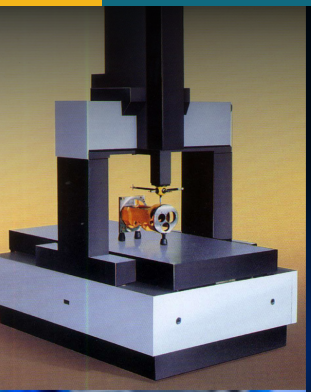
AUTOMATION AND CONTROL COLLECTION



Mobile Robots for Dynamic Environments

Emin Faruk Kececi
Marco Ceccarelli

MOMENTUM PRESS
ENGINEERING

MOBILE ROBOTS FOR DYNAMIC ENVIRONMENTS

MOBILE ROBOTS FOR DYNAMIC ENVIRONMENTS

EMIN FARUK KECECI
MARCO CECCARELLI

MOMENTUM PRESS
ENGINEERING

MOMENTUM PRESS, LLC, NEW YORK

First published by Momentum Press®, LLC
222 East 46th Street, New York, NY 10017
www.momentumpress.net

ISBN-13: 978-1-60650-821-3 (print)
ISBN-13: 978-1-60650-822-0 (e-book)

Momentum Press Automation and Control Collection

Cover and interior design by Exeter Premedia Services Private Ltd., Chennai, India

10 9 8 7 6 5 4 3 2 1

Printed in the United States of America

Contributing Authors

Editors

Emin Faruk Kececi
Istanbul Technical University, Department of Mechanical Engineering, Turkey

Marco Ceccarelli
LARM: Laboratory of Robotics and Mechatronics, University of Cassino and South Latium, Italy

Authors

Andrea Manuello Bertetto
Department of Mechanical, Chemical and Materials Engineering, University of Cagliari, Italy

Ruxu Du and Zheng Li
Institute of Precision Engineering, The Chinese University of Hong Kong, China

Cihat Bora Yiğit and Erdinc Altug
Istanbul Technical University, Department of Mechanical Engineering, Faculty of Mechanical Engineering, Turkey

Raul Ordonez
Dept. of Electrical and Computer Engineering, University of Dayton, Dayton, Ohio

Lino Marques
Dept. of Electrical and Computer Engineering, Institute of Systems and Robotics, University of Coimbra, Portugal

Baris Fidan
University of Waterloo, Dept. of Mechanical and Mechatronics Engineering, Canada

Veysel Gazi
Dept. of Electrical and Electronics Engineering, Istanbul Kemerburgaz University, Mahmutbey, Turkey

Filippo Bonaccorso, Carmelo Donato Melita, and Giovanni Muscato
Dipartimento di Ingegneria Elettrica Elettronica e Informatica (DIEEI), Università degli Studi di Catania, Italy

Michele Prestifilippo
Istituto Nazionale di Geofisica e Vulcanologia (INGV), Sezione di CATANIA, Italy

Contents

Contributing Authors v
Preface xi
1. Underwater robots: a fascinating challenge 1
Andrea Manuello
1.1 Introduction 1
1.2 Biomorfic fin propulsion 8
1.3 Biomorfic fin actuation 10
1.4 Biomorfic integrated fin-actuator systems: a case study 11
1.5 Conclusion 20
1.6 References 20
2. A novel Lighter Than Air Vehicle – Flying Octopus 27
Zheng Li
2.1 Introduction 27
2.2 The design of Flying Octopus – a bio-inspired design 31
2.2.1 Design considerations 31
2.2.2 The design 32
2.2.3 The wire-driven flapping wing 33
2.3 Modeling and simulation 36
2.3.1 Wing kinematics 36
2.3.2 Propulsion model 39
2.3.3 Propulsion simulation 40
2.4 Motion control 42
2.4.1 Wing flapping motion 42
2.4.2 Flying Octopus motion control 44
2.5 Prototype and experiment testing 46
2.5.1 Flying Octopus prototype 46
2.5.2 Indoor experiments 47
2.6 Conclusions 51
2.7 References 52
3. Visual attitude estimation and stabilization of flying robots 55
Cihat Bora Yiğit
3.1 Unmanned Aerial Vehicles 55
3.2 Attitude estimation with vision 58

3.3 Quadrotor UAV modeling and control 61
3.4 Robot design and manufacturing 66
3.5 Experiments 70
3.6 Closure 74
3.7 Acknowledgment 75
3.8 References 75
4. Robot swarms: dynamics and control 79
Veysel Gazi
4.1 Introduction 79
4.2 Agent dynamics 81
4.2.1 Fully actuated agent model 81
4.2.2 Non-holonomic agent dynamics 83
4.2.3 Simplified or high-level agent models 85
4.3 Problem definitions 86
4.3.1 Aggregation and social foraging 86
4.3.2 Formal control and swarm tracking 89
4.3.3 Source seeking 91
4.4 Control design approaches 93
4.4.1 Artificial potential functions 93
4.4.2 Neighborhood topologies 97
4.4.3 Gradient-based, lyapunov, and sliding mode methods 97
4.4.4 Adaptive control approaches 101
4.4.5 Other nonlinear methods 101
4.5 Swarm robotic applications 102
4.5.1 Static coverage 103
4.5.2 Dynamic coverage 104
4.5.3 Cooperative target localization and tracking 106
4.6 Concluding remarks 107
4.7 References 107
5. Mobile robots for earth exploration: applications, technologies and image processing techniques for navigation 127
Filippo Bonaccorso
5.1 Introduction 127
5.2 Applications of robots for earth explorations 128
5.2.1 Volcanic explorations 128
5.2.2 Meteorite search 130

5.2.3 Search and rescue 131
5.2.4 Humanitarian demining 131
5.2.5 Underground explorations 131
5.3 Related technologies 132
5.3.1 Sun synchronous robots 132
5.3.2 Traversability analysis 132
5.3.3 Localization and map building 133
5.3.4 Traction control 133
5.4 Current challenges 133
5.5 A road detection and obstacle avoidance method of using a stereo camera for autonomous navigation 134
5.5.1 Related works and overview 135
5.5.2 Drivable surface detection outline 136
5.5.3 Drivable surface detection setup 142
5.5.4 Obstacle detection 148
5.5.5 Control of the robotic platform 150
5.5.6 Final considerations 157
5.6 Conclusions and future work 157
5.7 References 158

Preface

For several decades now, mobile robots have been integral to the development of new robotic systems for new applications, even in non-technical areas. Mobile robots have already been developed for such uses as industrial automation, medical care, space exploration, demining operations, surveillance, entertainment, museum guides and many other industrial and non-industrial applications. In some cases these products are readily available on the market. A considerable amount of literature is also available; not all of which pertains to technical issues, as listed in the chapters of this book.

Mobile robots will always be further developed with the goal of performing locomotion tasks, those related to movement and interaction with the surrounding environment, within which a task can be fulfilled even without the supervision of human operators. The complexity of locomotion requires different solutions both for design and operation. As such, a large variety of mobile robots and mobile robotic systems has been, and still can be, developed. In fact, considerable advancements have been achieved within the last few decades, and a vast amount of literature is already available detailing a large variety of mobile robots. The literature emphasizes design issues, operational success, procedures and algorithms that can be used specifically for these applications, as opposed to general approaches for a variety of cases.

One key point for mobile robots is interaction with the environment in which the mobile robot moves and corresponding solutions can determine the success or failure of the motion. Indeed, the mechanical design is not very often considered a critical issue, but rather it is often included as an issue in the overall design of mechanical solutions within servo-controlled operation and environment interaction. A second important issue is the acceptance of robotic systems and the corresponding psychological aspects, when robots are proposed to operators and users in fields with very low levels of technical means in their current work practice.

These two subjects are the core of the discussions in this book and its companion volume, *Designs and Prototypes of Mobile Robots* (available separately from ASME Press), which aims to illustrate not only the potential but also the problems for the dissemination of mobile robots and mobile robotic systems in all human activities with service aims.

Authors have been invited from all over the world and chapters have been selected after review as to approach the most challenging aspects and applications of mobile robotic systems, with the aim to survey the current state-of-the-art and its future potential.

We believe that readers will enjoy this book and its companion, and will utilize the knowledge gained with satisfaction and will be assisted by its content in their interdisciplinary work for engineering developments of mobile robots, in both old and new applications. This book and its companion can be used as a graduate level course books or guide books for the practicing engineer who is working on a specific problem which is described in one of the chapters.

We are grateful to the authors of the chapters for their valuable contributions and for preparing their manuscripts on time. Also acknowledged is the professional assistance by the staff of ASME Press and especially by Dr. Vladimir Vantsevich, who has enthusiastically supported this book project, as the Robotics Series Editor.

We are also thankful to our families for their patience and understanding, without which the realization of this book and its companion, with the input of so many experts from different fields and countries, might not be possible. Moreover, our students hard and dedicated work taught us a lot.

Dr. Marco Ceccarelli and Dr. E. Faruk Kececi
Cassino, Italy and Istanbul, Turkey - July 2014

1. Underwater robots: a fascinating challenge

Andrea Manuello Bertetto, University of Cagliari, Italy

Abstract: This chapter provides an overview of underwater robots and describes a method of generating aquatic propulsion, particularly interesting for underwater robots. The design and applications of biomorphic devices are described and illustrated. These design choices represent an important way to achieve attractive performance in navigation.

1.1 Introduction

The history of the exploration of the underwater world has its roots in human curiosity towards nature. Underwater exploration is one of the most exciting endeavours in the world, as the ocean is the last frontier whose depths are still largely out of reach: less than one per cent of the Earth's sea floor has been explored. Although we can move through the air and travel to space, where we have a sweeping view of the ocean's surface, a detailed observation of the ocean's depths relies on submersible ships.

The Phoenicians travelled across the Mediterranean and through the Pillars of Hercules (Straits of Gibraltar), which were considered the ends of the earth. From there, they sailed to Britain and probably around Africa two thousand years before Vasco da Gama. Furthermore, there are plenty of mythological references to the sea fascinating, even in the Homeric poems Iliad and Odyssey. Aristotle (fourth century BC) is referred to as the father of marine biology: he identified a variety of species and marine creatures (i.e., recognizing that whales are mammals).

The design of systems and devices, including submarines, accompanied the human curiosity and attraction towards the deep sea. One of the earliest types of equipment for underwater work and exploration was the diving bell [1, 2].

Aristotle referred to the use of a diving bell in the 4th century BC, describing systems that allow for breathing underwater, via a dome toward the bass creating a bubble of trapped air. In the opinion of Roger Bacon, Alexander the Great made several dives to explore the deep Mediterranean Sea on the authority of Ethicus the astronomer [3].

The Italian scientist Guglielmo de Lorena designed, realised and used what is considered to be the first modern diving bell. Based on drawings by Leonardo da Vinci, from 1531 to 1535 he designed and built a large bell in which air was stored, allowing the person inside to breathe. The bell had a pipe from the surface into the bell supplying fresh air in so the inhabitant could breathe. It's early use was for commercial sponge fishing and salvaging sunken ships, for treasure or otherwise. In 1531 Guglielmo de Lorena himself used his bell, diving on two sunken galleys of the Roman emperor Caligula to explore them. The two ships, located on the bottom of the Lake of Nemi near Rome, were 70 meters long and more than 25 wide, and were built by Caligula in the first century BC in honour of the Egyptian goddess Isis and the goddess Diana, local patron of the hunt.

Caligula used them as pleasure galleys, floating palaces to live on and to simulate naval battles. After his death in 41 AD, the Roman Senate destroyed all the works of Caligula, including the Nemi ships, which were sunk to the bottom of the lake. The history of the ships, which were said to have carried treasure, was soon to become a legend.

The invention of Guglielmo de Lorena, which covered the diver's head and torso, was lowered through ropes. Thus, the diver could walk the lake bottom for about an hour before he had to rise to the water's surface to breathe.

In nature, an example of an underwater diving bell is the Argyroneta Aquatica, also called the diving bell spider or water spider, that is the only species of spider known to live entirely underwater. Argyroneta, means spinner of silver, and comes from the colour given by the air bubble where the spider lives (in ancient Greek the word means silver and spinner). This spider lives in Europe and Asia. It breathes air, which it traps in a bubble held by hairs on its abdomen and legs. The air supply is unnecessary in well-oxygenated water: the bell permits gas exchange with the surrounding water. There is net diffusion of oxygen into the bell and net diffusion of carbon dioxide out by differences in partial pressure. These animals may also supply air once every 24 hours [4, 5].

Underwater marine exploration is really a difficult challenge. The ocean is vast and the deep seas are dark, cold, and subject to high-pressure. Guglielmo de Lorena's pioneering attempt to improve diving efficiencies evolved over the centuries until the introduction of

human occupied vehicles (HOVs), formally called manned submersibles, which appeared to be the solution to conquering the deep for a short time in history, carrying people who make direct observations of the deep seas. Manned submersibles could not stay underwater long, however, and crew safety was always an important consideration.

It seemed that HOVs would allow deeper work for longer periods of time, but they had many of the same disadvantages as hyperbaric diving. The main problem was that HOVs required substantial dedicated support vessels, and still endangered humans underwater. They also were slow to launch and recover and had limited bottom time, making them economically inconvenient.

The introduction of robotic submersibles in the mid-seventies, formally called unmanned underwater vehicles, represented a safer and more efficient alternative to sending humans down into the depths, relegating HOVs to limited use in the science and the tourism industries.

Robotic submersibles include non-autonomous remotely operated vehicles (ROVs) and autonomous underwater vehicles (AUVs). Underwater visibility often prevents human personnel from performing research directly, whereas robots can break the darkness of the ocean to explore safely, and are priceless for rescue missions. ROVs are robotic submarines that are controlled and powered from the surface by an operator aboard a ship through a tether cable. The tether cable is an armored cable that contains a group of electrical conductors and fibre optics that carry electrical power, video, and data signals back and forth between the operator and the ROV. Once at the ROV, the electrical power is split and distributed among the different components of the ROV. In high power applications, most of the electrical power is used to drive a high-powered electrical motor which drives a hydraulic pump. The hydraulic pump is then used for propulsion and to power equipment such as torque tools and manipulator arms, where electrical motors would be too difficult to implement undersea. ROVs are common in deep water industries such as offshore hydrocarbon extraction, and are normally classified into categories based on their size, weight, ability or power. For example, micro-class ROVs are very small in size and weight (less than 3 kg), and are used as an alternative to a diver, specifically in places a diver might not be able to physically enter, such as a sewer, pipeline or small cavity.

Mini-class ROVs (around 15 kg) are also used as a diver alternative; in this case, one person may be able to transport the complete ROV

system out with them on a small boat, deploy it and complete the job without outside help. Light work class ROVs have propulsion systems delivering up to 5 HP, and occasionally they are equipped with small three finger manipulator grippers. These ROVs may be able to carry a sonar unit and are usually used on light survey applications, with a maximum working depth smaller than 1000 meters. Besides these types of ROVs, mainly used for inspection and research purposes, there are also large working class ROVs, which are a necessity to any major subsea endeavour, like digging a trench on the seafloor, laying communication cable, placing sensitive oceanographic equipment, or salvaging lost items [6].

