground. The leg moves again to the position 139° that presents the first contact with the ground (at the second arrow). The robot detects the sandy surface when at least two legs detect a sandy surface. As shown in the plot at the bottom of Figure 9.4.3 (FL leg), the FL leg touches the surface at data point 47 and angular position 155° (see the green rectangle in FL leg at the first step). The leg continues moving down until it reaches its maximal position 140°. The leg detects the sandy surface and tries to correct its position. It corrects its position to 153° (the second arrow). Here, the robot detects the sandy surface and decreases its speed to 40% (as shown in the green curve).

Now, the robot has detected the inclination and the sandy surface. It adapts its body posture and gait pattern based on our used approaches. The used approaches will interact with each other. The front and the rear legs change their span to adjust the body posture based on the direction of inclination. In our case (downhill walking), the front legs extend their span and the rear leg flex. For more clarification, we will analyze the movement of the FR leg at the end of the swing phase while it moves down as shown in the top of Figure 9.4.3. At the first step, the FR leg moves from position 130° to 160° before the inclination detection. But after the inclination detection, the leg moves to position 187° (see the second green rectangle at the fourth step). The FR leg has increased its span by 27° to adjust the robot's posture while walking downhill. This increase is clear in the fourth step of the FR leg that is surrounded by a second rectangle. At the fourth step, the leg touches the surface at data point 305 and angular position 170° (the first arrow indicates the point of contact with the ground). The leg moves down to the angular position 187°, then it stops moving down. It corrects its position to the point of contact with the surface. The leg returns to position 172° (the second arrow). The same scenario repeats itself at each end of the swing phase. The same scenario is executed by the FL leg. The FL leg moves at the first step to position 140° and then to position 110° in the next steps after the inclination detection.

9.4.2. Walking on an Inclined Sandy Surface with an Amputated Leg.

The leg amputation has been mentioned and discussed in the previous chapters. In this chapter, the effect of leg disturbances such as leg amputation on the robot's walking will be analyzed while the robot walks on such difficult terrain. The leg amputation will be conducted while the robot walks on a sandy inclined surface. It is simulated in software, i.e. the amputated leg is stiffened and moved on top of the robot's body similar to the previously conducted experiments that are mentioned previously. The robot reconfigures its legs's positions based on the direction of the inclination and the location of the amputated leg. Figure 9.4.4 shows the obtained results in case the ML leg is amputated while the robot walks down on a sandy surface. The top of Figure 9.4.4 shows the data obtained from the inclination sensor fixed on the robot body. In our experiment, Figure 9.4.4 shows that the inclination is detected before the leg amputation and the robot adjusts its body posture from 15° to 4° (at the first circle). At data point 1602, the ML leg is amputated and the leg loses its contact with the ground. The robot's legs reconfigure their position to ensure statically stable walking on the downhill surface. The neighboring legs of the amputated leg move toward the amputated leg to reduce the emergent space due to the leg amputation. Furthermore, the FR leg shifts its PEP and AEP counter clockwise. Then, the robot continues its forward walking with five reconfigured legs and achieves statically stable walking. It is noted that the robot tilts just one time forward as shown in the top of Figure 9.4.4 (at the second circle) and the robot walks in a curve, but the robot does not overturn or lose its statically stable walking. The bottom of Figure 9.4.4 shows the step gait pattern. As noticed from Figure 9.4.4, the robot walks with amputated leg and adjusted body posture. Its body is adjusted from 15° to 8°. As a result of the leg reconfiguration and adaptive walking on the sandy surface, the robot is not affected if one of its legs is amputated. Therefore, the combination of our approaches improves the robot's walking and enables it to walk on difficult terrain with an amputated leg.

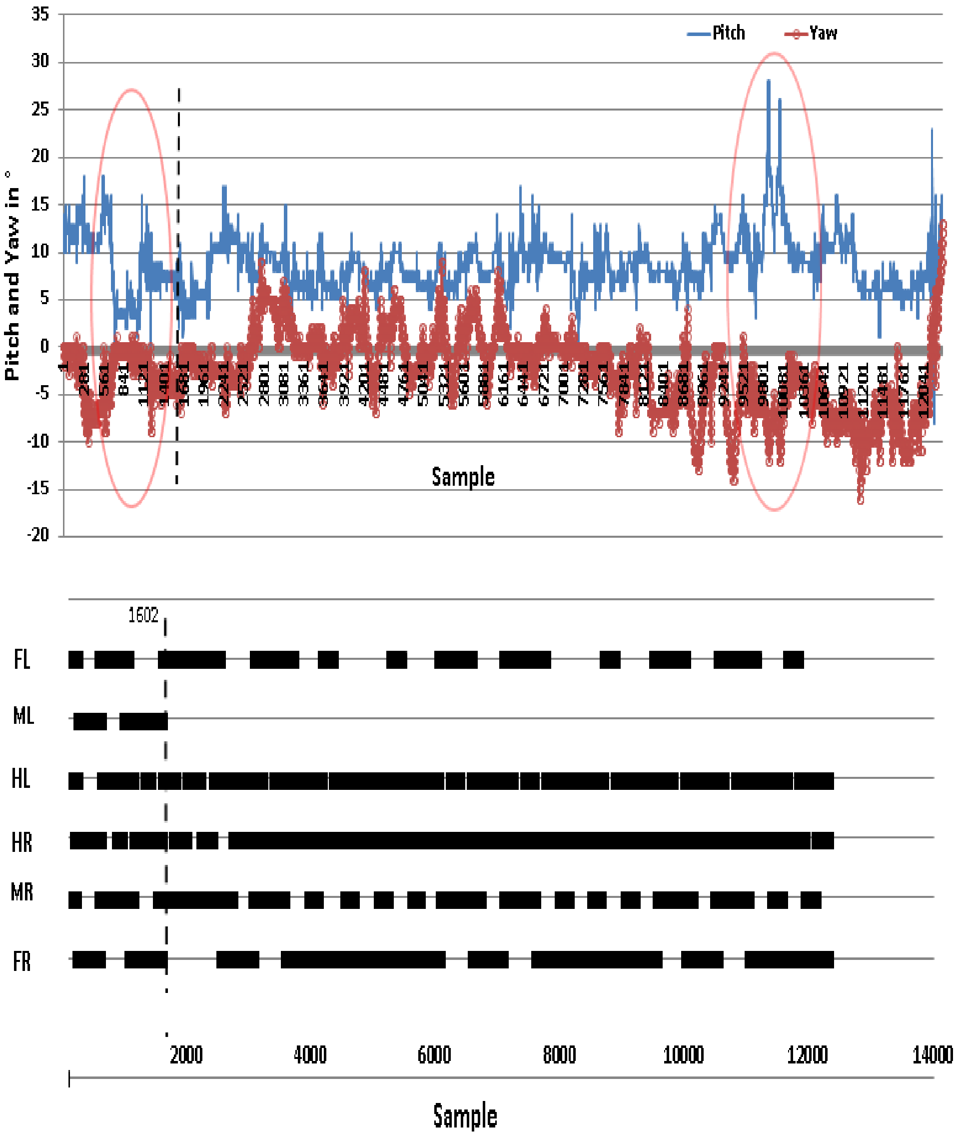


FIGURE 9.4.4. Top: pitch and yaw signals reading from a tilt sensor. Bottom: Step pattern. The black bars represent the stance phases and the spaces between the black bars represent the swing phases. The dashed vertical line represents the point of leg amputation.

9.4.3. Uphill Sandy Surface.

In this experiment the robot walks on uphill sandy surface. Figure 9.4.5 shows the obtained results while OSCAR walks uphill.