ROV technology proved to be good for offshore and military applications. However, the problems of continuous control and monitoring the remotely operated vehicle by using a cable forced researchers to develop a new technology to create a device able to complete its mission autonomously. Such thinking resulted in the idea of developing Autonomous Underwater Vehicles (AUVs). AUVs are robots programmed at the sea surface, then released to travel through the water on their own, following certain trajectories in the 3D space at defined speed and depth.

Admittedly, though AUV technology has gained popularity in recent years, the idea of AUV was not new, and probably originated from the Whitehead Fish Torpedo. Robert Whitehead is credited with designing, building, and demonstrating the first torpedo in Austria in 1866 [7, 8].

The first AUV was developed at the Applied Physics Laboratory at the University of Washington as early as 1957. The "Special Purpose Underwater Research Vehicles," or SPURV, were used to study diffusion, acoustic transmission and submarine wakes in the Arctic regions. Other early AUVs were developed at the Massachusetts Institute of Technology in the 1970s. One of these is on display in the Hart Nautical Gallery in MIT. At the same time, AUVs were also developed in the Soviet Union (although this was only commonly known much later).

Until relatively recently, AUVs have been used for a limited number of tasks, constrained by the available technology. Such tasks, requiring only a list of pre-programmed instructions to be accomplished, seemed not to demand a high level of intelligent behaviour.

With the development of more advanced processing capabilities and high-yield power supplies, AUVs began to be used for an

increasing number of tasks, with roles and missions continuously evolving. Contextually, related efforts were made to develop AUV systems which were more intelligent and better able to adapt to the environment in which they operate. Nowadays, AUVs are being used for mine clearance and battle space preparation in military application; the oil and gas industry uses AUVs to make detailed maps of the seafloor before they start building subsea infrastructure.

Vehicles range in size from man portable lightweight AUVs to large diameter vehicles of over 10 meters in length. Large vehicles are more efficient in terms of endurance and sensor payload capacity; smaller vehicles benefit significantly from lower logistics (i.e., launch and recovery systems).

The majority of these roles utilize a similar design and operate in a cruise (torpedo-type) mode. They collect data while following a preplanned route at speeds between 1 and 4 knots. Most AUVs follow the traditional torpedo shape, as this is considered the best compromise between size, usable volume, hydrodynamic efficiency and ease of handling. There are some vehicles that make use of a modular design, enabling components to be changed easily by the operators.

Primarily oceanographic tools, AUVs carry sensors to navigate autonomously and map features of the ocean. Sensors on board the AUV sample the ocean while the AUV is travelling through it, providing the ability to make both spatial and time series measurements. Multiple vehicle surveys, insuring proper temporal and spatial sampling, provide a means of investigating the space-time coherence of the ocean. Typical sensors include compasses, depth sensors, sonar, magnetometers, thermistors and conductivity probes.

The navigation and positioning accuracy required in an AUV is usually determined by the mission requirements. Positional accuracy is the error which the AUV makes in determining its geographic position and navigational accuracy is the precision with which the AUV can guide itself from one geographical point to another.

The navigation system of early AUVs was based on dead reckoning. AUVs could also navigate using acoustic transponder navigation systems, providing greater accuracy but at a significant logistics cost. Furthermore, inertial navigation systems were available, but with prohibitive costs for non-military users. With the progress of inertial platform technology, the cost of inertial navigation systems has significantly

diminished, making them accessible for lower cost AUVs. The use of an inertial measurement system in an underwater environment allows for determination of roll, pitch and yaw of the AUV, and for their stabilization to improve the operational behaviour of the vehicle.

Later AUVs have taken advantage of Global Positioning Systems (GPS), which provide position estimation of the vehicle in real time with respect to world; when the vehicle surfaces, an accurate position can be obtained and the update of on-board inertial systems is performed. Thus, sensor data collected by an AUV are automatically geospatially and temporally referenced with, normally, improved accuracy as well as precision.

There is also an increasing interest in navigation relative to the environment in which the vehicle operates by using acoustic methods. They are based on measurements of the time flying of sound generated by the AUV; LongBaseLine (LBL) acoustic positioning systems use a net of seafloor-mounted baseline transponders, generally deployed around the perimeter of a work site, as reference points for navigation. When a support ship is available, UltraShortBaseLine (USBL) or ShortBaseLine (SBL) positioning is used to calculate the position of the subsea vehicle relative to the known GPS position of the surface craft. This environment referenced navigation also uses pressure sensors to measure the vertical position.

Inertial navigation systems can be used in conjunction with a Doppler Velocity Log. The inertial navigation system is essentially a hyper-accurate compass, and by measuring the Doppler shift of the acoustic waves bouncing off the seafloor the AUV can determine its speed.

The problem of accurate robot localization consists of answering the question of where the robot is in an absolute frame of reference, or from a robot's point of view. It is worth noting that the robot's position is relative to some landmark, usually represented by the destination. Self-localization is a crucial issue, since the robot has to find out its location relative to a map or landmarks at known positions, to be able to operate and act properly. Accurate robot localization has been stated as the most fundamental problem to be solved in order to provide AUVs with truly autonomous capabilities [9].

Besides the acoustic approach, there is the visual approach as an emerging alternative. The visual-based methods can be classified into two categories: methods which explicitly recover 3D information using

stereo and others that use a single camera. The robot carries its camera to determine its position by calculating the camera's point of view through looking at land marks, whose positions are known in a world-centred coordinate system (WCCS). It is possible to recover the camera point of view with three feature points and a single image from one camera; after calculating the feature points' 3D coordinates in a camera-centred coordinate system (CCCS), the Euclidean transformation between the WCCS and the CCCS is obtained. This approach proves to be reliable and efficient underwater [10].

AUVs can rely on a number of propulsion techniques, but propeller-based thrusters or Kort nozzles are the most common by far. These thrusters are usually powered by electric motors and sometimes rely on a lip seal in order to protect the motor internals from corrosion. One consideration which impacts this process of waterproofing is the decision to use brushed motors or brushless motors. This same consideration also impacts reliability, efficiency and cost.

Underwater gliders do not directly propel themselves. By changing their buoyancy and trim, they repeatedly sink and ascend; airfoil "wings" then convert this up-and-down motion to forward motion. Because of their low speed and low-power electronics, the energy required to cycle trim states is far less than for regular AUVs, and gliders have endurances of months and transoceanic ranges.

The availability of energy on an AUV to run it for longer periods of time autonomously is another issue which is under investigation, in order to cope with the demanding requirements of an AUV for different long-period applications. The endurance of AUVs has increased from a few hours to tens of hours. Some systems now contemplate missions of days and a very few, even years. This extended endurance, however, is at the expense of sensing capability, as well as very limited transit speeds. Most AUVs are powered by rechargeable batteries and are implemented with some form of battery management system.

In the majority of early AUV systems, lead acid batteries were the workhorse for energy systems. Some AUV designs included silver zinc batteries, and some applications also utilized lithium batteries. In 1987 the use of an aluminium/oxygen semi cell was also proposed.

Solar Energy Systems, currently used to power AUVs, require a detailed design of onboard energy management, both during the acquisition phase and the utilization phase of operations. Needless to say,

solar energy is an inexhaustible source, but requires an AUV to surface while recharging [11].

The communication aspect of an AUV can be considered in two different ways: one is the communication between the vehicle and the control room, and the other can be between multiple vehicles performing the same mission in collaboration with each other.

In an underwater environment acoustic communication is probably the most reliable system available. Besides that, some development programs also investigated and evaluated other technologies such as short-range laser communication and larger range noise free communication using RF current field density techniques. However, the recent developments in acoustic communications have made it possible for relatively low error rate communications over kilometers at the bit rate of a few kbps [12].

Another issue of communication is the capability to connect multiple vehicles and/or bottom-mounted instrument platforms through a subsea network-based communication infrastructure. Efforts are spent to implement this network in order to have effective communications among and between multiple AUVs.

1.2 Biomorfic fin propulsion

Systematic observations of swimming fish have revealed a fascinating paradigm of locomotion, different from conventional propulsion used in artificial vehicles. Strong scientific interest in fish-like propulsion for naval applications within the scientific community is indicated by the amount of experimental and theoretical studies in this field. Biomimecity directly emulates the form and function of species existing in nature, in order to study and use the working natural principles to design environmentally friendly artificial devices. Mechanical bio-inspired engineering takes advantage of biologic organism observation to develop solutions with technological functionality as opposed to conventional engineering solutions. Aquatic species outperform conventional aquatic and underwater artificial devices in efficiency, manoeuvrability and control performances. In addition, animals little noise, easily blending into the environment.

Due to the large amplitude of the oscillatory motion, in comparison with dimensions of the fin and the oscillation frequency, the entire body motion can achieve swimming performance with great agility. The fin

and body motion are assisted by a flow control system which avoids the occurrence of forces opposed to the motion coming from the flow separation [13–17].

As proposed by Von Karman, the lift due to the non-stationary motion of an airfoil occurs due to the contribution of three factors: a quasi-steady lift, an inertial lift component and a component coming from the wake. The von Karman vortex sheet, with the repeating pattern of swirling vortices, is caused by the unsteady separation of flow of a fluid around blunt bodies. A flow pattern, coming from vortex forms around a body in a flow, changes the pressure distribution, and an alternate shedding of vortices can create periodic lateral forces on the body, as shown in Figures 1-1 and 1-2. The thrust was evaluated by calculating the suction which is realized for a thin airfoil. This evaluation is based on the work of von Karman and Burgers. Later, Garrick showed, on the basis of discussion of von Karman and Burgers, that the oscillating motion of a surface in the flow will lead to a positive driving force [18, 19].

In an analysis of fish propulsion mechanics [20], the propulsion is classified in two types: anguilliform (eel shaped) and carangiform.

Curves are traced to represent the thrust coefficient C_T, defined as the thrust per unit of projected fin area and per unit of fluid kinetic energy, vs. the reduced frequency Ω and the feathering parameter ϑ:

$$C_T = \frac{T}{0.5 \cdot S \cdot \rho \cdot w^2}; \Omega = \frac{\omega \cdot c}{U}; \theta = \frac{U \cdot \alpha}{\omega \cdot h}$$

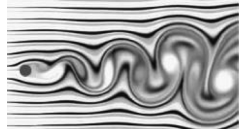


Figure 1-1 A qualitative representation of a vortex sheet downstream an object.

Figure 1-2 A qualitative representation of reverse Benard-von Karman vortex street produced by a flapping foil.

where T represents the thrust, S the fin area projected, w the water velocity relative to the fin, ω the oscillation frequency of the tail movement, c the chord length of the hydrofoil and U the fish forward speed. The parameter θ is the ratio between the amplitude of the tail angular oscillation a and the incidence angle that can be expressed as $(\omega \times h)/U$, h being the transverse displacement of the hydrofoil midpoint (Figure 1-3).

Many swimming aquatic animals generate propulsive forces by wake vortices. Many research studies in marine propulsion are focused on better understanding this form of propulsion, in order to enhance man-made propulsive mechanisms.

Classical Gray's studies, milestones of this research, underline the advantages of oscillating tail propulsion with respect to traditional propulsion. This kind of propulsion promises good efficiency, high manoeuvrability (propeller and rudder are represented by the same element), and easy motion control by imposing tail motion frequency and amplitude [21–23]. In fact, this kind of propulsion could be used for an underwater robot: it is more nimble, and less polluting than the traditional screw propeller for delicate ecosystems like the sea depth, channels subjected to wave erosion, marshes, and marine parks [24–26].

1.3 Biomorfic fin actuation

The choice of actuation strategy for moving the oscillating tail is fundamental for a successful system. Some solutions are characterized by controlled electrical motors having alternate motion on the shaft directly linked to the oscillating tail. In other prototypes, articulated mechanisms, interposed between the motor and the oscillating tail, allow for converting the continuous rotating motor shaft motion in the tail alternate oscillating movement [27–29]. Although well tested, these solutions may be heavy and complex. Furthermore, the tail motion is not natural and fish-like, because of the presence of rigid bodies assembled and articulated to simulate the natural structure. An interesting alternative for the tail actuation system consists in flexible actuators, with a deformable continuous structure moved by the pressure of an actuating

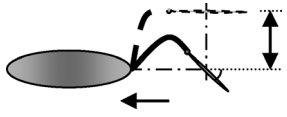


Figure 1-3 Tail geometry and parameters.

fluid, which acts in chambers with proper geometry and preferential direction of rigidity [30–34].

Research studies about these actuators for different applications are reported in literature [35–44]. For the tail motion of underwater robots, a special flexible actuator was realised and described [45–52].

1.4 Biomorfic integrated fin-actuator systems: a case study

A special flexible actuator was developed to move the oscillating tail of the fish-like robot. This actuator is a rubber-made device and has a cylindrical shape with two chambers divided by a longitudinal wall. The extreme heads permit linking to the robot body and to the tail, sealing of the chambers and the supply. Around the cylindrical rubber body, numerous rigid rings prevent radial deformations, allowing the axial one. The actuator end effector moves in a plane.

The flexible actuator is represented in Figure 1-4. The lower photograph (a) shows the assembled actuator; the different parts are shown in the upper photograph (b). One can distinguish the cylindrical elastomeric body, supplied by the two ports present on one head, the conical internal caps for sealing, and the rigid support allowing the link to the tail. The rigid rings prevent the radial deformation.

The elastomeric body section is shown in Figure 1-5. Two chambers are divided by a vertical longitudinal wall: by unbalancing the pressure values in the chambers the actuator bends in a plane perpendicular to the longitudinal wall.

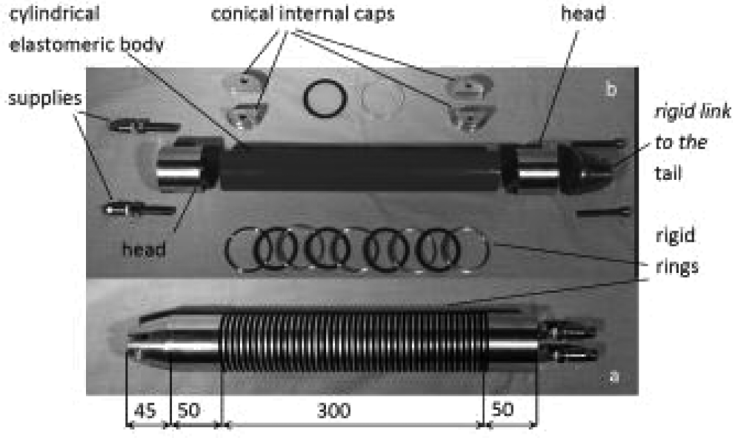


Figure 1-4 The flexible actuator with principal parts and dimensions. Source: [45] (with permission).

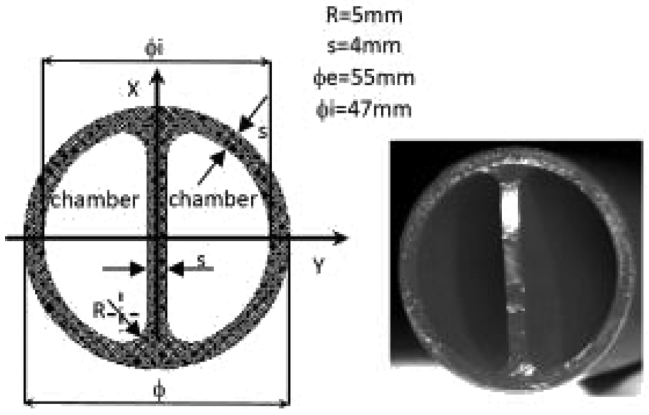


Figure 1-5 The flexible actuator section. Source: [45] (with permission).

The rubber is an elastomeric material with a 70 shore A hardness, allowing high deformations, keeping an elastic behaviour and minimising the residual deformation due to load velocity.

The mechanical stress-strain characteristic was experimentally determined [52–54]. The experimental stress-strain curve is shown in Figure 1-6; the strain varies from 20% in compression to 35% in traction. The mechanical characteristic appears strongly non-linear. The Young's modulus in the origin is about 5MPa.

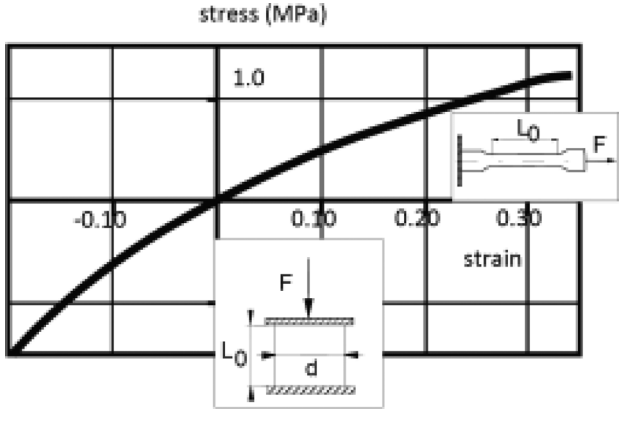


Figure 1-6 The elastomeric material stress-strain curve. Source: [45] (with permission).

Knowledge of the mechanical characteristics of the material allows for a representation of the stress field and strain distribution of the actuator section by a FEM model, when a pressure drop between the chambers loads the actuator. The middle actuator section, represented in Figure 1-5 with its main dimensions, has two geometrical symmetry planes, but only the horizontal one is a symmetry plane, when a pressure difference drop is generated between the two chambers.

Therefore, the FEM model analyses only a half section, imposing symmetry restraints; these restraints force the points on the symmetry plane to remain on it. The model takes into account the material non-linearity by a five constants Mooney-Rievlin model, yielding about 1200 plane hyper-elastic non-linear elements. The mesh is represented in Figure 1-7. Contact elements, between the cylindrical rubber external surface and the external rings preventing radial deformation, allow for representation of the radial external constraint. The load is given by internal pressures in the chambers. In particular, the model highlights the critical zone at the junction between the internal longitudinal wall and the cylindrical surface represented in Figure 1-7. In this zone, for a given actuator working condition, the stress field shows a dangerous stress concentration, strongly influenced by the radius value *R*. The optimal value must comply with different requirements: an acceptable maximum stress value in any working conditions, a limited increment of the moment of inertia, an adequate workspace shape and extension coming from a suitable section of the chamber available for the fluid.

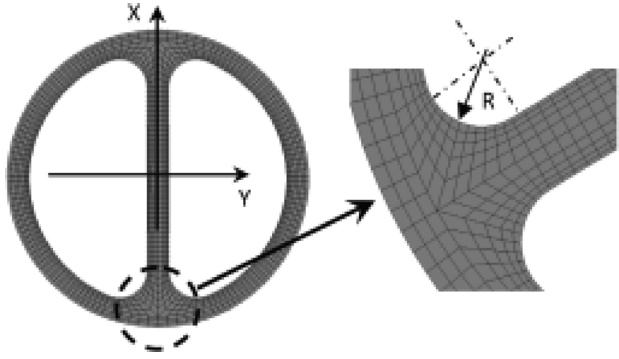


Figure 1-7 The mesh and a zoom of the fillet zone at the wall junction in the section of the flexible actuator. Source: [45] (with permission).

The actuator section, deformed by a load due to a pressure of 6 bar relative acting in chamber 1, while chamber 2 is connected to the exhaust, is shown in Figure 1-8. The longitudinal wall is strongly deformed and the junction with the cylinder corresponds to the maximum stress zone.

In the same figure the maximum von Mises stress trend is shown vs. pressure drop between the chambers, for different fillet radius R. The pressure level, in the 0–7 bar relative range, is not as influent as the pressure drop ΔP between the chambers and the link radius R.

The knowledge of the actuator working space is an important step to evaluate the suitability of the fin actuator for aquatic propulsion. The actuator workspace is traced by an experimental set up specially designed. The test rig allows the actuator clamping, vertically suspended, and the measurement of the end effector position in a 3D space, by a trigonometric indirect measure. The experimental apparatus, represented in Figure 1-9, provides the length variation of the edges of a pyramid having its basis on a horizontal plane and the upper peak linked to the actuator and the effector. The end effector position, measured at given values of the fluid pressures in the chambers, allows the workspace definition, relative to the chambers supply pressure.

The upper window in Figure 1-9 also represents the actuator when a pressure drop occurs in the chambers.

The actuator workspace is a plane surface perpendicular to the longitudinal internal wall. The workspace shape is shown in Figure 1-10, where three curves are represented corresponding to three different

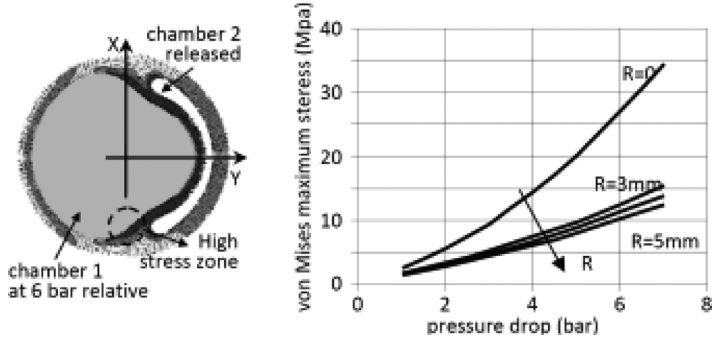


Figure 1-8 The actuator middle deformed section under pressure load and the von Mises maximum stress vs. fillet radius R. Source: [45] (with permission).

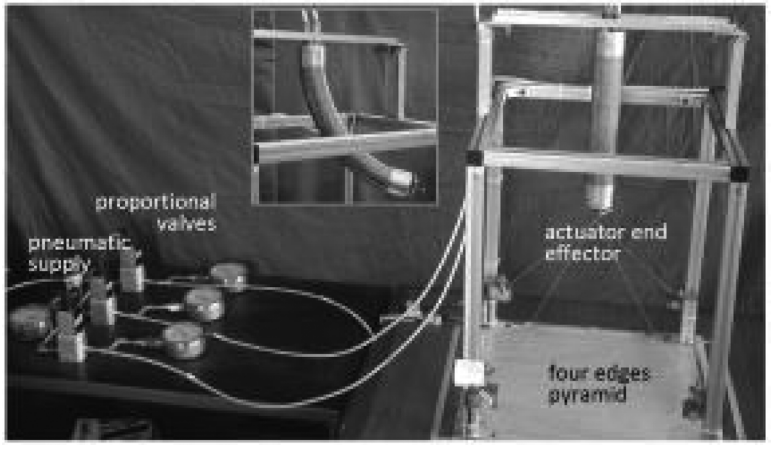


Figure 1-9 The test rig for the actuator workspace definition. Source: [45] (with permission).

maximum operating pressures. The continuous line has been drawn with a maximum pressure equal to 5 bar relative.

The end effector moves from point A, in which both chambers are discharged, along the path A-B by increasing the pressure only in chamber 1. Point B corresponds to a pressure of 5 bar in chamber 1 and zero

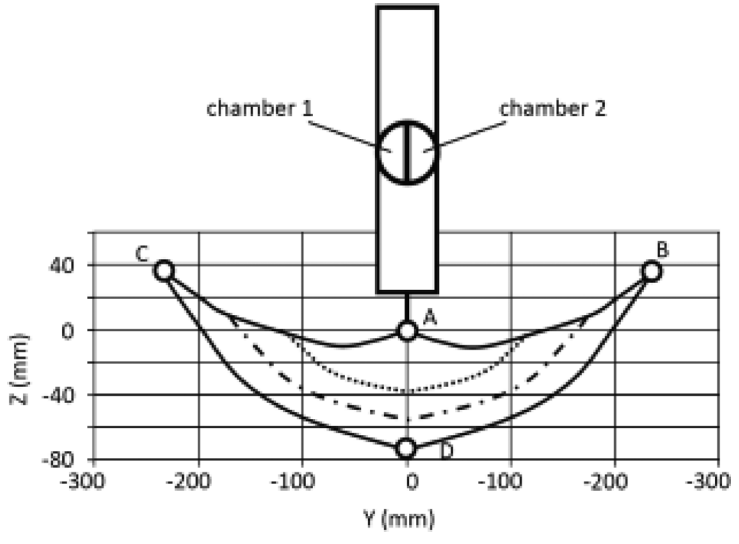


Figure 1-10 The actuator work space. Source: [45] (with permission).

in the other one. Increasing the pressure in chamber 2, the end effector describes the curve B-C, where in C the fluid pressure is 5 bar in both chambers. By discharging the chamber 1, the curve C-D is described. Finally, the end effector reaches again point A, when both chambers are discharged. All points inside the edge can be reached by the actuator end effector, combining the pressures in the chambers within the range 0–5 bar.

The flexible actuator was linked to a tail, having a geometry referred to as a carangiform real fish (Figure 1-11). The flexible actuator and the fin were assembled in a tapered body, allowing the link to a strain gauges instrumented beam for measuring the action coming from the body, dragged by the fin and moved by the flexible actuator.

The device was inserted in a hydraulic channel to measure forces acting on the body in different working conditions. Figure 1-12 shows the robot in the channel during tests and the frame of reference.

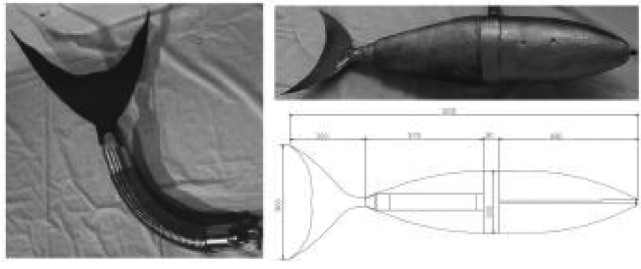


Figure 1-11 The actuator linked to the fin and assembled in a tapered body (dimension in mm). Source: [45] (with permission).

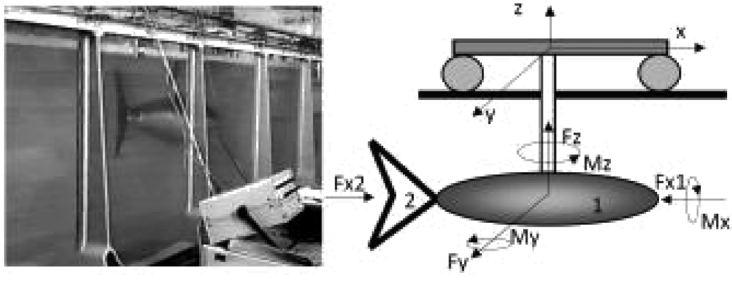


Figure 1-12 The robot in the channel during tests and the frame of reference [45].

Figure 1-13 shows the channel scheme and the force measured by the instrumented beam during the tests. Inside the channel, which is 16 m long and 1 m wide, it is possible to generate a water flow rate up to 500 l/s. On the top, a cart can be moved on a rail at a maximum speed of 2 m/s. The cart is dragged by a wire driven by a controlled electrical motor. The cart carries the measuring system with the instrumented beam and the PLC aimed at controlling the pneumatic valves to supply the flexible actuator.

In Figure 1-13 test results are reported for an oscillating tail frequency of 0.5 Hz and a pneumatic supply pressure of 6 bar relative. These test conditions perform an arc amplitude of the tail oscillating motion of 0.5 m, referred to the thrust centre of the tail.

Three curves are represented in Figure 1-14:

- the net force, obtained as a mean in time of the measured force values, dragging the fish at given velocity and operating the tail with the flexible actuator;
- the resistance, measured directly by dragging the fish, at different velocities, with a non-operating tail;
- the thrust of the tail, computed as the sum of the previously described force components.

One can note that the net force, which is directly measured, is positive up to a fish velocity of 0.6 m/s. This should indicate the forward velocity of such a fish robot propelled by this tail at these values of frequency and amplitude tail motion.

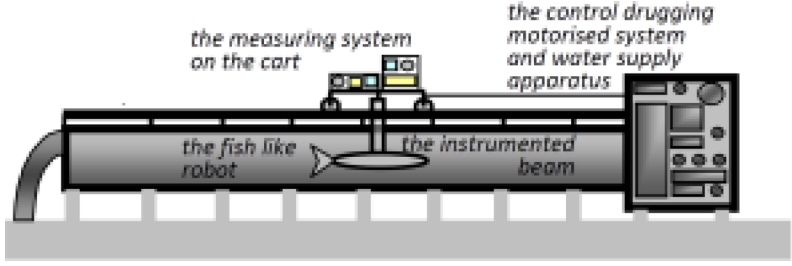


Figure 1-13 The channel scheme and generated forces in channel axis direction for oscillating tail frequency of 0.5 Hz and motion tail amplitude of 0.5 m. Source: [45] (with permission).

The resistance was measured without tail actuation. That can be significant as the resistance in working conditions, i.e., actuating the tail, can be meaningfully lower, thus leading one to overestimate tail's thrust [55].

The net force was measured directly, making it independent from this possible overestimation. The obtained results encouraged further efforts to develop new prototypes of oscillating-tail propulsion systems with increased performance. To this aim it seemed convenient to follow a biomimetic, real fish-inspired approach. The most promising improvements concern the caudal fin and the flexible actuator.

An in-depth study has been carried out in order to realise a fin with high thrust performance. For this purpose, it was conducted as a procedure, starting from the analysis of a natural fin belonging to a carangiform fish, characterised by very high swimming velocities. The authors emphasized that no animals were killed for this study, as the used caudal fin was a refuse of food supply.

The fin is shown in Figure 1-14. The high ratio between the span and the chord length suggests that the fin is highly performing. By observing the fin, one can note a junction zone, having a nearly cylindrical section, which allows the link and the mutual rotation between

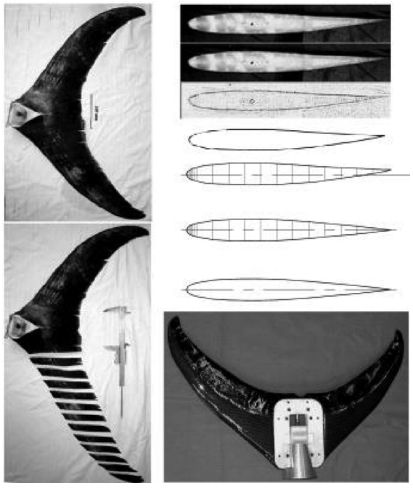


Figure 1-14 The natural and the artificial fin. Source: [47] (with permission).

the fin and the tail body. The fin zone near to the leading edge is quite more rigid than the trailing edge, where flexible strips are located.