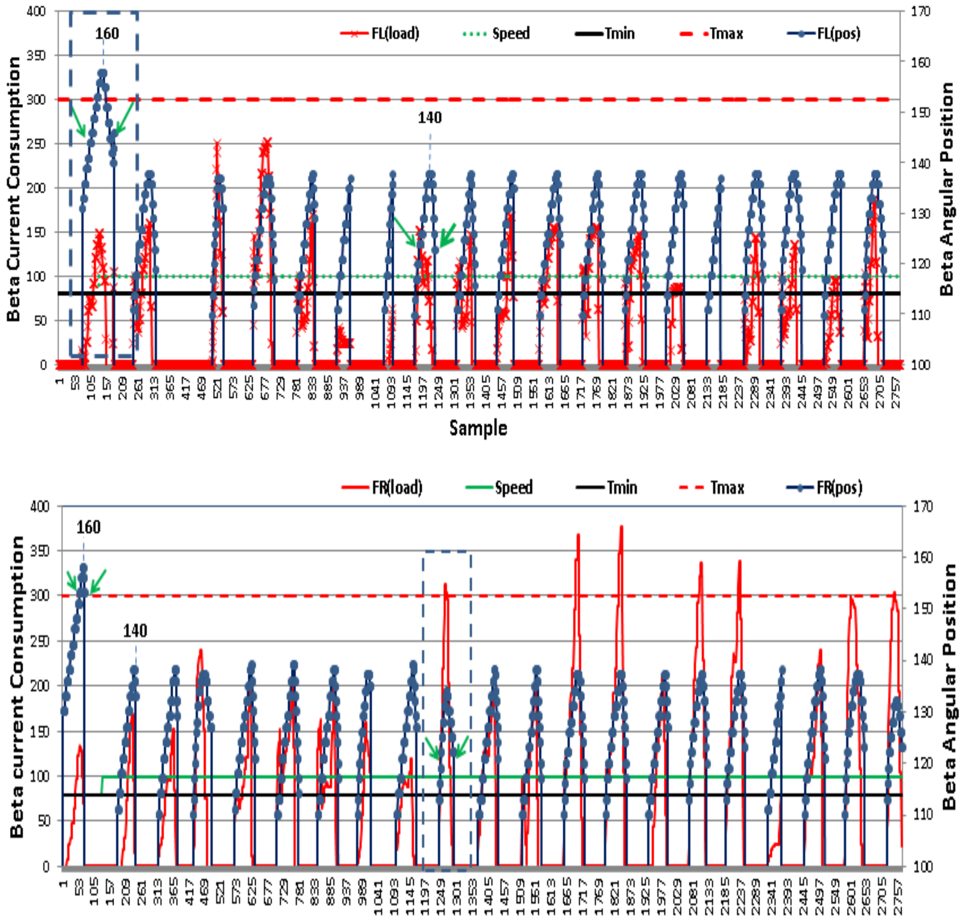


FIGURE 9.4.5. The β-current consumption and the β-angular position while walking on uphill sandy surface

To clarify how the robot detects the sandy ground and how each leg controls its position while walking on sandy inclined surface, Figure 9.4.5 shows the obtained results from the front right and left legs (FR and FL) that detect sandy surface in this experiment. The red curve represents the β-current consumption and the blue curve represents the β-angular position while the leg moves down at the end of the swing phase, the stance phases are not depicted in this figure, because they are not relevant to the sandy ground detection and to the control of the leg position. In this experiment, the β-joint moves the leg down from position 130° to 160° in case the leg does not yet detect sandy surface. For more clarification, we will analyse the first step of the FR leg at the bottom of Figure 9.4.5. The FR leg touches the surface at data point 42 and β-angular position 153° (first arrow). The leg continues its movement until it reaches its maximal position unless the β-current consumption rises above T_{max}. When the β-joint reaches its maximal position 160°, the leg corrects its position to the point of contact with the ground at 153° (second arrow). The leg finishes its swing phase. Here, the leg has detected the sandy surface due to the obvious angular position changes in addition to the rising of the current consumption above the predefined threshold T_{min}. The robot detects the sandy surface, when at least two legs detect sandy surface. As shown in the plot at the top of Figure 9.4.5 (FL leg). The FL leg touches the surface at data point 113 and β-angular position 146° (the first arrow in the first step). The leg continues moving down untill it reaches its maximal position 160°. The leg detects the sandy surface and tries to correct its position. It corrects its position to 146° (second arrow). Here the robot detects the sandy surface and activates several procedures that improve the walking on this terrain. It increases its speed by 20% (the speed is changed from 80 to 100 as shown in the green curve). Additionally, it decreases it body height. The β-joint moves now from 110° to 140° due to decreasing the robot body height as shown in the following depicted steps in the FR and FL legs in the Figure 9.4.5. After the two mentioned adaptations, the robot adapts its gait pattern based on using the synchronization between the stance legs. The legs synchronization can be also activated while the robot walks on sandy downhill surface. Without legs synchronization on downhill sandy surface, the robot can progress forward. But, it has observed that the robot can be more effective with using legs synchronization while it walks on downhill surface.

Figure 9.4.6 shows the recorded gait pattern during OSCAR walking on uphill sandy surface. The black bars represent the time of contact with the ground and the spaces between them are swing phases. The stance phases are depicted in blue bars above the black bars. It begins as soon as the leg is at the AEP and touches the ground. It ends when the leg is at the PEP. The leg switches into swing phase again when the Cruse's rule is fulfilled. The red bars represent the active movement of the leg during the stance phase. Before sandy ground detection, it begins and finishes with the stance phase as shown in Figure 9.4.6. After sandy ground detection, it depends on legs synchronization. In our case the robot detects sandy surface at data point 667 (see the first black vertical line). After this point, the robot synchronizes its legs. For more clarification we will analyse the cases that are marked with three circles. The vertical dashed blue line represents the moment of movement of all three legs (MR, HL and FL), when all legs are at the AEP. The FL leg reaches its AEP first (red circle), its stance phase is activated, but it waits for the other two legs. Then the MR leg reaches its AEP (blue circle), its stance phase is activated, but it waits untill another leg reaches its AEP. Finally, the HL leg reaches its AEP (green circle). At this moment, the three legs move together executing the stance phase. The same scenario is repeated for all legs. The other similar cases are obvious at the two dashed vertical lines. Due

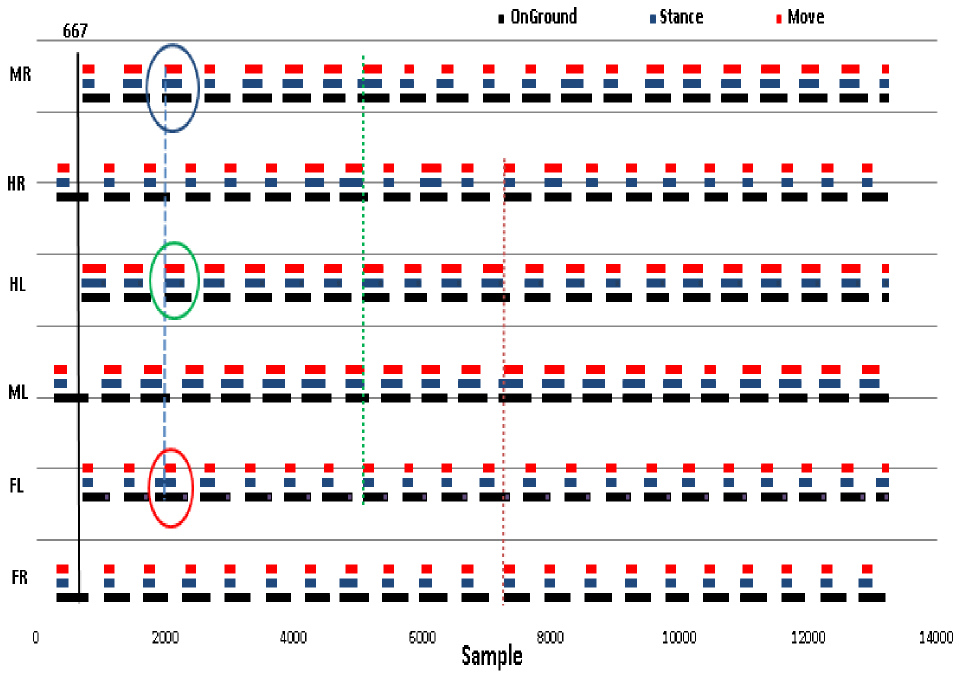


FIGURE 9.4.6. Gait pattern while walking on sandy uphill surface

to the legs synchronization, the robot can now walk on sandy uphill surface more efficiently and quickly. It has been found through experiments that without this measure, our robot cannot walk on sandy uphill surface at all.