In order to realise a device as realistic and as efficient as possible, the tail has been dissected every 20 mm along the span 620 mm long. A photograph was taken and digitised for each section, and the hydrodynamic profile was driven as reference for the design. The fin skeleton was covered by a layer of carbon fibres. In order to imitate the flexible trailing edge, a thin elastomeric foil was linked to the skeleton and covered with coated silicon. The artificial tail was provided by an articulated peduncle with controlled stiffness.

A new actuator, fundamentally similar to the first one, was designed and realised: it has an elastomeric body, divided internally in two longitudinal chambers, and it is equipped with external rings to prevent radial deformations and with two heads linking the actuator to the fin and the fish body, which supply the actuator.

The fundamental difference is the shape, which is conical to emulate the fish body. In this way it is no longer necessary to cover the actuator with a passive material to achieve a biomorphic shape.

This actuator, represented in Figure 1-15, has the same length as the cylindrical actuator (400 mm), the same diameter in the lower section (55 mm) and a diameter of 180 mm in the upper section.

The complete robot scheme is shown in Figure 1-16. The front body part is capable of housing the navigation system and energy power devices, and the rear part is represented by the actuator moving the fin. Figure 1-16 also represents the fish robot prototype suspended to the instrumented beam for the tests in hydraulic channel.

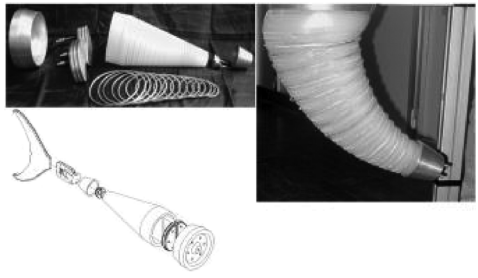


Figure 1-15 The conical actuator parts, the actuator bending and the assembled scheme of the actuation device. Source: [47] (with permission).

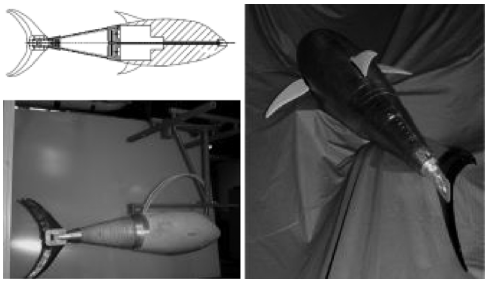


Figure 1-16 The robot fish section, the instrumented suspension and the complete robot. Source: [47] (with permission).

1.5 Conclusion

This chapter discussed, in general, the topic of underwater robots, and illustrated some particularly innovative realizations.

1.6 References

[1] Bevan, J., 1999, Diving bells through the centuries, *Journal of South Pacific Underwater Medicine Society*, **29** (1), pp. 43–50.

[2] Bachrach, A.J., 1998, History of the Diving Bell, *Historical Diving Times*, Issue 21.

[3] Thorndike, L., *A History of Magic and Experimental Science*, Vol. II, Columbia University Press, New York, 1923, ISBN 0-231-08795-0.

[4] Flynn, M.R., Bush, J.W.M., 2008, Underwater breathing: The mechanics of plastron respiration, *Journal of Fluid Mechancs*, **608**, pp. 275–296.

[5] Piper, R., 2007, *Extraordinary Animals: An Encyclopedia of Curious and Unusual Animals*, Greenwood Publishing Group, ISBN 978-0-313-33922-6.

[6] Christ, R.D., Wernli, R.L., Sr., 2007, *The ROV Manual: A User Guide for Observation-Class Remotely Operated Vehicles*, Butterworth Heinemann, Oxford (UK) and Waltham, Massachusetts (United States). ISBN 10: 0750681489/ISBN 13: 9780750681483

[7] Von Alt, C., 2000, News from the Front – Why some UUVs are in Demand, *Proceedings: UUVS 2000 Conference*, Surrey, UK, Sept. 2000.

[8] Delgado, J.P., Cussler C., 2011, *Silent Killers: Submarines and Underwater Warfare*, Osprey Publishing, Oxford, United Kingdom.

[9] Borenstein, J., Everett, H.R., Feng, L., Wehe, D., 1997, Mobile Robot Positioning & Sensors and Techniques, *Journal of Robotic Systems*, Special Issue on Mobile Robots. Vol. **14** (4), pp. 231–249.

[10] Zhang, P., Milios, E.E., Gu, J., 2004, Underwater Robot Localization using Artificial Visual Landmp, *Proceedings IEEE International Conference on Robotics and Biomimetics*, Shenyang, China, pp. 705–710.

[11] Patch, D.A., 2000, A solar energy system for long-term deployment of AUVs, *International Unmanned Undersea Vehicle Symposium*, April 2000, Newport, Rhode Island, USA.

[12] Marques, E.R.B., Pinto, J., Kragelund, S., Dias, P.S., Madureira, L., Sousa A., Correia, M., Ferreira, H., Goncalves, R., Martins, R., Horner, D.P., Healey, A.J., Goncalves, G.M., Sousa, J.B., AUV Control and Communication using Underwater Acoustic Networks, *Proceedings of the Oceans Europe '07 Conference*, IEEE, Aberdeen, Scotland, June 2007.

[13] Borazjani, I., 2013, The functional role of caudal and anal/dorsal fins during the C-start of a bluegill sunfish, *The Journal of Experimental Biology*, **216**, pp. 1658–1669.

[14] Wolfgang, M.J., Anderson, J.M., Grosenbaugh, M.A., Yue, D.K., Triantafyllou, M.S., 1999, Near-body flow dynamics in swimming fish, *Journal of Experimental Biology*, **202**, pp. 2303–2327.

[15] Lauder, G.V., Anderson, E.J., Tangorra, J., Madden, P.G.A., 2007, Fish biorobotics: Kinematics and hydrodynamics of self-propulsion, *Journal of Experimental Biology*, **210**, pp. 2767–2780.

[16] Lauder, G.V., 2000, Function of the caudal fin during locomotion in fishes: Kinematics, flow visualization and evolutionary patterns, *American Zoologist*, **40** (1), pp. 101–122.

[17] Flammang, B.E., Lauder G.V., Pectoral fins aid in navigation of a complex environment by bluegill sunfish under sensory deprivation conditions, *The Journal of Experimental Biology*, **216**, pp. 3084–3089.

[18] von Karman, T., Burgers, J.M., *General Aerodynamic Theory-Perfect Fluids*, Vol. II, Julius Springer, Berlin, 1935.

[19] Garrick, I.E., Propulsion of a Flaping and Oscillating Airfoil, NACA TR No. 567, May 1936.

[20] Harper, K.A., Berkemeier, M.D., Sheril, G., 1998, Modeling the Dynamics of Spring-Driven Oscillating Foil Propulsion, *IEEE Journal of Oceanic Engineering*, **23** (3), pp. 285–293.

[21] Gray, J., 1933, Studies in animal locomotion I. The movement of fish with special reference to the eel, *Journal of experimental Biology*, **10**, pp. 88–104.

[22] Gray, J., 1933, Studies in animal locomotion II. The relationship between the waves of muscular contraction and the propulsive mechanism of the eel, *Journal of experimental Biology*, **10**, pp. 386–390.

[23] Gray, J., 1933, Studies in animal locomotion III. The propulsive mechanism of the whiting Gadus Merlangus. *Journal of Experimnetal Biology*, **10**, pp. 391–400.

[24] Garrick, I.E., 1936, Propulsion of A Flapping And Oscillating Airfoil. *NACA Tech. Report 567*.

[25] Lighthill, M.J., 1969, Hydromechanics of aquatic animal propulsion, *Journal of Fluid Mechanics*, **44**, pp. 265–301.

[26] Barret, D.S., Triantafillou, M.S., Yue, D.K.P., Groosenbaugh, M.A., Wolfgang, M.J., 1999, Drag Reduction in Fish Like Locomotion, *Journal of Fluid Mechanics*, **392**, pp. 183–212.

[27] Triantafyllou, G.S., Triantafyllou, M.S., Grosenbaugh, M.A., 1993, Optimal Thrust Development in Oscillating Foils with Application to Fish Propulsion, *Journal of Fluids and Structures*, **7** (2), pp. 205–224.

[28] Rediniotis, O.K., Lagoudas, D.C., 1997, An SMA Actuator for Aquatic Biomimetics, *SPIE 1997 Smart Structures and Materials Conference*, San Diego, CA, USA.

[29] Rediniotis, O.K., Lagoudas, D.C., Garner, L.J. Wilson, N., 1998, Experiments and Analysis of an active Hydrofoil with SMA actuators, AIAA Paper No. 98-0102, *36th Aerospace Sciences Meeting*, Reno, Nevada, USA.

[30] Inoue, K., 1991, Bridgstone Corporation Products Use (ACFAS), *System Driven by Rubberactuators Industrial Robot*, **1**, pp. 35–36.

[31] Fukuda, T., Guo, S., Wang, I., Kosuge, K., Arai, F., 1992, Artificial Muscular Type of Actuators, Structure and Characteristics of Serial-Parallel Type of Pneumatic Actuator. *IFToMM-jc Int. Symp. on Theory of Machines and Mechanism*, Nagoja, Japan, pp. 862–867.

[32] Bruce, J., Davies, C., 1991, Elephant Trunks, an Unforgettable Alternative to Rigid Mechanism, *Industrial Robot*, **18** (4), pp. 29–31.

[33] Taniguchi, H., Tanaka, H., 1996, Applying flexible-microactuators to multi-fingered robot hand, *Third JHPS Int. Symp. Fluid Power*, Yokohama, Japan, pp. 427–430.

[34] Nobumoto, Y., Tanaka, H., 1996, Enhanced FMA Legs for quadruped Walking. *Third JHPS International Symposium on Fluid Power*, Yokohama, Japan, pp. 115–120, 1996.

[35] Ferraresi, C., Manuello Bertetto, A., Franco, W., 1998, Modelisation and Characterisation of a Pneumatic Muscle Actuator for non Conventional Robotics, *7th Int. Workshop on Robotics in Alpe Adria Danube Region RAAD 98*, Smolenice Castle, Slovakia, pp. 279–284.

[36] Ferraresi, C., Manuello Bertetto, A., Mazza, L., 1997, Design and Realisation of a Flexible Pneumatic Actuator for Robotics, *5th Scandinavian Int. Conf. on Fluid Power SICFP 97*, Linkoping University, Sweden, Vol. III, pp. 29–43.

[37] Ferraresi, C., Franco, W., Manuello Bertetto, A., 2001, Flexible Pneumatic Actuators: A Comparison between the McKibben and the Straight Fibres Muscle. *J. of Robotics and Mechatronics*, **13** (1), pp. 56–63.

[38] Manuello Bertetto, A., Ruggiu, M., 2001, In-Pipe inch-Worm pneumatic flexible Robot, *IEEE ASME Int. Conf. On Advanced Intelligent Mechatronics*, Como, Italy, p. 1226–1231.

[39] Akimasa Fukuhara, Tetsuya Akagi, Shujiro Dohta, 2010, Development and Application of Flexible Pneumatic Cylinder with Linear Encoder, SICE Annual Conference 2010, Taipei, Taiwan, pp. 768–772.

[40] Tetsuya Akagi, Shujiro Dohta, Hisashi Matsushita, Akimasa Fukuhara, 2012, Development of Flexible Pneumatic Cylinder with Built-in Flexible Linear Encoder and Flexible Bending Sensor, *Journal of System Design and Dynamics*, **6** (4), pp. 359–372.

[41] Ahmad `Athif M. Faudzi, M. Rusydi M. Razif, Ili Najaa Aimi M. Nordin, Suzumori K., Wakimoto S., Hirooka, D., 2012, Development of Bending Soft Actuator with Different Braided Angles, *The 2012 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Kaohsiung, Taiwan, pp. 1093–1098.

[42] Manuello Bertetto, A., Meili S., Concu A., Crisafulli, A., 2012, An Inflatable Pneumatic System For Blood Pressure Recovery, *Mechanics Based Design of Structures and Machines, Taylor and Francis*, **40** (4), pp. 506–518.

[43] Manuello Bertetto, A., Ruggiu, M., 2004, A Novel Fluidic Bellow Manipulator, *Journal of Robotics and Mechatronics*, **16** (6), pp. 604–612.

[44] Manuello Bertetto, A., Ruggiu M., 2001, Pneumatic Robot for Pipe Inspection, *International Journal of Mechanics and Control*, **2** (2), pp. 31–38.

[45] Cataudella, C., Ferraresi, C., Manuello Bertetto, A., 2001, Flexible actuator for oscillating tail marine robot, *International Journal of Mechanics and Control*, **2** (2), pp. 13–21

[46] Manuello Bertetto, A., Picasso, B., Ruggiu, M., 2001, Fish and ships: Can fish inspired propulsion outperform traditional propeller based systems?, *Proceedings of the Conference Marine Technology IV*, Szczecin, Poland, pp. 279–287.

[47] Ferraresi, C., Manuello Bertetto, A., Costamagna, A., Golle, D., 2003, Integrated Fin-Actuator Systems For a Marine Robot,

Proceedings of RAAD 2003, 12th International Workshop on Robotics in Alpe-Adria-Danube Region, Cassino Italy.

[48] Manuello Bertetto, A., Ruggiu, M., 2006, Tail Actuator Propulsion Device for Aquatic Robot, *Journal of Robotics and Mechatronics*, **18** (1), pp. 89–96.

[49] Ando, Y., Kato, N., Suzuki, H., Ariyoshi, T., Suzumori, K., Kanda, T., Endo, S., 2006, Elastic Pectoral Fin Actuators for Biomimetic Underwater Vehicles, *Proceedings of the 16th International Offshore and Polar Engineering Conference*, San Francisco, California, USA, pp. 260–267.

[50] Koichi Suzumori, Satoshi Endo, Takefumi Kanda, Naomi Kato, Hiroyoshi Suzuki, 2007, A Bending Pneumatic Rubber Actuator Realizing Soft-bodied Manta Swimming Robot, 2007 *IEEE International Conference on Robotics and Automation*, Roma, Italy, pp. 4975–4980.

[51] Naomi Kato, Yoshito Ando, Ariyoshi Tomokazu, Hiroyoshi Suzuki, Koichi Suzumori, Takefumi Kanda, Satoshi Endo, 2008, Elastic Pectoral Fin Actuators for Biomimetic Underwater Vehicles, *Bio-mechanisms of Swimming and Flying*, editors: Naomi Kato Ph.D. and Shinji Kamimura Ph.D., ISBN: 978-4-431-73379-9 (Print) 978-4-431-73380-5 (Online).

[52] Mizuho Shibata, Yuusuke Onishi, Sadao Kawamura, 2010, Experimental Evaluation of a Flexible Joint Driven by Water Pressure for Underwater Robots, *The 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan.

[53] ASTM D412-83 - Standard Test Methods for rubber properties in tension.

[54] ASTM D575-83 - Standard Test Methods for rubber properties in compression.

[55] Gray, J., 1968, *Animal Locomotion*, Weidenfeld & Nicolson, London.

2. A novel Lighter Than Air Vehicle – Flying Octopus

Zheng Li and Ruxu Du, The Chinese University of Hong Kong, China

Abstract: This chapter presents a novel Lighter-Than-Air-Vehicle (LTAV), named Flying Octopus. It is comprised of a helium balloon, four independently controlled wire-driven flapping wings, and a midsection which connects the balloon and flapping wings (as well as housing all the electronics). Each wing is a continuous membrane, and can be bent up to 100 degrees by the wires. The wing flapping is generated by the bending motion. With proper control, the Flying Octopus can fly in 3D space freely, including vertical and horizontal movement.

2.1 Introduction

People have been dreaming of flying for thousands of years. Numerous attempts were made, but the first successful attempt was the lighter-than-air approach. In 1785, Jean-Pierre Blanchard crossed the English Channel in an air balloon [1]. In fact, before fixed-wing airplanes, the lighter-than-air-vehicle (LTAV) was the sole method of aerial transportation. The first commercial airship was built by the French engineer Henri Giffard in 1852 [2]. Since then, airships have been used for many decades. Compared to other aerial transportation methods, LTAVs have several advantages.

First, they are energy efficient. LTAVs utilize buoyance to suspend themselves in air and their payload could be very large. Second, they can hover in the air and their speed could be very slow. This makes the LTAV very suitable for sightseeing, aerial photography, aerial monitoring, etc. Although helicopters can also stay still in the air, they are noisy and the durance is limited by the onboard power. The LTAV also has several disadvantages: its size is large, its speed is slow and its motion is strongly affected by wind. Moreover, the usage of flammable gas is always a safety concern. With these disadvantages, the LTAV is gradually being replaced by fixed-wing airplanes. However, LTAVs still play an important role in certain applications, such as advertising, unmanned surveillance and indoor entertainment.

One representative LTAV is the airship. Conventional airships use streamlined balloons and are propelled by screw propellers. The airship built by Henri Giffard in 1852 used a soft balloon with a propulsion system and a steering system [2]. The length was 143 feet (43.6 m) and the diameter is 40 feet (12.2 m). It could fly over 17 miles (27.5 km) at a speed of 5 mph (8 km/h). Since then, many airships have been built. The airship balloon can be non-rigid, semi-rigid or rigid. A non-rigid airship is also called a blimp. Its shape is sustained by the pressure difference between the two sides of the membrane of the balloon. Non-rigid airships are simple in structure, easy and inexpensive to make, however, they are usually small. Semi-rigid airships have a rigid keel at the bottom of the balloon, which helps to sustain the shape of the balloon and to distribute the loads evenly. Many sightseeing airships are of this kind. Rigid airships usually employ a metal framework to maintain the shape. The overall weight is the largest, so is its size. The volume of a rigid airship is mostly over 50,000 m^3. Figure 2-1 shows one modern airship named Zeppelin NT [3]. The hull is 75 m long, with a volume of 8,225 m^3. The standard cruising speed is 70 km/h, and the operational altitude is between 300 m and 2,600 m. Its maximum takeoff weight is over 10,000 kg, with a payload of 1,900 kg.

In recent decades, the development of LTAVs shows two trends. One is that hull shapes are much more diversified and the other is that indoor LTAVs are using novel propulsion methods. Streamlined LTAVs

Figure 2-1 Outdoor LTAV example - Zeppelin NT airship. Source: [3] (with permission).

are the most common. This shape can reduce air drag, but the volume to surface ratio is small, which increases the overall dimension of the LTAV. Speed is not a primary concern for entertainment LTAVs, hence spherical airships are built, such as the one developed by Flying Yachts, Inc. [4]. It's shaped like a football, and the cabin is inside the ball. To improve the stability of the airship, Advanced Technologies Group Ltd. built a double hull airship [5]. Perhaps to satisfy people's curiosity, a British company, Thermo Skyship, built lenticular airships [6], which look like spaceships from science fiction. An airship with fixed wings was also proposed [6], as it is believed this would improve the speed. These LTAVs are all for outdoor applications. Indoor LTAVs are mostly for entertainment. Their shapes are even more diversified, looking like jelly, fish and even cartoon figures [7]. Figure 2-2 shows the air jelly developed by Festo Co. [8] and air swimmers developed by William Mark Co. [9].

Compared to the diversity in shape, the change in propulsion method is rather limited. For large-sized outdoor LTAVs, the screw propeller remains the sole way of actuation. For indoor applications, the weight of an LTAV is small, and the size is medium. It is easier to incorporate novel actuation methods, such as the bio-inspired flapping propulsion. Examples include the aforementioned air jelly and air swimmers. Air jelly has eight flapping wings, mimicking the tentacles of the jelly fish. They are driven by just one motor. By adjusting the flapping frequency, the flying velocity and altitude of the LTAV can be controlled. To steer the air jelly, a pendulum system is used to change the mass center of the

Figure 2-2 Indoor LTAVs: air jelly (left) Source: [8] (with permission) and air swimmers (right) Source: [9] (with permission).

LTAV. The air swimmers mimic the fish swimming. Instead of flapping the wings, they move around by flapping the tail. By adjusting the flapping amplitude, frequency and flapping pattern, the speed and direction are controlled. The pitch motion is controlled by a balance weight, which is also used to change the mass center of the LTAV. It should be pointed out that flapping is a highly efficient way to generate thrust and propulsion. Birds and fish live in different environments, but they both adopted flapping as a means of movement over the course of millions of years of evolution. Clearly, flapping has its advantages. Table 2-1 compares the screw propeller and the flapping wing or tail in driving the LTAV. From the table, it can be seen that flapping is indeed an interesting propulsion method that may be better suited for certain applications.

The authors developed a biomimetic wire-driven mechanism. It mimics not only the skeleton structure but also the muscle arrangement of fish and hence can generate highly efficient propulsion in water [10–17]. In this chapter, we introduce a novel LTAV propelled by the wire-driven mechanism [18]. It is named "Flying Octopus," as it looks like an octopus. The rest of the chapter is organized as follows: Section 2.2 presents the design, Section 2.3 gives the modeling and computer simulation results, Section 2.4 discusses the control of the Flying Octopus, Section 2.4 presents the prototyping and experiment testing results, and Section 2.5 contains conclusions.

Table 2-1 Comparison between screw propeller and flapping wing/tail.

	Screw Propeller	Flapping Wing/Tail
Motion type	Unidirectional rotation	Oscillatory motion
Efficiency	Low	High
Thrust range	$0 \sim 10^6$ N	$0 \sim 10^2$ N
Speed of vehicle	High	Low
Maneuverability	Low	High
Acoustic noise	Noisy	Quiet

2.2 The design of Flying Octopus – a bio-inspired design

The design of our Flying Octopus mimics that of a real octopus. The overall density of an LTAV is usually close to its environmental medium. In nature, some aquatic animals, like the octopus, behave like an LTAV. An octopus consists of a round body (head) and several tentacles (or arms). Similarly, our Flying Octopus is made of a body (helium balloon) and several tentacles (flapping wings for propulsion). The design is detailed below.

2.2.1 Design considerations

Our Flying Octopus is designed as an indoor LTAV. It is composed of a helium balloon, several wire-driven flapping wings and a middle section. For indoor applications, the size and the weight of the LTAV are the primary concerns; speed and agility are secondary.

Figure 2-3 shows the general considerations in designing an indoor LTAV. The first consideration is overall size. There should be enough space for the LTAV to fly. In a typical setting, the size of the LTAV should be at least 10 times smaller than the showroom. For our Flying Octopus, the height is determined by the size of the balloon and the length of the flapping wings. A spherical balloon has the maximum volume to surface ratio. Its dimension is the smallest with the same buoyance. Thus, it is the preferred choice. The flapping wings are hanging on the balloon. Their lengths are adjusted according to the balloon diameter.

The second consideration is weight. The LTAV is designed to be suspended in the air by the helium balloon. Thus, its overall weight is close to the buoyance provided by the balloon. For our Flying Octopus, the

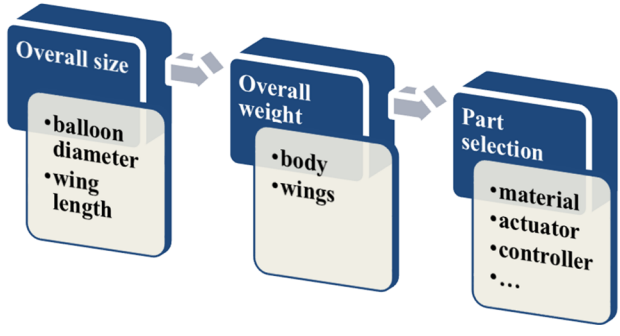


Figure 2-3 Flying Octopus design considerations.

weight is determined by the balloon, the flapping wings, the structure, and the driving system including the motors, the actuators, the power supply, the control circuits, etc. Parts should be selected according to the weight constraint. A common practice is to distribute the weight to each part, and then start to design or order commercial parts.

2.2.2 The design

The octopus lives in water, and its overall body density is close to water. It has a round body and eight tentacles, or arms. The Flying Octopus mimics the look of the octopus, but flies in the air. As mentioned in the previous section, the size and weight of the Flying Octopus are the primary design constraints. The Flying Octopus has a spherical balloon as the body, and four flapping wings as the tentacles. The wings are connected to the balloon via a middle section. Figure 2-4 shows the overall design of the Flying Octopus.

The diameter of the balloon is 1.5 m, with a volume of 1.767 m^3. When it is filled with helium, the buoyance is about 19.3 N. In other words, the overall weight of the Flying Octopus is limited to 2.02 kg. Of course one can use a larger balloon to increase loading capacity.

The Flying Octopus is required to fly in 3D space by flapping its wings without other assisting facilities. To minimize the overall weight, the number of wings should be as little as possible. To realize 3D flying, the minimum number of wings is three. However, by using four wings, motion control can be simplified. As a result, the Flying Octopus has four wings arranged orthogonally. The wings are independently controlled.

The wings are connected to the balloon via the middle section as shown in Figure 2-5. The middle section also supports other parts, such

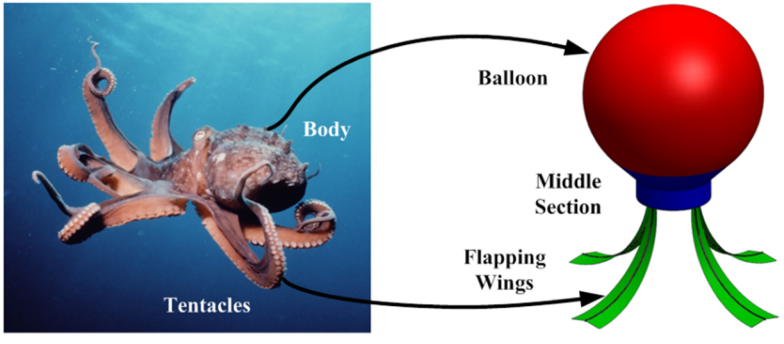


Figure 2-4 Flying Octopus overall design.

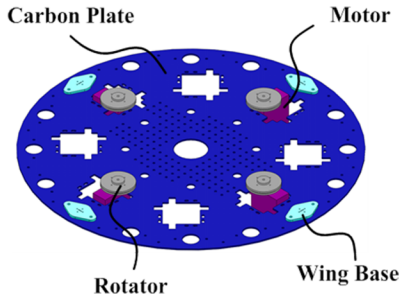


Figure 2-5 Middle section design.

as the actuators, the power supply and the control circuit. The midsection structure should be lightweight and strong enough for this purpose. Therefore, a carbon plate is used.

In the carbon plate midsection, the evenly distributed square slots are used to mount the servomotors and the circular holes are used to hold the wings. Each wing is actuated by a servo motor via the aforementioned wire-driven mechanism [10, 13]. The detailed design is shown in the subsequent section. The midsection is also used to hold the balloon, the power supply and the control circuit. The midsection is circumscribed by a fence, as shown in Figure 2-4. This not only protects the midsection but also helps to secure the balloon. The top of the fence presses against the balloon. This increases the contact area between the balloon and the midsection. As a result, swaying is reduced.

2.2.3 The wire-driven flapping wing

The flapping wings propel the Flying Octopus. They not only provide thrust but also control the motion of the Flying Octopus: up or down, left or right, forward or backward. The design considerations include: (a) large flapping motion; (b) light weight; and (c) easy to control. To meet these requirements, the single segment continuum wire-driven mechanism with tapered wire configuration is used.

1) Continuum wire-driven mechanism

A wire-driven mechanism was first proposed for flexible robots [10, 13]. It imitates the musculoskeletal system of animals, especially that of snakes and an octopus arm. It is composed of a serpentine or continuum backbone, and a number of wire pairs. Each pair of wires

controls one bending degree of freedom (DOF) of the backbone. It is underactuated, and can bend to large angles. Figure 2-6 shows the single segment continuum wire-driven mechanism. It has a continuous backbone, one pair of wires and a couple of vertebras. The wires are guided by the vertebras, with one end connected to the Vertebra-0, which is the distal end, and the other end connected to the actuator. At the resting position, the backbone is straight and the two wires lie symmetrically on both sides of the backbone. They are equally prestressed. By pulling one wire and loosening the other, the wires exert a bending movement to the continuous backbone. The backbone bends toward the pulled wire. The backbone deformation is shown in Figure 2-6(b). When there are no other external loads, the deflected backbone is close to a circular arc, or the backbone curvature is constant. By pulling the two wires alternatively, the backbone bends side to side, generating the flapping motion.

2) Flapping wing design

The designed flapping wing is shown in Figure 2-7. It has four parts: the membrane, the backbone, the vertebras and the wires. The overall length of the wing is 680 mm.

The membrane defines the profile of the wing and also serves as a secondary backbone. In this design, the membrane width shrinks from the wing base to the distal tip. The width close to the wing base is 100 mm, and the width at the tip is 40 mm. The end of the wing is a lunate flipper. The backbone connects to the Flying Octopus body and supports the wing structure. It is slim and has a rectangular cross section. The

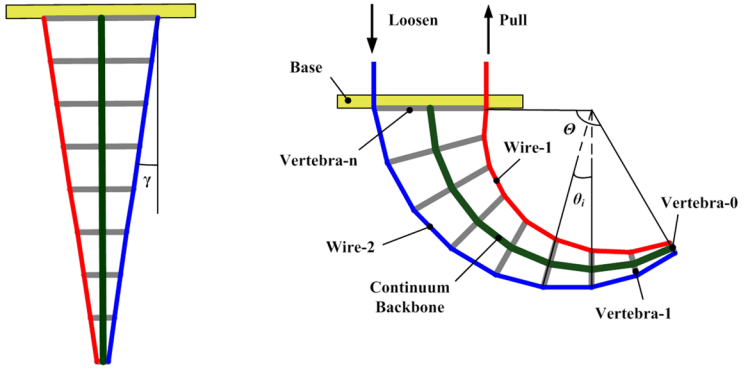


Figure 2-6 Continuum wire-driven mechanism.