Previously, we have mentioned that the control of the robot's body posture has been ignored while the robot walks uphill on sandy surface. In the following we interpret why it is ignored. In context of this research, the walking on sandy uphill surface with inclination detection and the control of the robot's body has been tested. When the robot detects uphill surface, it adjusts its body posture. The rear legs expand and the front legs flex to make the robot body horizontal while the robot walks on sandy uphill surface and vice versa while walking on downhill surface. Due to the robot body posture adaptation, the robot walks on sandy uphill surface very slowly. The rear legs, which are expanded, are making a big effort to push the robot body forward. In some cases, the rear legs stop moving backward during executing the stance phase. The reason is, the produced torque in the α-joint is less than the applied torque on it due to the expansion of the rear legs to the maximal position and due to the leg's sinking into sand. Therefore, the control of the robot's body posture has been ignored while walking on sandy uphill surface and it was replaced with our suggested strategy.

In context of this work, different experiments have been conducted on variant inclined sandy surface (15°, 20° and 27°) as well as on sandy flat terrain. It has been found that the effective walking on inclined sandy surface is from 0° to 20°. Tables 1 and 2 show the measured speed of our hexapod robot on different types of terrains:

Inclined sandy ground	**Uphill**		**Downhill**	
	Normal walking	**Walking with amputated leg**	**Normal walking**	**Walking with amputated leg**
15°	0.65 cm/sec	0.13 cm/ sec	3 cm/sec	1.72 cm/sec
20°	0.3 cm/sec	Can not	2.7 cm/sec	1.66 cm/sec
27°	Can not /Overturn	Can not /Overturn	4 cm/sec	Overturn

TABLE 1. Speed of our hexapod robot on inclined sandy ground

Flat Sandy Ground	**With Legs Synchronization**		**Without Leg Synchronization**	
	Normal Walking	**(ML) Leg Amputation**	**Normal Walking**	**(ML) Leg Amputation**
	2 cm/sec	1 cm/sec	1.75 cm/sec walking in curve (Left) with the slippery legs	1 cm/sec walking in curve (Left) with the slippery legs

TABLE 2. Speed of our hexapod robot on sandy flat ground

9.5. Discussion

In context of this work, several experiments have been conducted to test the effect of legs synchronization on the robot walking. The Legs synchronization has been tested on different types of terrains which are Hard Flat Ground, walking on Inclined Sandy Surface and Sandy Flat Surface. This study aims to show the effect of this method on the robot walking.

9.5.1. Hard Flat Surface.
While the robot walks on flat hard ground, the legs synchronization is not activated, because the robot does not detect sandy ground. The legs synchronization activates just when the robot walks on sandy ground. On hard surface, the robot walks in tripod gait without using legs synchronization. The gait pattern looks like synchronized, because the legs reach their AEP at the end of the swing phase mostly at the same time. Therefore, the legs switch into stance phase nearly at the same time. On hard ground, the synchronization between the stance legs does not play any important role.

9.5.2. Sandy Flat Surface. Walking on flat sandy surface is tested based on using legs synchronization and without using legs synchronization. Based on using legs synchronization, the robot walks forward in straight direction. While it walks in curve when it does not use legs synchronization. The reason is the slipping of some legs due to the low friction coefficient between the leg and the sandy surface. The speed of robot progress in both cases has been measured. The progress of the robot based on using legs synchronization is slightly faster than walking without using legs synchronization. OSCAR's speed based on using legs synchronization is 2 cm/sec. It speed without using legs synchronization is 1,75 cm/sec. Table 2 shows the measured speed of our hexapod robot in both cases.

9.5.3. Sandy Inclined Surface.
The robot is tested on inclined sandy surface with variant degree (15°, 20° and 27°) of inclination.

- **Downhill walking**: On such type of terrain, the effect of legs synchronization and the control of the robot body posture have been tested. Regarding the use of the legs synchronization or not, the robot progresses forward very well in both cases. Walking on downhill surface leads to shift the center of gravity forward which helps the robot to progress forward on downhill sandy surface. But the center of gravity has to be within the formed polygon of the touching legs that achieve stance phase to prevent the robot from overturning. To increase the statically stable walking, the robot decreases its body height as soon as it detects sandy surface. This procedure increases the static stability of robot walking on inclined surface and prevents the robot from overturning unless the degree of inclination increases above a specific limit. Therefore, the control of robot body posture plays an important role on such type of terrain. Based on our conducted experiments, the robot can achieve adaptive walking without using the control of the robot body posture and without overturning up to 20° of inclination. If the degree of inclination increases above the 20° of inclination, the robot has to use the control of body posture to prevent itself from overturning. Therefore, the robot adjusts its body posture as soon as it detects downhill surface. The limit degree for adaptive walking on downhill sandy surface is 27° of inclination. As noted from our experiments, the robot walks on downhill sandy surface very well without using legs synchronization that will be activated as soon as the robot detects sandy surface. The legs synchronization does not play any important role on such type of terrain. The robot speed is measured for variant degree of inclination. It is 3 cm/sec for 15°, 2,7 cm/sec for 20° and 4 cm/sec for 27°. The leg amputation is tested while the robot walks on such terrain. The robot limps during its walking. The direction of the tilting is based on the position of the amputated leg, e.g. the robot limps forward left if the ML leg is amputated. Additionally, the robot walks in a curve. The curve direction is based also on the position of the amputated leg, e.g. the robot walks in a left curve if the ML leg is amputated. The limit degree of inclination with amputated leg is up to 20°. Table 1 shows the measured speed of our hexapod robot in several cases.

- **Uphill walking**: On such type of terrain, the legs synchronization plays an important role. It helps the robot to progress forward more efficiently and quickly. It has been found through experiments that without this measure, our robot cannot walk on sandy uphill surface at all. The control of robot body posture is ignored while the robot walks on uphill sandy surface due to its negative effect on the robot forward progress. The robot can walks on inclined sandy surface up to 20° of inclination. The speed of robot progress decreases if the degree of inclination increases. Leg amputation on such terrain is also tested. The robot can walks on sandy inclined surface with amputated leg and based on using legs synchronization up to 15° of inclination. In addition, the speed of robot progress with amputated leg is very low (see table 1).

9.6. Conclusion

This chapter introduced our approaches to enable the hexapod robot to achieve adaptive walking on inclined sandy surface based only on analyzing the joints current consumption signals and joints angular position. The first introduced approach enables our hexapod robot OSCAR to walk on sandy downhill surface. This approach combines our previous approaches, which are the inclination detection approach, the sandy ground detection approach and the control of the leg position on compliant surface approach. The robot has to detect the direction of inclination and to adapt its body posture according to the direction and degree of the inclination. Additionally, the robot has to detect the sandy surface and to execute adaptive walking on the sandy surface. Moreover, The control of the leg position improved the robot walking on the sandy inclined surfaces. Furthermore, the leg amputation has been tested while the robot walks downhill on sandy surface.

The second introduced approach has been demonstrated in this chapter. It enables the hexapod robot to walk on sandy uphill surface, which is considered one of the most difficult terrains. It is based on our approaches for sandy ground detection and control of the robot leg position on compliant surfaces as well as on our new suggested strategy for walking on sandy inclined surface. When the robot detects sandy surface, it changes some parameters and achieves legs synchronization that helps the robot to walk on sandy inclined surface effectively based on our suggested strategy, which is the legs synchronization between the legs that are executing the stance phase. The last introduced approach can be used to enable the hexapod robot to walk on different types of terrains such as sandy inclined surface or sandy flat surface.

CHAPTER 10

Conclusion and Outlook

10.1. Conclusion

The approaches introduced in this work aim to improve the walking behavior of legged machines on different complex rough terrains. This work introduces our approaches that are inspired from biological principles. Four approaches are presented in this work. They are tested on the low-cost hexapod robot OSCAR, which is based on the Bioloid robot kit.

All our approaches are based on monitoring and analyzing of the consumed current and angular position of the servo motors as a proprioception sensing without using any additional external sensors.