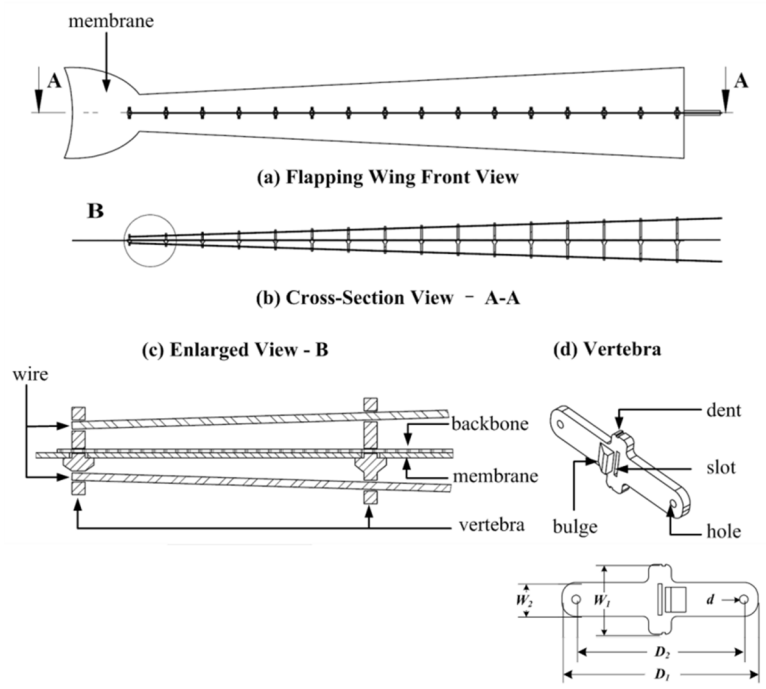


Figure 2-7 Wire-driven flapping wing design. Source: [18] (with permission).

width is 5 mm and the thickness is 0.5 mm. The bending rigidity of the thickness direction is 100 times less than the width direction. Together with the membrane, the wing can bend over 100 degrees without twisting. As shown in Figure 2-7(b), the wires are symmetrically placed on both sides of the backbone and there is a taper angle between the backbone and the wires. The vertebras are used to guide the wires. In this design, all the vertebras in the flapping wing are similar in structure. They have the same features: bulge, slot, hole, dent and rib, as shown in Figure 2-7(d). From the flipper to the wing base the rib length increases gradually. The growth rate is determined by the taper angle.

The key parameters of the vertebra are defined in Figure 2-7, and the parameters for each vertebra are shown in Table 2-2. Note that the graduate changes of D_1 and D_2 create the change of tapper angle. There are 16 vertebras in each wing. The vertebras are evenly distributed along the backbone. Their locations, or distance from the wing base to the

Table 2-2 The design parameters of the vertebra.

Num.	D_1 (mm)	D_2 (mm)	W_1 (mm)	W_2 (mm)	d (mm)	Location (mm)
1	82	77	12.5	6	1.5	10
2	78.5	73.5	12.5	6	1.5	50
3	75	70	12.5	6	1.5	90
4	71.5	66.5	12.5	6	1.5	130
5	68	63	12.5	6	1.5	170
6	64.5	59.5	12.5	6	1.5	210
7	61	56	12.5	6	1.5	250
8	57.5	52.5	12.5	6	1.5	290
9	54	49	12.5	6	1.5	330
10	50.5	45.5	12.5	6	1.5	370
11	47	42	12.5	6	1.5	410
12	43.5	38.5	12.5	6	1.5	450
13	40	35	12.5	6	1.5	490
14	36.5	31.5	12.5	6	1.5	530
15	33	28	12.5	6	1.5	570
16	29.5	24.5	12.5	6	1.5	610

vertebra, are also shown in Table 2-2. The connection of the four parts is shown in Figure 2-7(c). From the enlarged cross-section view, we can see that the vertebras are placed in the rectangular slots of the membrane and are positioned by the dent on the vertebra. There is a bulge and a slot around the center of each vertebra. The bulge presses against the membrane, leaving the vertebra slot on the other side of the membrane. The backbone inserts into the slots of the vertebras and locks the vertebras like a latch. This makes the connection of the three parts very simple and solid. After the connection, the wires are passed through the vertebra's hole and fasten to the distal vertebra.

2.3 Modeling and simulation

2.3.1 Wing kinematics

The kinematic model of a wire-driven mechanism with paralleled wires is presented in [13]. In the Flying Octopus, the wires are intersecting.

The kinematic model is derived in a similar way. For the flapping wing, the flapping angle is a concern.

Assume the backbone is inextensible. Under pure bending, the backbone curve is a circular arc. As all the (n + 1) vertebras are evenly distributed along the backbone, the angle between each vertebra is the same. We can treat two adjacent vertebras as a virtual joint, and there are n virtual joints. Assuming the rotation of each joint is θ, the wing will bend $\Theta = n \cdot \theta$.

The wire length change in the joint is illustrated in Figure 2-8. For each joint, at resting position, adjacent vertebras are parallel and their distance is h. The lengths of the two wires between the vertebras are equal, and could be represented as:

$$l_{i0} = \sqrt{h^2 + (r_i - r_{i-1})^2} \tag{1}$$

where r_i and r_{i-1} are the length of the vertebra rib. Initially, the overall wire length is $L_0 = n \cdot l_{i0}$.

When bent, the curvature of the backbone becomes:

$$R = \frac{h}{\theta} \tag{2}$$

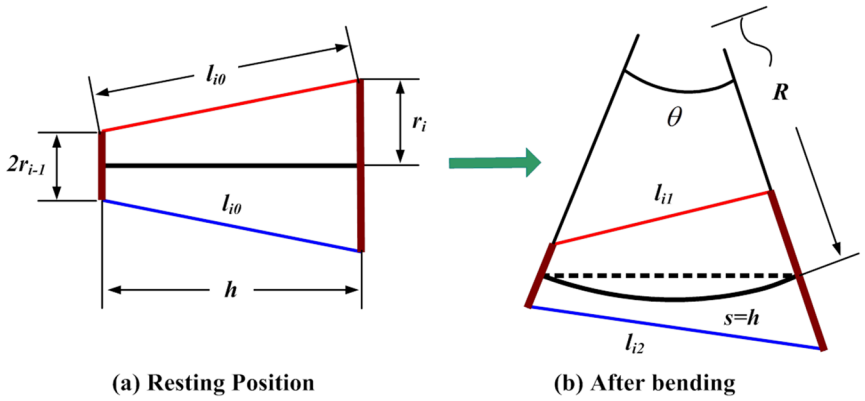


Figure 2-8 Continuum wire-driven mechanism bending illustration. Source: [18] (with permission).

The lengths of the two wires can be represented using the law of cosines:

$$\begin{cases} l_{i1} = l_{i0} - \Delta l_{i1} = \sqrt{(R - r_i)^2 + (R - r_{i-1})^2 - 2(R - r_i)(R - r_{i-1})\cos\theta} \\ l_{i2} = l_{i0} + \Delta l_{i2} = \sqrt{(R + r_i)^2 + (R + r_{i-1})^2 - 2(R + r_i)(R + r_{i-1})\cos\theta} \end{cases} \tag{3}$$

In the wire-driven mechanism, the joint rotation θ is usually small. Hence, it is reasonable to make the simplification: $\cos(\theta) \approx 1 - \theta^2/2$. The length change within each joint for the two wires can be approximated as:

$$\Delta l_{i1} \approx \Delta l_{i2} \approx 0.5 \cdot (r_{i-1} + r_i) \cdot \theta \cdot \cos(\gamma) \tag{4}$$

For the Flying Octopus, the taper angle is 2.5°. When the wing flapping angle is 150°, the length changes approximation error for the two wires is less than 1.6%.

The total length after bending is obtained by summing all the segments. The overall length changes for the two wires are therefore:

$$\begin{cases} L_1 \approx n \cdot l_{i0} - 0.5 \cdot \sum_{i=1}^{n} (r_{i-1} + r_i) \cdot \theta \cdot \cos(\gamma) \\ L_2 \approx n \cdot l_{i0} + 0.5 \cdot \sum_{i=1}^{n} (r_{i-1} + r_i) \cdot \theta \cdot \cos(\gamma) \end{cases} \tag{5}$$

From Equation (5), the wing flapping angle can also be approximated as:

$$\Theta = n \cdot \theta \approx \frac{n(L_2 - L_1)}{\sum_{i=1}^{n} (r_{i-1} + r_i) \cdot \cos(\gamma)} \tag{6}$$

The wire velocities are obtained by deriving equation (5) once with respect to time:

$$\begin{cases} V_1 \approx -0.5 \cdot \sum_{i=1}^{n} (r_{i-1} + r_i) \cdot \cos(\gamma) \cdot \dot{\theta} \\ V_2 \approx +0.5 \cdot \sum_{i=1}^{n} (r_{i-1} + r_i) \cdot \cos(\gamma) \cdot \dot{\theta} \end{cases} \tag{7}$$

2.3.2 Propulsion model

As shown in Figure 2-9, the forces acting on the Flying Octopus include gravity G, drag force F_d, buoyance F_b, thrust F_t and lateral force F_l. By delicate adjustment, gravity and buoyance balance each other, and the Flying Octopus can suspend in air. The thrust and lateral forces are generated by the flapping wings. They are used to control the moving direction and velocity of the Flying Octopus. Drag force is opposite to the moving direction.

The external forces acting on the flying octopus are shown in Table 2-3. In the table m is the overall mass of the Flying Octopus, g is gravity constant, ρ_{air} is the density of air, C_d is the drag coefficient, A is the projected area in the moving direction, v is the velocity of the Flying Octopus, V is the overall volume, C_L is the thrust coefficient, $u(q)$ is the velocity of the points on the membrane, q is the position along the

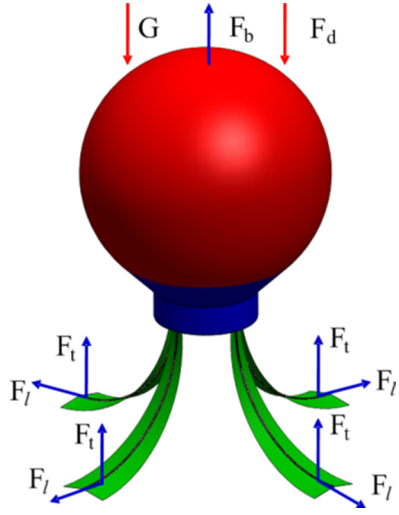


Figure 2-9 Flying octopus force analysis. Source: [18] (with permission).

Table 2-3 Forces on the Flying Octopus.

Gravity:	$G = mg$
Drag force:	$F_d = 0.5\rho_{air} C_d A v^2$
Buoyance:	$F_b = \rho_{air} V g$
Single wing thrust:	$F_t = \int_0^{\Theta} \left[0.5\rho_{air} C_L u(q)^2 \sin(\alpha) \right] dS$
Single wing lateral force:	$F_l = \int_0^{\Theta} \left[0.5\rho_{air} C_L u(q)^2 \cos(\alpha) \right] dS$

backbone starting from the wing base, α is the complement angle of $u(q)$ and v, and dS is the infinitesimal area of the membrane at angle α.

As shown in Figure 2-10, the general coordinate q is set as the arc distance from the wing base to the point, along the backbone. Based on the constant curvature assumption and wing kinematics, the velocity of the point at q is:

$$u = \dot{\Theta} \cdot \frac{L}{\Theta^2} \sqrt{\alpha^2 - 2[\alpha \sin(\alpha) + \cos(\alpha) - 1]} \tag{8}$$

where $\alpha = q \cdot \Theta / L$.

2.3.3 Propulsion simulation

A Matlab® program was developed to simulate the motion of the Flying Octopus. Figure 2-11 shows the user interface of the program as well as a simulation result. The upper left window shows the instantaneous wing motion. For better visualization, only two wings are shown in the figure. The right window shows the simulation parameter and the lower

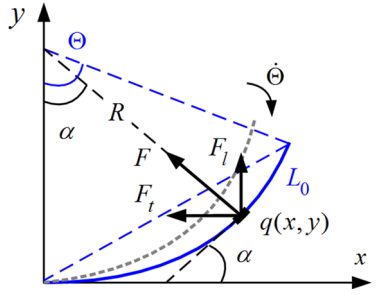


Figure 2-10 Thrust force illustration. Source: [18] (with permission).

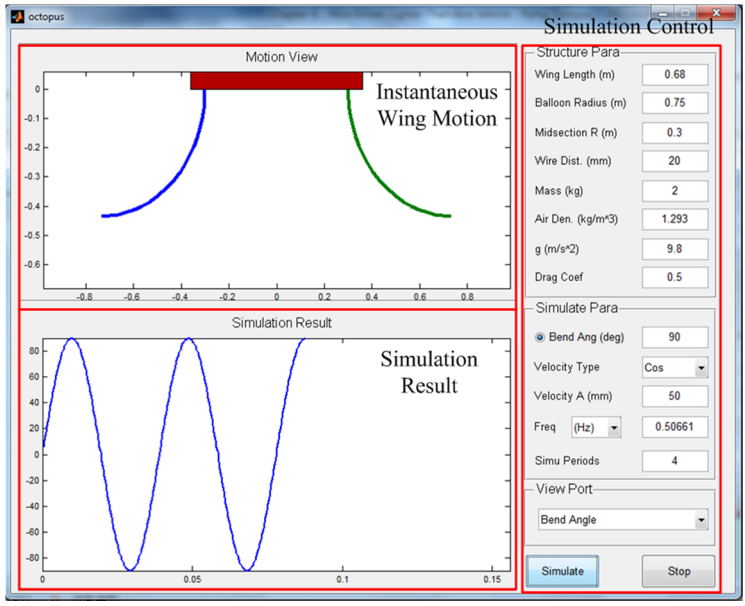


Figure 2-11 Flying Octopus motion simulation.

left window shows the simulation result. The program could simulate the bending angle of the wing, the bending velocity of the wing, the velocity of the tip, the thrust force, the resistance force, the resultant force, and the velocity of the Flying Octopus. The simulation parameters could be input from the right window. The parameters include the structure parameters of the Flying Octopus as well as the simulation parameters. Structure parameters include the length of the wing, radius of the balloon, radius of the midsection, wire pair distance, overall mass, air density and drag coefficient. The simulation parameters include the wing flapping velocity profile (cosine, rectangular wave, etc.), the flapping amplitude, the flapping frequency and the simulation periods.

Figure 2-12 shows one simulation example. In the simulation, the Flying Octopus mass is 1.592 kg, the wing length is 0.68 m, balloon radius is 0.75 mm and drag coefficient is 0.5. The wing flapping amplitudes are all 60° and flapping frequency is 1 Hz. Figure 2-12(a) shows the Flying Octopus velocity. The horizontal axis represents the time (s) and vertical axis is the velocity (m/s). From the results, in the first few flapping cycles the Flying Octopus velocity increase with fluctuation. After 25 seconds, the speed of the Flying Octopus becomes stable. However, the velocity remains fluctuating with the wing flapping motion.

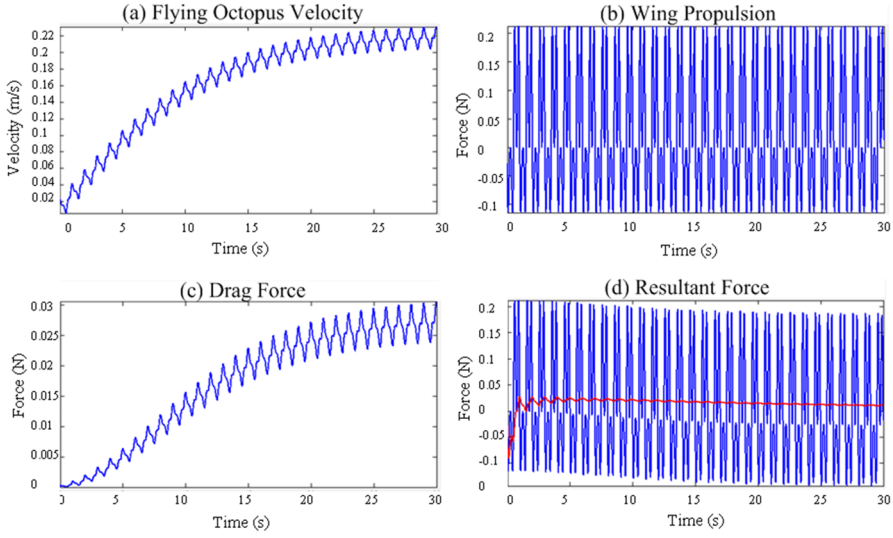


Figure 2-12 Simulation results in 30 flapping cycles.

The average stabilized velocity of the Flying Octopus is 0.215 m/s. Figure 2-12(b) shows the propulsion generated by the flapping wings. It is shown that during one flapping cycle the instantaneous propulsion is between −0.12 N and 0.21 N. The average propulsion is positive, i.e., upward. Figure 2-12(c) shows the drag force. It has a similar trend as the Flying Octopus velocity, which is straightforward. Figure 2-12(d) shows the resultant force acted on the Flying Octopus. The blue curve is the instantaneous resultant force, and the red curve is the historical average. From the result, the averaged resultant force is decreasing after a few seconds. This is also revealed by the velocity increase rate.

2.4 Motion control

In the propulsion simulation, by flapping the wire-driven wings upward propulsion can be obtained. This shows the Flying Octopus is able to take off. To control movement of the Flying Octopus in 3D space, motion control is needed for each wing.

2.4.1 Wing flapping motion

The four wings are orthogonally arranged. They are paired into two groups. To better distinguish them, they are named *X* wing group and *Y* wing group. For each group there are two flapping modes as shown in Figures 2-13 and 2-14.

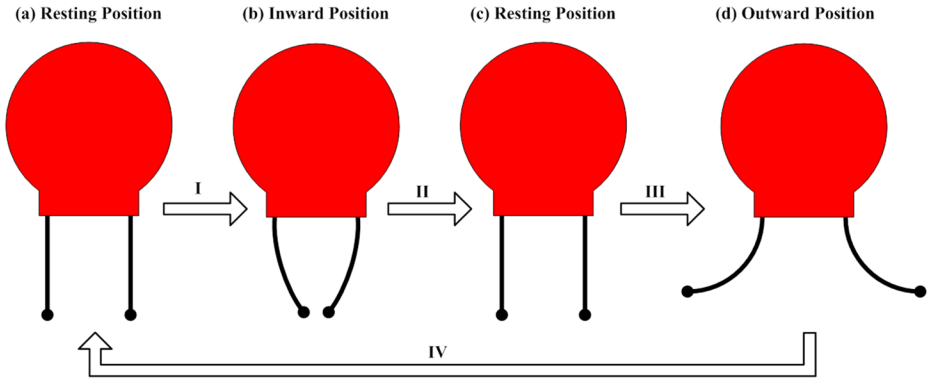


Figure 2-13 Flapping illustration – vertical mode.

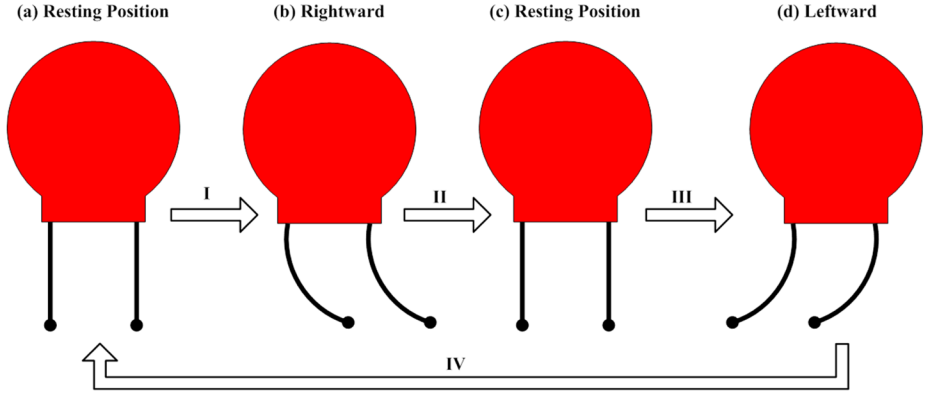


Figure 2-14 Flapping illustration – lateral mode.

In the vertical flapping mode, the two wings flap in opposite directions. In the lateral flapping mode, the two wings flap in the same direction. For both modes, the flapping cycle is composed of four stages, as seen in the figures. For a single wing, during one flapping cycle, the resultant force has two components: one in the vertical plane and the other in the horizontal. The force direction depends on the velocities and amplitudes in the four stages. In the vertical flapping mode, the forces in the horizontal plane generated by the two wings balance each other. The net force is in the vertical plane. From the previous analysis, the propulsion is positively related to the velocity. Therefore, upward propulsion is generated when the velocities in stage II and stage IV are larger than those in stage I and stage III. On the contrary, the propulsion

is downward. In the lateral flapping mode, the two wings flap identically, and forces in the vertical and horizontal plane are both strengthened.

The motion of each wing is controlled via Pulse Width Modulation (PWM). As shown in Figure 2-15, the command is sent to the Micro Control Unit (MCU) by a remote controller. On receiving the command, the MCU generates ion(PWM) signals. The positions and velocities of the servo motors are controlled via the PWM signals. The wire-driven mechanism transmits the motor's rotation to the wings flapping. Thrust and lateral forces are generated to drive the Flying Octopus torso and control its motion.

2.4.2 Flying Octopus motion control

A complex 3D motion can be deconstructed into several basic motions. For the Flying Octopus, there are five motion modes: a) flying upward, b) flying downward, c) hovering, d) flying in *X* direction and e) flying in *Y* direction. In these modes, the wings are controlled as following:

a) Flying upward: Both the *X* and *Y* wing group flap in vertical mode. Before flapping, all wings return to the resting position. The outer flapping amplitude is larger than the inner amplitude. In stages I and III, the velocity is slower than that of stages II and IV. As the four wings flap identically, the lateral forces cancel out each other. Since the inward flapping velocity is larger than the outward flapping velocity, from the thrust representation in Table 2-4, the thrust in stages I and III is smaller than that in stages II and IV. The net thrust in one flapping cycle is upward. Therefore, the Flying Octopus will ascend.

b) Flying downward: Both the *X* and *Y* wing group flap in vertical mode. Compared to upward flying, the difference is that in this mode, the inward flapping velocity is smaller than outward. As a result, the net thrust in one flapping cycle is downward.

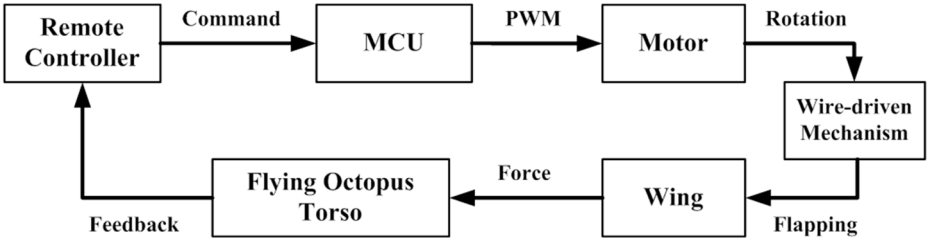


Figure 2-15 Motion control of the Flying Octopus. Source: [18] (with permission).

Table 2-4 Five basic motions.

	Forces by X wing group	Forces by Y wing group
Fly upward	↑	↑
Fly downward	↓	↓
Hovering	↑ or ↓	↑ or ↓
Fly in *X* direction	↑ and →	↓
Fly in *Y* direction	↓	↑ and →

c) Hovering: Ideally, the flapping octopus suspends in the air when the flapping wings keep still. However, disturbances always exist. Gyros can be used to detect the status of the Flying Octopus. When the Flying Octopus is dropping, the wings flap as in upward mode; if the Flying Octopus is ascending, the wings flap as in downward mode.

d) Flying in *X* direction: In this motion mode, a lateral force is needed. The *X* wing group flap in the lateral mode and the *Y* wing group flap in the vertical mode suppress the vertical force generated by the *X* wing group. For example, if the Flying Octopus is desired to travel in $+X$ direction, the wings in *X* group flap slowly toward $+X$ direction in stages I and IV, and flap with greater velocity toward $-X$ direction in stages II and III. The generated forces are a lateral force in $+X$ direction and an upward thrust. A downward force is obtained from the *Y* wing group. Their motions are the same as that in the downward scheme, i.e., fast bending outward and slowly inward. In one flapping cycle the net force is in $+X$ direction. This drives the Flying Octopus to glide in $+X$ direction. Switching the $+X$ and $-X$ flapping velocities of the *X* wing group, the Flying Octopus will move in $-X$ direction.

e) Flying in *Y* direction: In this mode, a net force in *Y* direction is needed. The scheme is similar to that in the previous mode. As the four wings are orthogonally arranged, by switching the motions of *X* wing group and *Y* wing group, the Flying Octopus will move in *Y* direction. The forces generated by the two wing groups in the five basic motions are shown in Table 2-4. In the table, the arrow shows the force direction, and the thickness of the arrow represents the magnitude of the force.

2.5 Prototype and experiment testing

2.5.1 Flying Octopus prototype

A Flying Octopus prototype was developed as shown in Figure 2-16. A polyethylene (PE) balloon with 1.5 m diameter is used as the round body. It is filled with helium to provide buoyance. The backbone in the wing is a carbon beam. Its width is 5 mm and the thickness is 0.5 mm. The vertebras are fabricated by 3D printing and the material used is ABS plastic. The membrane is made from a 0.5 mm thick ABS plate. The overall length of the flapping wing is 680 mm. The four wings are evenly distributed on the middle section. Steel wires covered with plastic sheath are used to control the wing flapping. Four servomotors pull the wires via a rotator under the control of a commercial MCU. Each wing weights 39 g. The total mass of the Flying Octopus is 1592 g.

The details of the Flying Octopus are shown in Table 2-5. The weight of the Flying Octopus is finely adjusted to balance the buoyance. As a result, the Flying Octopus is able to hover in the air when all the wings are resting. At resting, the four wings are relaxed as shown in Figure 2-16(b). By pulling the outer wires, the wings bend outward as shown in Figure 2-16(c). The maximum bending angle is constrained by the stiffness of the backbone. For this prototype, the maximum bending angle is around 180°. Over bending will exert a large movement on the backbone and twist the flapping wing. The Flying Octopus is a low-cost LTAV. The total cost mainly depends on the motor and MCU. For this prototype, the estimated parts cost is less than 150 USD.

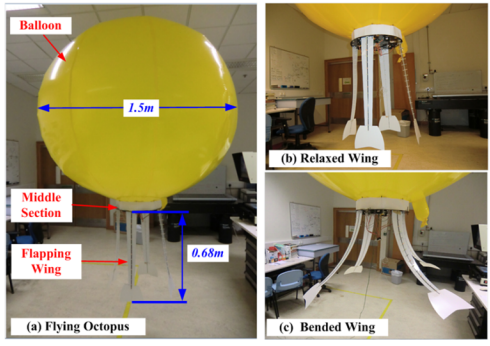


Figure 2-16 Flying Octopus prototype. Source: [18] (with permission).

Table 2-5 Flying Octopus component list.

Component	Description	Mass (g)	Number
Balloon	1.5 m Diameter	568	1
Midsection plate	Carbon	302	1
Motor	Towerpro MG 995	59	4
Drum wheel	ABS Plastic	7	4
Wing membrane	ABS Plastic	39	4
Battery box	4 AAA battery	69	1
Control board	MCU Atmega 128	156	1
Fence	ABS Plastic	38	1
Wire	Steel	0.25	4
Others	Bolts, etc.	38	–

2.5.2 Indoor experiments

Indoor flying tests are carried out. To eliminate wind disturbance, all the air conditioner and exhaust fans are turned off. The weight of the Flying Octopus is adjusted to the same as the buoyance. From the control scheme, it is seen that in the five motion modes, the wing's motion in flying upward, flying downward and hovering are similar. The difference is the flapping speed in the four stages. Flying in X direction and flying in Y direction are similar. Therefore, in the experiment, two fundamental modes, i.e., flying upward and flying in X direction, are tested.

Experiment 1 – flying upward

Figure 2-17 shows the four stages of the wing's flapping cycle in the flying upward mode. At the beginning, the Flying Octopus stays on the ground with all four wings relaxed. Next, the four wings are flexed by the wire-driven mechanism slowly. When reaching the outer amplitude position, the four wings flap back to resting position. From the figure, it can be seen that the time for stage I is 2.08 s, while it only takes 0.46 s for stage II. After getting to the resting position, the wings continue bending inward. When reaching the innermost position, the four wings flap back to the resting position and finish one flapping cycle. The Flying Octopus flies upward 1 m with five flapping cycles. The distance

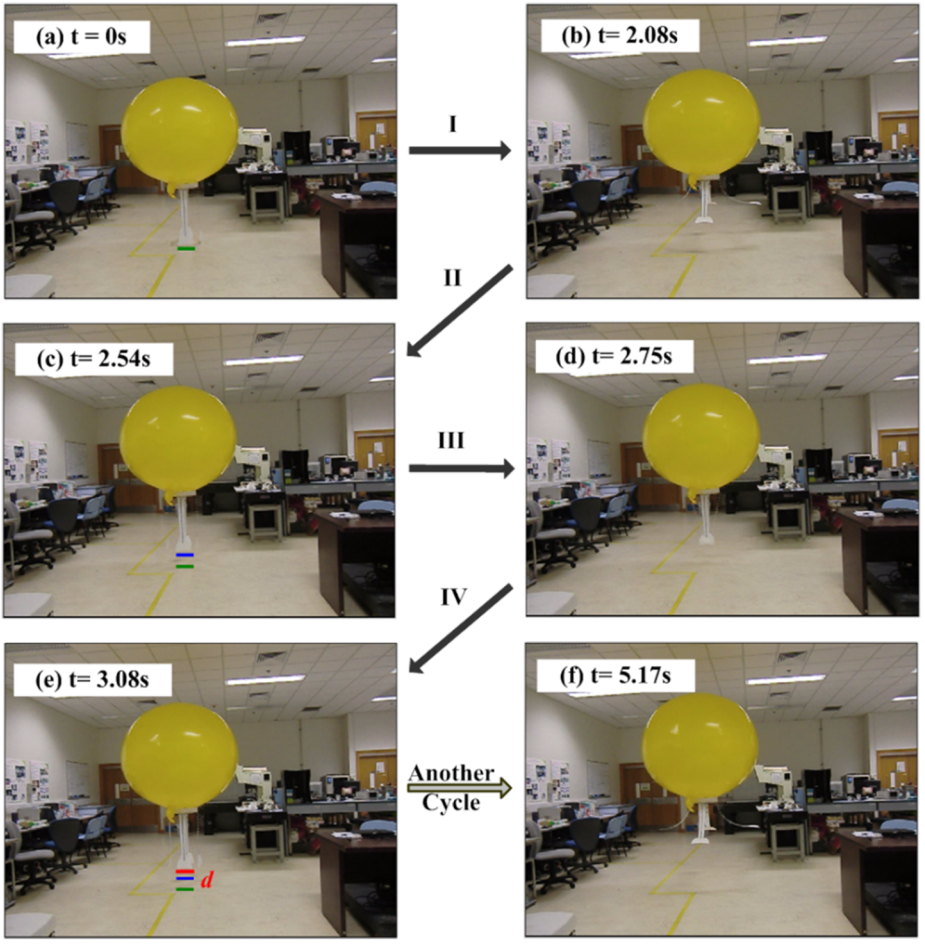


Figure 2-17 Flapping cycle of flying upward. Source: [18] (with permission).

it travels in one cycle is about 20.0 cm, and the average speed is about 6.5 cm/s. As the travelling distance is short, the Flying Octopus may not reach the maximum velocity. This is consistent with the simulation result as shown in Figure 2-18. From the simulation, it can be seen that during the first five flapping cycles, the velocity increases cycle by cycle. The average speed is around 5.3 cm/s.