The first presented approach enables our hexapod robot OSCAR to detect the inclination direction and to achieve adaptive walking on a detected inclined terrain. In this approach, the consumed current is monitored and analyzed in the distal joint (gamma joint) while the robot walks on flat and inclined surfaces. The robot has to recognize two cases, either the robot walks on flat ground or inclined ground. The current consumption in the $\gamma-$joint of the front and rear legs is monitored and analysed during the stance phase of the supporting legs. As soon as the robot detects inclined ground, it adjusts its body posture. The test cases for the body posture adaptation have been focused on two critical situations of uphill and downhill walking. Other cases such as walking on sidelong inclined terrain have not been studied in this approach. The presented approach enables the hexapod robot to adapt its body posture relevant to the direction and degree of the inclination. The robot continues its mission without using a tilt sensor even if the robot walks on inclined ground. This approach has been implemented in the ORCA architecture. Additionally, the robustness of this approach also has been tested by applying some disturbances such as hitting an object or losing one leg while walking uphill or downhill.

The second presented approach enables the hexapod robot to detect slippery surfaces. A slippery surface is detected by monitoring the consumed current in the α-joint in the front and middle legs during the stance phase. As soon as the robot detects a slippery surface, it reverses its forward walking into backward walking. Then the robot control system changes the direction of the path. This approach enables the robot to avoid walking on slippery surfaces without using additional sensors such as acceleration sensors.

The third presented approach has focused on the detection of the softness and the hardness of the walking substrate, e.g. sandy or hard ground. The softness or hardness of the ground is detected through monitoring and analyzing the current consumption and angular position of the middle joint (β-servo motor) while the leg moves down at the end of the swing phase. The robot adapts its locomotion as soon as it defines the type of the walking terrain. The robot's speed is decreased, the length of the leg's step is shortened and the step height is increased as a result of the robot's adaptation to the sandy surface. The effect of disturbance such as leg amputation on our introduced approach has been tested in the context of this work. The robot can walk on sandy surfaces with an amputated leg. On a sandy surface, the robot walks, in some cases, in a curve due the low friction between the legs and the substrate. Therefore, as soon as the robot detects a sandy surface, the slippery ground detection is deactivated to prevent the robot from confusing sandy and slippery surface, because in both cases, the friction coefficient is low.

The fourth presented approach has focused on the control of the leg position. This approach discusses how the robot improves its walking on compliant surfaces such as sandy or gravel ground. In this approach, each leg adapts its position independent of the other legs. It controls the distance between the robot body and the substrate based on monitoring the current consumption and angular position of the middle joint (β-joint) at the end of the swing phase while it moves down. All of the robots' legs adapt their position in the same way while walking on a compliant surface. This approach prevents the legs from sinking too deeply into the compliant surface and leads to an emergent body motion. The center of mass swings up and down. This motion helps the supporting legs to achieve their stance phase more effectively, especially when the robot walks on a sandy surface.

Finally, our introduced approaches have been tested together while the robot walked on one of the most difficult terrains, which is a sandy inclined surface. In these experiments, the robot walks on downhill and uphill sandy surface, The robot has to detect a sandy and inclined surface as well as to control the leg position while it walks on sandy surface. Additionally, the effect of leg amputation on the robot walking while the robot walks downhill on sandy inclined surface has been tested and discussed .

For walking on an uphill sandy surface, it was noted that our hexapod robot needs a specific strategy to enable the robot to push its body forward in this case. The introduced strategy is based on our approaches for sandy ground detection and control of the robot leg position on compliant surfaces as well as on our new suggested strategy which is legs synchronization between the legs that are executing the stance phase. The combination between our approaches and also the legs synchronization in addition to decreasing the robot body height help the robot to walk on uphill sandy surface effectively.

Our introduced approaches improve the robot walking on different terrains without using any additional sensors, which would increase the cost of the robot and the complexity of the control system. Moreover, these approaches can be used as alternative sensing in the event of a malfunction of the used hardware sensors. In this case the robot can continue its mission even if the used sensor becomes damaged. This work enhances the control system that is based on fault tolerance.

Table 1 shows the summary of our introduced approaches that enable the hexapod robot OSCAR to detect and achieve adaptive walking on different types of terrains. Table 2 shows the reaction of the hexapod robot as soon as it detects the type of walking terrains.

Type of Terrain	**Detection Conditions for one Leg**	**Robot Decision**
Flat Hard Ground	**if** $T_{max} < I^L_\beta(t)$ **and** $\Delta P^L_\beta(t) < D$	Hard ground unless the robot detects another type.
Slippery Ground	$D^L_{slip}(t) = $ true, if $I^L_\alpha(t) < T_{slip}$	$D_{slip}(t) = $ true (slippery ground), if $D^{FL}_{slip}(t)$ and $D^{FR}_{slip}(t)$ and $D^{ML}_{slip}(t)$ and $D^{MR}_{slip}(t)$
Flat Sandy Ground	$D^L_{sand}(t) = $ true, if $T_{min} < I^L_\beta(t) \le T_{max}$ and $\Delta P^L_\beta(t) \ge D$	$D_{sand}(t) = $ true, if $D^{FL}_{sand}(t)$ and $D^{FR}_{sand}(t))$ or $(D^{FL}_{sand}(t)$ and $D^{ML}_{sand}(t))$ or $(D^{FL}_{sand}(t)$ and $D^{MR}_{sand}(t))$ or $(D^{FR}_{sand}(t)$ and $D^{ML}_{sand}(t))$ or $(D^{FR}_{sand}(t)$ and $D^{MR}_{sand}(t))$ or $(D^{ML}_{sand}(t)$ and $D^{MR}_{sand}(t))$
Inclined Hard Ground (Downhill walking)	$D^L_{inc}(t) = $ true, if $I^L_\gamma > T_{down}$	$D_{inc}(t) = $ true, downhill, if $I^{FL}_\gamma(t) > T_{down}$ and $I^{FR}_\gamma(t) > T_{down}$
Inclined Hard Ground (Uphill Walking)	$D^L_{inc}(t) = $ true, if $I^L_\gamma > T_{up}$	$D_{inc}(t) = $ true, uphill, if $I^{HL}_\gamma(t) > T_{up}$ and $I^{HR}_\gamma(t) > T_{up}$
Inclined Sandy Ground	Each leg detects sandy ground and inclined ground based on the above mentioned approaches	The robot decides that the walking terrain is sandy inclined ground based on the combination between above mentioned approaches. The sandy ground detection approach detects sandy surface and the inclined ground detection approach detects the inclined surface.

TABLE 1. The introduced approaches for walking on different types of ground

Type of Terrain	**Reaction**	**Remarks**	**Amputation**
Hard Flat Ground	The leg stops moving down		possible
Slippery Ground	The robot reverses its forward walking into backward walking. This reaction prevents the robot from walking on a slippery surface that maybe leads to unpredictable results and stops the robot from continuing its mission.	The rear legs help the robot to achieve backward walking and to move the robot to the rough surface again.	not tested
Flat Sandy Ground	(1) Reducing the speed of all servo motors to 60%. (2) Increase the step height of each leg. (3) Decreasing the length of the leg's step. (4) Control the leg position.	(1) The slippery surface detection is deactivated. (2) The robot walks sometimes in a curve.	possible
Inclined Hard Ground	Adjust the robot body posture relative to the inclination direction based on changing the β- and γ-servo motor position.	The center of gravity is moved inside the vertical projection of the support feet.	possible up to 20°
Inclined Sandy Ground (Uphill Walking)	(1) Decreasing the height of the robot body. (2) Increasing the overall servo motor speed by 20%. (3) Synchronization between the legs.	(1) The limited degree for acceptable walking is 20°. (2) The control of the body posture due to the inclination detection was ignored in this terrain.	possible up to 20°, walking is very slow
Inclined Sandy Ground (Downhill Walking)	The same reactions for flat sandy ground are used here. Additionally, the control of robot body posture is used.	The limited degree for effective walking is 27°.	possible up to 20°

TABLE 2. Reaction to different types of terrains

10.2. Outlook

Natural terrains contain many cases that a robot has to overcome. This work has focused on the most important cases that may be faced by the walking robot. These approaches could be expanded to enable the walking robot to achieve adaptive walking on other terrains such as walking on sidelong inclined surfaces, walking on inclined graved surfaces or walking on inclined surfaces containing gaps or which are slippery. Moreover, the introduced approaches could be transferred and tested on other robot test platforms.

Bibliography

[1] Logistical vehicle off-road mobility. *Fort Eustis, Va.: U.S. Army Transportation Combat Developments Agency*, (Project TCCO 62-5), 1967.

[2] Alison Abbott. Biological robotics: Working out the bugs. *Nature*, 445(7125):250–253, 2007.

[3] Turgay Akay and Ansgar Büschges. Load signals assist the generation of movement-dependent reflex reversal in the femur–tibia joint of stick insects. *Journal of Neurophysiology*, 96(6):3532–3537, 2006.

[4] Turgay Akay, Björn Ch Ludwar, Marie L Göritz, Josef Schmitz, and Ansgar Büschges. Segment specificity of load signal processing depends on walking direction in the stick insect leg muscle control system. *The Journal of Neuroscience*, 27(12):3285–3294, 2007.

[5] A. Al-Homsy, J. Hartmann, and E. Maehle. Inclination detection and adaptive walking for low-cost six-legged walking robots using organic computing principles. In *Proceedings of the 15th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, pages 173–182. World Scientific, 2012.

[6] A. Al-Homsy, J. Hartmann, and E. Maehle. Slippery and sandy ground detection for hexapod robots based on organic computing principles and somatosensory feedback. In *Robotic and Sensors Environments (ROSE), 2012 IEEE International Symposium on*, pages 43–48, Nov 2012.

[7] A. Al-Homsy, J. Hartmann, and E. Maehle. Adaptive control of leg position for hexapod robot based on somatosensory feedback and organic computing principles. *Proceedings of the 16th International Conference on Climbing and Walking Robots, Sydney, Australia - Nature-Inspired Mobile Robotics*, pages 215–223, 2013.

[8] A. Al-Homsy, J. Hartmann, and E. Maehle. Adaptive walking on uphill sandy surface based on organic computing and somatosensory feedback. *Proceedings of the 18th International Conference on Climbing and Walking Robots, Hangzhou, Zhejiang Province, China - Nature-Inspired Mobile Robotics*, pages 157–166, 2015.

[9] Richard Altendorfer, Ned Moore, Haldun Komsuoglu, Martin Buehler, H Benjamin Brown Jr, Dave McMordie, Uluc Saranli, R Full, and Daniel E Koditschek. RHex: a biologically inspired hexapod runner. *Autonomous Robots*, 11(3):207–213, 2001.

[10] The DLR Crawler: an Actively Compliant Walking Hexapod Robot Online avilable: http://www.dlr.de/rmc/rm/en/desktopdefault.aspx/tabid-6838/.

[11] Adam El Sayed Auf, Marek Litza, and Erik Maehle. Distributed fault-tolerant robot control architecture based on organic computing principles. In *Biologically-Inspired Collaborative Computing*, pages 115–124. Springer, 2008.

[12] Adam-Pharaoun El Sayed Auf. Eine Organic Computing basierte Steuerung für einen hexapoden Laufroboter unter dem Aspekt reaktiver Zuverlässigkeit und Robustheit. Dissertation 2010.

[13] Online available: http://elifesciences.org/content/elife/2/e00231/F14.
[14] The Robotics Research Library. Online available. http://rrlib.cs.uni-kl.de/mca-kl/. http://rrlib.cs.uni-kl.de/mca-kl/projects-using-mca2.
[15] Online available: http://www.bostondynamics.com.
[16] Online available. http://www.lynxmotion.com/c-92-ah3-r.aspx.
[17] Online available: http://www.openservo.org/.
[18] Online available. www.manoonpong.com/AMOSII.html.
[19] Joseph Ayers and Jan Witting. Biomimetic approaches to the control of underwater walking machines. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1850):273–295, 2007.
[20] Joseph Ayers, Jan Witting, Cricket Wilbur, Paul Zavracky, Nicol McGruer, and Donald Massa. Biomimetic robots for shallow water mine countermeasures. In *Proc. of the Autonomous Vehicles in Mine Countermeasures Symposium*, volume 200. Citeseer, 2000.
[21] JE Bares, Martial Hebert, Takeo Kanade, Eric Krotkov, Tom Mitchell, Reid Simmons, and WL Whittaker. Ambler: An autonomous rover for planetary exploration. *Computer*, 22(6):18–26, 1989.
[22] Randall D Beer, Hillel J Chiel, Roger D Quinn, and Roy E Ritzmann. Biorobotic approaches to the study of motor systems. *Current Opinion in Neurobiology*, 8(6):777–782, 1998.
[23] Randall D Beer, Roger D Quinn, Hillel J Chiel, and Roy E Ritzmann. Biologically inspired approaches to robotics: What can we learn from insects? *Communications of the ACM*, 40(3):30–38, 1997.
[24] George A Bekey. *Autonomous robots: from biological inspiration to implementation and control*. MIT press, 2005.
[25] Gary N Boone and Jessica K Hodgins. Reflexive responses to slipping in bipedal running robots. In *Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, volume 3, pages 158–164. IEEE, 1995.
[26] Jürgen Branke, Moez Mnif, Christian Muller-Schloer, and Holger Prothmann. Organic Computing–addressing complexity by controlled self-organization. In *Leveraging Applications of Formal Methods, Verification and Validation, 2006. ISoLA 2006. Second International Symposium on*, pages 185–191. IEEE, 2006.
[27] W. Brockmann, E. Maehle, and F. Mösch. Organic fault-tolerant control architecture for robotic applications. In *Proc. IARP/IEEERAS/EURON Workshop on Dependable Robots in Human Environments*, 2005.
[28] Christopher A Brooks and Karl Iagnemma. Vibration-based terrain classification for planetary exploration rovers. *Robotics, IEEE Transactions on*, 21(6):1185–1191, 2005.
[29] Rodney A Brooks. A robot that walks; emergent behaviors from a carefully evolved network. *Neural Computation*, 1(2):253–262, 1989.
[30] Rodney Allen Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23, 1986.
[31] DG Caldwell and HA Warren. Is there a future for climbing and walking robotic systems in military operations? In *Proceedings CLAWAR, 4th International Conference on Climbing and Walking Robots*, pages 969–977, 2001.
[32] Enric Celaya and Josep M Porta. Control of a six-legged robot walking on abrupt terrain. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 3, pages 2731–2736. IEEE, 1996.

[33] Jorge G Cham, Jonathan K Karpick, and Mark R Cutkosky. Stride period adaptation of a biomimetic running hexapod. *The International Journal of Robotics Research*, 23(2):141–153, 2004.

[34] DFG Priority Program 1183 Organic Computing, SPP 1183, Online available visited Januar 2015: http: //www.organic-computing.de/SPP.

[35] Holk Cruse. The control of body position in the stick insect (carausius morosus), when walking over uneven surfaces. *Biological Cybernetics*, 24(1):25–33, 1976.

[36] Holk Cruse. A new model describing the coordination pattern of the legs of a walking stick insect. *Biological Cybernetics*, 32(2):107–113, 1979.

[37] Holk Cruse. What mechanisms coordinate leg movement in walking arthropods? *Trends in Neurosciences*, 13(1):15–21, 1990.

[38] Holk Cruse, Volker Dürr, and Josef Schmitz. Insect walking is based on a decentralized architecture revealing a simple and robust controller. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1850):221–250, 2007.

[39] Holk Cruse, Volker Dürr, Josef Schmitz, and Axel Schneider. Control of hexapod walking in biological systems. In *Adaptive Motion of Animals and Machines*, pages 17–29. Springer, 2006.

[40] Holk Cruse, Thomas Kindermann, Michael Schumm, Jeffrey Dean, and Josef Schmitz. Walknet: a biologically inspired network to control six-legged walking. *Neural Networks*, 11(7):1435–1447, 1998.

[41] Holk Cruse and A Knauth. Coupling mechanisms between the contralateral legs of a walking insect (carausius morosus). *Journal of Experimental Biology*, 144(1):199–213, 1989.

[42] Holk Cruse, Simone Kühn, Seung Park, and Josef Schmitz. Adaptive control for insect leg position: controller properties depend on substrate compliance. *Journal of Comparative Physiology A*, 190(12):983–991, 2004.

[43] Holk Cruse, D Riemenschneider, and W Stammer. Control of body position of a stick insect standing on uneven surfaces. *Biological Cybernetics*, 61(1):71–77, 1989.

[44] Holk Cruse, Josef Schmitz, U Braun, and A Schweins. Control of body height in a stick insect walking on a treadwheel. *Journal of Experimental Biology*, 181(1):141–155, 1993.

[45] Fred Delcomyn. Biologically inspired robots. *Bioinspiration and Robotics: Walking and Climbing Robots*, pages 279–300, 2007.

[46] EA Devjanin, VS Gurfinkel, EV Gurfinkel, VA Kartashev, AV Lensky, A Yu Shneider, and LG Shtilman. The six-legged walking robot capable of terrain adaptation. *Mechanism and Machine Theory*, 18(4):257–260, 1983.

[47] Rüdiger Dillmann, Jan Albiez, Bernd Gaßmann, Thilo Kerscher, and Marius Zöllner. Biologically inspired walking machines: design, control and perception. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1850):133–151, 2007.

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

[49] Gregory Dudek, Philippe Giguere, Chris Prahacs, Shane Saunderson, Junaed Sattar, Luz Abril Torres-Mendez, Michael Jenkin, Andrew German, Andrew Hogue, Arlene Ripsman, et al. Aqua: An amphibious autonomous robot. *IEEE Computer*, 40(1):46–53, 2007.

[50] Edmond M DuPont, Rodney G Roberts, and Carl A Moore. The identification of terrains for mobile robots using eigenspace and neural network methods. In *Proc. of the Florida Conf. on Recent Advances in Robotics*. Citeseer, 2006.

[51] V. Durr. Stereotypic leg searching movements in the stick insect: kinematic analysis, behavioural context and simulation. *Journal of Experimental Biology*, 204(9):1589–1604, 2001.

[52] Jaak Duysens, F Clarac, Holk Cruse, et al. Load-regulating mechanisms in gait and posture: comparative aspects. *Physiological Reviews*, 80(1):83–134, 2000.

[53] SERGIY DZHANTIMIROV, FRANK PALIS, ULRICH SCHMUCKER, ANDRIY TELESH, and YURIY ZAVGORODNIY. Hil/sil by development of six-legged robot slair2. In *Proceedings of 10th International Conference (CLAWAR 2007)*, pages 16–18. World Scientific, 2007.

[54] A. El Sayed Auf, Florian Mösch, and Marek Litza. How the six-legged walking machine oscar handles leg amputations. In *Workshop on Bio-inspired Cooperative and Adaptive Behaviours in Robots, Rome, Italy*, 2006.

[55] S. Epstein and D. Graham. Behaviour and motor output of stick insects walking on a slippery surface: I. forward walking. *Journal of Experimental Biology*, 105(1):215–229, 1983.

[56] Mustafa Suphi Erden. Six-legged walking machine: the Robot-EA308. Doctoral dissertation, MIDDLE EAST TECHNICAL UNIVERSITY, 2006.

[57] R Franklin, WJ Bell, and R Jander. Rotational locomotion by the cockroach blattella germanica. *Journal of Insect Physiology*, 27:249–255, 1981.

[58] Robert J Full. Integration of individual leg dynamics with whole body movement in arthropod locomotion. In *Proceedings of the workshop on Locomotion Control in Legged Invertebrates on Biological neural networks in invertebrate neuroethology and robotics*, pages 3–20. Academic Press Professional, Inc., 1993.

[59] Kevin C Galloway, Galen Clark Haynes, B Deniz Ilhan, Aaron M Johnson, Ryan Knopf, Goran A Lynch, Benjamin N Plotnick, Mackenzie White, and Daniel E Koditschek. X-RHex: A highly mobile hexapedal robot for sensorimotor tasks. *University of Pennsylvania Scholarly Commons, Department of Electrical and Systems Engineering*, 2010.

[60] E. Garcia, J. Estremera, and P. Gonzalez de Santos. A control architecture for humanitarian-demining legged robots. In *Proceedings of the 6th International Conference on Climbing and Walking Robots (CLAWAR)*, pages 383–390. ULB, 2003.

[61] Philippe Giguere, Gregory Dudek, Shane Saunderson, and Chris Prahacs. Environment identification for a running robot using inertial and actuator cues. In *Robotics: Science and Systems*, 2006.

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

[63] M. Gorner, T. Wimbock, Andreas Baumann, Matthias Fuchs, Thomas Bahls, Markus Grebenstein, Christoph Borst, Jörg Butterfass, and Gerd Hirzinger. The DLR-crawler: a testbed for actively compliant hexapod walking based on the fingers of DLR-Hand II. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 1525–1531. IEEE, 2008.

[64] D. Graham. Pattern and control of walking in insects. *Advances in Insect Physiology*, 18:31–140, 1985.

[65] D. Graham and Holk Cruse. Coordinated walking of stick insects on a mercury surface. *Journal of Experimental Biology*, 92(1):229–241, 1981.
[66] D. Graham and S. Epstein. Behaviour and motor output for an insect walking on a slippery surface: II. Backward walking. *Journal of Experimental Biology*, 118(1):287–296, 1985.
[67] M Görner, Th Wimböck, and G Hirzinger. The DLR crawler: evaluation of gaits and control of an actively compliant six-legged walking robot. *Industrial Robot: An International Journal*, 36(4):344–351, 2009.
[68] Matthias Gruhn, Lyuba Zehl, and Ansgar Büschges. Straight walking and turning on a slippery surface. *Journal of Experimental Biology*, 212(2):194–209, 2009.
[69] Steffen Haidacher, Joerg Butterfass, Max Fischer, Markus Grebenstein, Klaus Jöhl, Klaus Kunze, M Nickl, Nikolaus Seitz, and Gerd Hirzinger. DLR hand ii: Hard-and software architecture for information processing. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 1, pages 684–689. IEEE, 2003.
[70] Freyr Hardarson. Locomotion for difficult terrain. *Dept. Mach. Des., Royal Inst. Technol., Stockholm, Sweden, Tech. Rep. TRITA-MMK*, 3:1400–1179, 1998.
[71] Hitoshi Hasunuma, Katsumi Nakashima, Masami Kobayashi, Fumisato Mifune, Yoshitaka Yanagihara, Takao Ueno, Kazuhisa Ohya, and Kazuhito Yokoi. A tele-operated humanoid robot drives a backhoe. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 3, pages 2998–3004. IEEE, 2003.
[72] Paul Horn. Autonomic computing: IBM perspective on the state of information technology. 2001.
[73] http://www.fzi.de/forschung/projekte/kairo-ii/
http://www.fzi.de/en/forschung/projekte/lauron/.
[74] http://www.uni magdeburg.de/ieat/robotslab/.
[75] Markus C Huebscher and Julie A McCann. A survey of autonomic computing: degrees, models, and applications. *ACM Computing Surveys (CSUR)*, 40(3):7, 2008.
[76] Bojan Jakimovski. Biologically inspired approaches for locomotion of a hexapod robot oscar. In *Biologically Inspired Approaches for Locomotion, Anomaly Detection and Reconfiguration for Walking Robots*. Springer, 2011.
[77] Bojan Jakimovski, Marek Litza, Florian Mösch, and Adam El Sayed Auf. Development of an organic computing architecture for robot control. In *GI Jahrestagung (1)*, pages 145–152, 2006.
[78] Bojan Jakimovski, Benjamin Meyer, and Erik Maehle. Self-reconfiguring hexapod robot OSCAR using organically inspired approaches and innovative robot leg amputation mechanism. In *International Conference on Automation, Robotics and Control Systems, ARCS-09, Orlando, USA*, 2009.
[79] Brett Kennedy, Avi Okon, Hrand Aghazarian, Mircea Badescu, Xiaoqi Bao, Yoseph Bar-Cohen, Zensheu Chang, Borna E Dabiri, Mike Garrett, Lee Magnone, et al. Lemur IIb: a robotic system for steep terrain access. *Industrial Robot: An International Journal*, 33(4):265–269, 2006.
[80] Brett Kennedy, Avi Okon, Hrand Aghazarian, Mike Garrett, Terry Huntsberger, Lee Magnone, Matthew Robinson, and Julie Townsend. The Lemur II-class robots for inspection and maintenance of orbital structures: a system description. In *Climbing and Walking Robots*, pages 1069–1076. Springer, 2006.

[81] Jeffrey O Kephart and David M Chess. The vision of autonomic computing. *Computer*, 36(1):41–50, 2003.

[82] Fumitaka Kikuchi, Yusuke Ota, and Shigeo Hirose. Basic performance experiments for jumping quadruped. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 4, pages 3378–3383. IEEE, 2003.

[83] Haldun Komsuoglu, Dave McMordie, Uluc Saranli, Ned Moore, Martin Buehler, and Daniel E Koditschek. Proprioception based behavioral advances in a hexapod robot. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 4, pages 3650–3655. IEEE, 2001.

[84] Motohide Konaka. National project on advanced robot technology in japan. In *Advanced Robotics, 1991.'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, pages 24–31. IEEE, 1991.

[85] M. Konyev, F. Palis, Y. Zavgorodniy, A. Melnikov, A. Rudskiy, A. Telesh, U. SCHMUCKER, and V. Rusin. Walking robot ANTON: Design, Simulation, Experiments. In *Proc. of 11th Int. Conf. on Climbing and Walking Robots (CLAWAR)*, pages 922–929, 2008.

[86] Noriho Koyachi, Hironori Adachi, Makoto Izumi, and Takeshi Hirose. Control of walk and manipulation by a hexapod with integrated limb mechanism: Melmantis-1. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 4, pages 3553–3558. IEEE, 2002.

[87] Andre F Krause and Volker Dürr. Tactile efficiency of insect antennae with two hinge joints. *Biological Cybernetics*, 91(3):168–181, 2004.

[88] William A Lewinger and Roger D Quinn. A hexapod walks over irregular terrain using a controller adapted from an insect's nervous system. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3386–3391. IEEE, 2010.

[89] Chen Li, Paul B Umbanhowar, Haldun Komsuoglu, Daniel E Koditschek, and Daniel I Goldman. Sensitive dependence of the motion of a legged robot on granular media. *Proceedings of the National Academy of Sciences*, 106(9):3029–3034, 2009.

[90] RA Liston. Increasing vehicle agility by legs: The quadruped transporter. In *38th National Meeting of the Operations Research Society of America*, volume 143, 1970.

[91] RA Liston and RS Mosher. A versatile walking truck. In *Proceedings of the Transportation Engineering Conference*, 1968.

[92] Yasushi Mae, Atsushi Yoshida, Tatsuo Arai, Kenji Inoue, Kunio Miyawaki, and Hironori Adachi. Application of locomotive robot to rescue tasks. In *Intelligent Robots and Systems, 2000.(IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, volume 3, pages 2083–2088. IEEE, 2000.

[93] Erik Maehle, Werner Brockmann, Karl-Erwin Grosspietsch, Adam El Sayed Auf, Bojan Jakimovski, Stephan Krannich, Marek Litza, Raphael Maas, and Ahmad Al-Homsy. Application of the organic robot control architecture orca to the six-legged walking robot oscar. In *Organic Computing: A Paradigm Shift for Complex Systems*, pages 517–530. Springer, 2011.

[94] RB McGhee. Vehicular legged locomotion in advances in automation and robotics, ed. GN Saridis, 1984.

[95] RB McGhee, DE Orin, DR Pugh, and MR Patterson. A hierarchically structured system for computer control of a hexapod walking machine. In *Theory and Practice of Robots and Manipulators*, pages 375–381. Springer, 1985.

[96] Robert B McGhee and Geoffrey I Iswandhi. Adaptive locomotion of a multi-legged robot over rough terrain. *Systems, Man and Cybernetics, IEEE Transactions on*, 9(4):176–182, 1979.

[97] D. McMordle and M. Buehler. Towards pronking with a hexapod robot. Technical report, 4th Int. Conf. on Climbing and Walking Robots, Karlsruhe, Germany, 2001.

[98] Hamid Mehdigholi and Saeed Akbarnejad. Optimization of watt's six-bar linkage to generate straight and parallel leg motion. *International Journal of Advanced Robotic Systems*, 9, 2012.

[99] Christian Müller-Schloer, Hartmut Schmeck, and Theo Ungerer. *Organic Computing: a paradigm shift for complex systems.* Springer, 2011.

[100] EZ Moore, D Campbell, Felix Grimminger, and Martin Buehler. Reliable stair climbing in the simple hexapod 'RHex'. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 3, pages 2222–2227. IEEE, 2002.

[101] RA Morrison. Iron mule train. In *Mobility testing: proceedings of the off-road Mobility Research Symposium*, pages 381–400, 1968.

[102] Florian Mösch, Marek Litza, Adam El Sayed Auf, Bojan Jakimovski, Erik Maehle, and Werner Brockmann. Organic fault-tolerant controller for the walking robot oscar. In *Architecture of Computing Systems (ARCS), 2007 20th International Conference on*, pages 1–5. VDE, 2007.

[103] Florian Mösch, Marek Litza, Adam El Sayed Auf, Erik Maehle, Karl E Großpietsch, and Werner Brockmann. ORCA - towards an organic robotic control architecture. In *Self-Organizing Systems*, pages 251–253. Springer, 2006.

[104] C. Muller-Schloer. Organic computing-on the feasibility of controlled emergence. In *Hardware/Software Codesign and System Synthesis, 2004. CODES+ ISSS 2004. International Conference on*, pages 2–5. IEEE, 2004.

[105] Mohammad Reza Nami and Mohsen Sharifi. A survey of autonomic computing systems. In *Intelligent Information Processing III*, pages 101–110. Springer, 2007.

[106] Gabriel M Nelson and Roger D Quinn. Posture control of a cockroach-like robot. *Control Systems, IEEE*, 19(2):9–14, 1999.

[107] Neil Neville, Martin Buehler, and Inna Sharf. A bipedal running robot with one actuator per leg. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 848–853. IEEE, 2006.

[108] Takashi Nishiyama, Hiroshi Hoshino, Kazuya Sawada, Yoshihiko Tokunaga, Hirotatsu Shinomiya, Mitsunori Yoneda, Ikuo Takeuchi, Yukiko Ichige, Shizuko Hattori, and Atsuo Takanishi. Development of user interface for humanoid service robot system. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 3, pages 2979–2984. IEEE, 2003.

[109] K. Nonami and QJ. Huang. Humanitarian mine detection six-legged walking robot comet-ii with two manipulators. In *Proceedings of the Third International Conference on Climbing and Walking Robots (CLAWAR2001)*, pages 989–996, 2001.

[110] Toru Omata, Kazuo Tsukagoshi, and Osamu Mori. Whole quadruped manipulation. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 2, pages 2028–2033. IEEE, 2002.

[111] Available online: http://kodlab.seas.upenn.edu/RHex/Home.

[112] Modular Controller Architecture Version 2. Online website visited April 2014, http://www.mca2.org/introduction.html.

[113] MAYUR PALANKAR and III LUTHER PALMER. A force threshold-based position controller for legged locomotion over irregular terrain. In *Proceedings of the Fifteenth International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, pages 563–570. World Scientific, 2012.

[114] Keir G Pearson. Generating the walking gait: role of sensory feedback. *Progress in Brain Research*, 143:123–129, 2004.

[115] KG Pearson. Central programming and reflex control of walking in the cockroach. *Journal of Experimental Biology*, 56(1):173–193, 1972.

[116] KG Pearson and R Franklin. Characteristics of leg movements and patterns of coordination in locusts walking on rough terrain. *The International Journal of Robotics Research*, 3(2):101–112, 1984.

[117] Dennis R Pugh, Eric A Ribble, Vincent J Vohnout, Thomas E Bihari, Thomas M Walliser, Mark R Patterson, and Kenneth J Waldron. Technical description of the adaptive suspension vehicle. *The International Journal of Robotics Research*, 9(2):24–42, 1990.

[118] Marc H Raibert. Legged robots. *Communications of the ACM*, 29(6):499–514, 1986.

[119] Marc H Raibert. Foreword. *The International Journal of Robotics Research*, 9(2):2–3, 1990.

[120] Christian Ridderström. Legged locomotion: Balance, control and tools-from equation to action. 2003.

[121] Christian Ridderstrom and Johan Ingvast. Quadruped posture control based on simple force distribution-a notion and a trial. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 4, pages 2326–2331. IEEE, 2001.

[122] Roy E Ritzmann, Roger D Quinn, James T Watson, and Sasha N Zill. Insect walking and biorobotics: A relationship with mutual benefits. *Bioscience*, 50(1):23–33, 2000.

[123] Jonathan M Roberts, Peter I Corke, and Graeme J Winstanley. Development of a 3500-tonne field robot. *The International Journal of Robotics Research*, 18(7):739–752, 1999.

[124] A Roennau, G Heppner, L Pfotzer, and R Dillmann. LAURON V: Optimized leg configuration for the design of a bio-inspired walking robot. pages 563–570, 2013.

[125] Mark E Rosheim. In the footsteps of leonardo [articulated anthropomorphic robot]. *Robotics and Automation Magazine, IEEE*, 4(2):12–14, 1997.

[126] Mark E Rosheim. Leonardo's lost robots. 2006.

[127] M Russell. Odex I: The first functionoid. *Robotics Age*, 5(5):12–18, 1983.

[128] SJ Russell and P Norvig. Artificial intelligence: A modern approach. 2002.

[129] Yoshiaki Sakagami, Ryujin Watanabe, Chiaki Aoyama, Shinichi Matsunaga, Nobuo Higaki, and Kikuo Fujimura. The intelligent ASIMO: System overview and integration. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 3, pages 2478–2483. IEEE, 2002.

[130] Uluc Saranli, Akin Avci, and MC Oeztuerk. A modular real-time fieldbus architecture for mobile robotic platforms. *Instrumentation and Measurement, IEEE Transactions on*, 60(3):916–927, 2011.

[131] Uluc Saranli, Martin Buehler, and Daniel E Koditschek. RHex: A simple and highly mobile hexapod robot. *The International Journal of Robotics Research*, 20(7):616–631, 2001.

[132] Uluc Saranli, Alfred A Rizzi, and Daniel E Koditschek. Model-based dynamic self-righting maneuvers for a hexapedal robot. *The International Journal of Robotics Research*, 23(9):903–918, 2004.

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

[134] Josef Schmitz, Axel Schneider, Malte Schilling, and Holk Cruse. No need for a body model: positive velocity feedback for the control of an 18-dof robot walker. *Applied Bionics and Biomechanics*, 5(3):135–147, 2008.

[135] U. Schmucker, A. Schneider, and T. Ihme. Sechsbeiniger Schreitroboter mit Kraftrückführung-Konzeption und erste Ergebnisse. In *Brandenburger Workshop Mechatronik, Brandenburg*, pages 45–53, 1994.

[136] Axel Schneider, Holk Cruse, and Josef Schmitz. Decentralized control of elastic limbs in closed kinematic chains. *The International Journal of Robotics Research*, 25(9):913–930, 2006.

[137] Kay-Ulrich Scholl, Jan Albiez, and Bernd Gassmann. Mca-an expandable modular controller architecture. In *3rd Real-Time Linux Workshop*, 2001.

[138] JE Shigley. The mechanics of walking vehicles, land locomotion laboratory report no. ll-71. *US Army Tank-automotive Command, Warren, Michigan (September 1960)*, 1960.

[139] Manuel F Silva and JA Tenreiro Machado. A historical perspective of legged robots. *Journal of Vibration and Control*, 13(9-10):1447–1486, 2007.

[140] Silke Steingrube, Marc Timme, Florentin Wörgötter, and Poramate Manoonpong. Self-organized adaptation of a simple neural circuit enables complex robot behaviour. *Nature Physics*, 6(3):224–230, 2010.

[141] Yuuya Takahashi, Tatsuo Arai, Yasushi Mae, Kenji Inoue, and Noriho Koyachi. Development of multi-limb robot with omnidirectional manipulability and mobility. In *Intelligent Robots and Systems, 2000.(IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, volume 3, pages 2012–2017. IEEE, 2000.

[142] Hiroshi Takemura, Masato Deguchi, Jun Ueda, Yoshio Matsumoto, and Tsukasa Ogasawara. Slip-adaptive walk of quadruped robot. *Robotics and Autonomous Systems*, 53(2):124–141, 2005.

[143] Klaus Uhl and Marco Ziegenmeyer. MCA2–an extensible modular framework for robot control applications. In *10th International Conference (CLAWAR 2007)*. World Scientific, 2007.

[144] Online Available visited in April 2014. http://support.robotis.com/en/.

[145] Online Available visited in April 2014. http://www.earthlife.net/insects/anat-thorax.html.

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

[147] Barbara Webb. Validating biorobotic models. *Journal of Neural Engineering*, 3(3):R25, 2006.

[148] Joel D Weingarten, Daniel E Koditschek, Haldun Komsuoglu, and Chris Massey. Robotics as the delivery vehicle: A contexualized, social, self paced, engineering education for life-long learners. *in Robotics Science and Systems Workshop on "Research in Robots for Education*, 2007.

[149] Christian Weiss, Nikolas Fechner, Matthias Stark, and Andreas Zell. Comparison of different approaches to vibration-based terrain classification. In *EMCR*, 2007.

[150] David Wettergreen, Chuck Thorpe, and Red Whittaker. Exploring mount erebus by walking robot. *Robotics and Autonomous Systems*, 11(3):171–185, 1993.

[151] Norbert Wiener. Cybernetics: Or, control and communication in the animal and the machine, 1961.

[152] H. Yamamoto and K. Ohnishi. An approach to stable walking on unknown slippery floor for biped robot. In *Industrial Electronics Society, 2001. IECON'01. The 27th Annual Conference of the IEEE*, volume 3, pages 1728–1733. IEEE, 2001.

[153] Kazuhiko Yokoyama, Hiroyuki Handa, Takakatsu Isozumi, Yutaro Fukase, Kenji Kaneko, Fumio Kanehiro, Yoshihiro Kawai, Fumiaki Tomita, and Hirohisa Hirukawa. Cooperative works by a human and a humanoid robot. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 3, pages 2985–2991. IEEE, 2003.

[154] Steffen Zenker, Eren Erdal Aksoy, Dennis Goldschmidt, Florentin Worgotter, and Poramate Manoonpong. Visual terrain classification for selecting energy efficient gaits of a hexapod robot. In *Advanced Intelligent Mechatronics (AIM), 2013 IEEE/ASME International Conference on*, pages 577–584. IEEE, 2013.

Own Publications

- A. Al-Homsy, J. Hartmann, and E. Maehle. Adaptive Walking on Uphill Sandy Surface Based on Organic Computing and Somatosensory Feedback. Proceedings of the 18th International Conference on Climbing and Walking Robots, Hangzhou, Zhejiang Province, China - Nature-Inspired Mobile Robotics, pages 157-167, 2015.
- A. Al-Homsy, J. Hartmann, and E. Maehle. Adaptive control of leg position for hexapod robot based on somatosensory feedback and organic computing principles. Proceedings of the 16th International Conference on Climbing and Walking Robots, Sydney, Australia - Nature-Inspired Mobile Robotics, pages 215-223, 2013.
- A. Al-Homsy, J. Hartmann, and E. Maehle. Inclination detection and adaptive walking for low-cost six-legged walking robots using organic computing principles. In Proceedings of the 15th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, pages 173-182. World Scientic, 2012.
- A. Al-Homsy, J. Hartmann, and E. Maehle. Slippery and sandy ground detection for hexapod robots based on organic computing principles and somatosensory feedback. In Robotic and Sensors Environments (ROSE), 2012 IEEE International Symposium on, pages 43-48, Nov 2012.
- Erik Maehle, Werner Brockmann, Karl-Erwin Grosspietsch, Adam El Sayed Auf, Bojan Jakimovski, Stephan Krannich, Marek Litza, Raphael Maas, and Ahmad Al-Homsy. Application of the organic robot control architecture ORCA to the six-legged walking robot OSCAR. In Organic Computing: A Paradigm Shift for Complex Systems, pages 517-530. Springer, 2011.