Experiment 2 – flying in horizontal plane

Figure 2-19 shows the wing's flapping cycle in the mode of flying in *X* direction. At the beginning, all the wings are relaxed. The wings in *X*

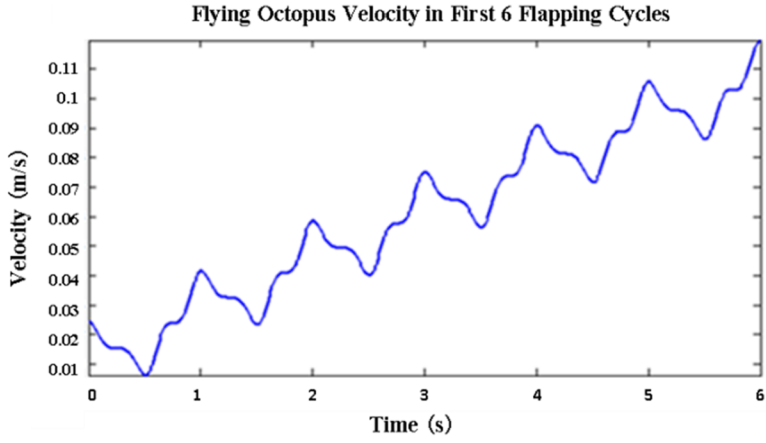


Figure 2-18 Flying Octopus velocity in the first 6 flapping cycles.

group flap to $-X$ direction and the wings in Y group flap inward quickly in stage I. In stage II and stage III, the X wings flap to $+X$ direction and the Y wings flap outward slowly. After that, the X wings flap in $-X$ direction and Y wings flap inward quickly in stage IV. The distance traveled in one flapping cycle is around 25 cm, and the average traversing speed is 7.4cm/s.

Due to the small size of the room, the performance of the Flying Octopus is not fully exhibited in the experiments (such as in the ascending test, where the LTAV reaches the ceiling by five flapping cycles). The maximum velocity should be larger than 6.5 cm/s. It is also expected that the Flying Octopus could move faster after wing shape and flapping parameters optimization. From the simulation, as in Figure 2-12, the maximum ascending speed under this flapping motion is 22 cm/s. The speed can be controlled via several approaches: i) increase the flapping frequency only; ii) increase the flapping amplitude; iii) control the differential flapping speed. As the flapping wings are flexible, the mode frequencies of the wing are low. Hence, flapping frequency cannot increase much. The outer amplitude is around 180° and the inner amplitude is confined by the diameter of the middle section. At large flapping amplitudes, the wing may twist. Also, energy consumption increases sharply at amplitudes close to the limit, due to the large wire tension. The optimal flapping amplitude range is 90°–150°. To increase

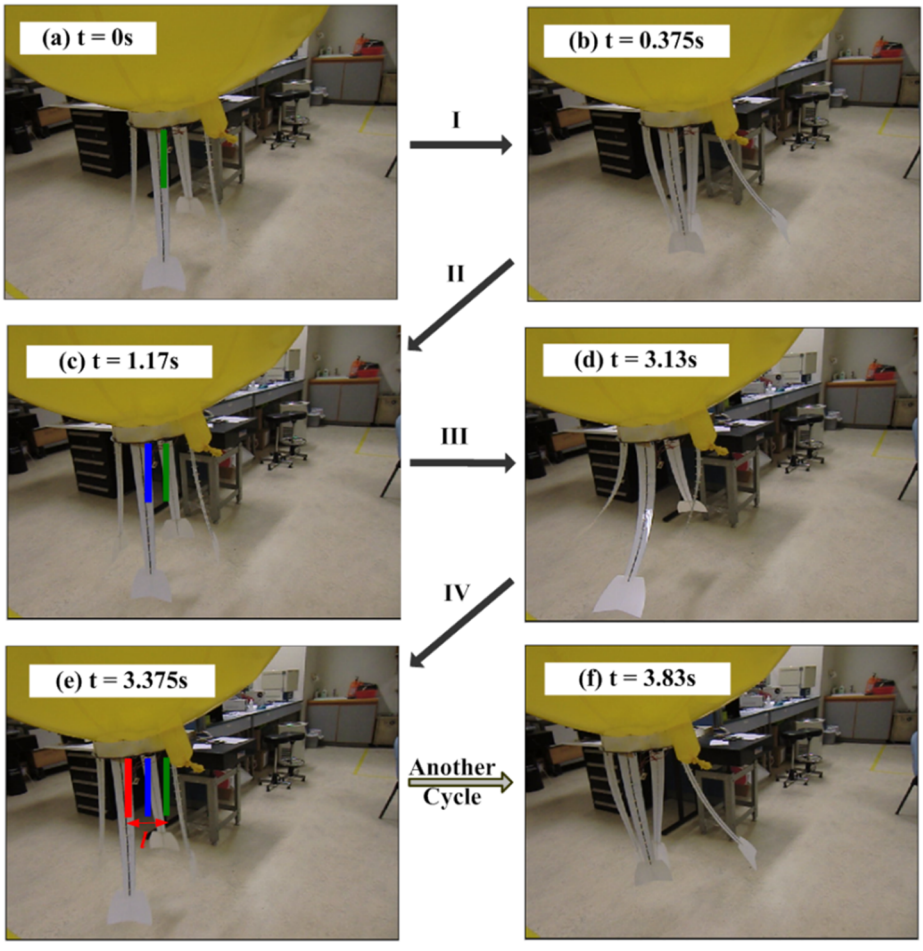


Figure 2-19 Flapping cycle of flying in *X* direction. Source: [18] (with permission).

the speed, the difference between the inward and outward speed should be as large as possible. However, when the outward speed is too small, the overall flapping frequency is reduced. The three approaches should be considered comprehensively in speed control.

Flapping wing is a new application of the wire-driven mechanism, in addition to continuum robot arm and wire-driven robot fish. The wire-driven flapping wing is compact, lightweight and easy to control. Also, the flapping range is quite large. In this prototype, the maximum

flapping angle is about 180°. However, it also has drawbacks. Due to the wing's flexibility, flapping frequency is limited and the wing is inclined to twist at large bending angles. To avoid this twist, the backbone and membrane shouldn't be too slim. Also, a small web is beneficial.

The wire tension in flapping is greatly affected by the wing's mass distribution and wire configuration. The ratio between the wire tension and force acted on the wing is equivalent to the backbone length over the first vertebra's rib length. This is usually large; in this design the ratio is 30. Hence, it is beneficial to reduce the wing's weight, especially the weight at the tip. As a result, the tilted wire configuration is better than parallel wires. The reason is that with a shorter rib length the vertebra weight is reduced while keeping the same ratio.

2.6 Conclusions

This chapter discusses the recent research efforts in making Lighter-Than-Air-Vehicle (LTAV) robots, and introduces a novel LTAV actuated by wire-driven flapping wings. Based on the preceding discussion, the following conclusions can be drawn. First, compared to other methods of flying, LTAVs have advantages and disadvantages. Its advantages include high energy efficiency, good maneuverability and low acoustic noise. Therefore, it would be useful in such applications as exploration, advertisement and entertainment. Second, we built an LTAV robot called Flying Octopus. Its "head" is a helium balloon and its four "wings" are driven by four one-segment planar continuum wire-driven mechanisms. The wire-driven mechanism makes the wings compact, lightweight and easy to control. Also, the flapping range is quite large. The Flying Octopus can effectively fly upward and downward, hover, and fly in X direction and Y direction. It is well suited for indoor entertainment and advertisement.

In the future development of LTAVs, efforts should be made in the following areas. For indoor LTAVs, the look and drive methods can be more diversified. In nature, there are vast samples, mainly water species. Current indoor LTAVs fly slowly and elegantly. In the future LTAVs may be more agile. For outdoor LTAVs, stability is a primary concern. If disturbances from wind can be addressed on outdoor LTAVs, especially smaller ones, LTAVs will have a very promising future.

2.7 References

[1] L. Winter and G. Degner, *Minute epics of flight*. New York: Grossett & Dunlap, 1933.

[2] L. Liao and I. Pasternak, "A review of airship structural research and development," *Progress In Aerospace Sciences*, vol. 45, pp. 83–96, May–Jul 2009.

[3] AngMoKio, *Zeppellin NT*. Available: http://en.wikipedia.org/wiki/File:Zeppellin_NT_amk.JPG (visited on Nov. 11, 2013).

[4] *Flying Yachts*. Available: http://www.theflyingyacht.com/index.php/about/ (visited on Nov. 11, 2013).

[5] B. M and e. Lando M, "Peculiar performance of a new lighter-than-air platform for monitoring," in *AIAA 4th Aviation Technology, Integration and Operations Forum*, Chicago, IL, USA, Sept. 2004, pp. 1–14.

[6] G. A. Khoury and J. D. Gillett, "Airship Technology," ed. Cambridge, UK: Cambridge University press, 1999.

[7] *Air Swimmer*. Available: http://airswimmers.com/ (visited on Nov. 11, 2013).

[8] Festo. (2010). *Airjelly*. Available: http://www.festo.com/cms/en_corp/9771.htm (visited on Nov. 11, 2013).

[9] W. M. Corporation. (2011). *Airswimmers*. Available: http://airswimmers.com/ (visited on Nov. 11, 2013).

[10] Z. Li and R. Du, "Design and Analysis of a Bio-Inspired Wire-Driven Multi-Section Flexible Robot," *Int J Adv Robotic Sy*, vol. 10, 2013.

[11] Z. Li and R. Du, "Design and analysis of a biomimetic wire-driven flapping propeller," in *Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS & EMBS International Conference on*, 2012, pp. 276–281.

[12] Z. Li and R. Du, "Design and implementation of a biomimetic wire-driven underactuated serpentine manipulator," *Transaction on Control and Mechanical Systems*, vol. 1, pp. 250–258, 2012.

[13] Z. Li, R. Du, M. C. Lei, and S. M. Yuan, "Design and Analysis of a Biomimetic Wire-Driven Robot Arm," in *Proceedings of the*

ASME 2011 International Mechanical Engineering Congress & Exposition, 2011, pp. 11–17.

[14] Z. Li, R. Du, Y. Zhang, and H. Li, "Robot Fish with Novel Wire-Driven Continuum Flapping Propulsor," presented at the *The 2nd International Conference on Mechatronics and Applied Mechanics (ICMAM)*, Hong Kong, 2012.

[15] Z. Li, R. Du, Y. Zhang, and H. Li, "Robot Fish with Novel Wire-Driven Continuum Flapping Propulsor," *Applied Mechanics and Materials*, vol. 300, pp. 510–514, 2013.

[16] Z. Li, W. Gao, R. Du, and B. Liao, "Design and Analysis of a Wire-Driven Robot Tadpole," in *International Mechanical Engineering Congress & Exposition (IMECE 2012)*, Houston, Texas, USA, 2012.

[17] B. Liao, Z. Li, and R. Du, "Robot Tadpole with a Novel Biomimetic Wire-driven Propulsor," presented at the *IEEE International Conference on Robotics and Biomimetics (ROBIO 2012)*, Guang Zhou, China, 2012.

[18] Z. Li, R. Du, and Y. Yao, "Flying Octopus - A LTAV with Wire-Driven Flapping Wings," in *International Mechanical Engineering Congress & Exposition (IMECE 2012)*, Houston, Texas, USA, 2012.

3. Visual attitude estimation and stabilization of flying robots

Cihat Bora Yiğit and Erdinç Altuğ, Istanbul Technical University, Turkey

Abstract: This chapter deals with visual attitude estimation in flying robots, and using this information to stabilize them. Unmanned Air Vehicles (UAVs) are currently in widespread use in many applications ranging from military operations to civilian tasks. Successful control of a UAV requires accurate and fast estimation of the vehicle attitude. Usually inertial navigation systems (INS) are used to obtain this attitude information in UAVs. As an alternative or as an additional sensor, vision systems can also be used to obtain vehicle states. Vision systems are readily available on various UAV platforms and can be used for this purpose. The use of vision for attitude estimation is reliable and affordable. In this chapter we present the use of a vision system that can be used to estimate vehicle attitude. Unlike previous works that use natural or artificial features such as blobs or parallel lines on the environment, this work involves the use of no special feature but the natural scene. Processing of this natural scene around the robot leads to attitude information which is used by the control algorithm to stabilize the robot. The vision processing and control are performed on board the vehicle using a vision computer. First, the algorithm, the UAV modeling and control are presented. A quadrotor flying robot is chosen as an experimental platform in this study. Second, a detailed presentation of the developed quadrotor system and experimental set-up are given. Finally, we present the experiments and the results of the estimation and control algorithms.

3.1 Unmanned Aerial Vehicles

Unmanned Air Vehicles (UAVs) have been widely used for various tasks in recent years. UAVs have clear advantages to manned aircraft, such as higher maneuverability, lower cost, decreased radar signature, strength and decreased risk for human life. These advantages led to the use of these vehicles in many applications, from military operations to civilian tasks, and extensive research has been carried out in many laboratories.

Hovering ability is the most important step toward the autonomy of flying vehicles. In order to control a flying vehicle at least six parameters (pose of the vehicle) should be known; Euler angles representing the orientation of the vehicle (attitude) and a vector of coordinates, representing the position of the vehicle. In order to determine the pose of the vehicle accurately and rapidly, the regular approach is to use inertial sensors with other sensors and apply sensor fusion. Some sensors used for this purpose are the Global Positioning Sensor (GPS) and inertial navigation sensor (INS), as well as other sensors such as altitude sensors and speedometers. These sensors have some limitations. The GPS sensor, for example, is not available at some locations or its readings are subject to error. It is also subject to jamming in hostile areas. INS has the disadvantage of accumulation of errors. To overcome these limitations, vision-based navigation approaches have been developed. These approaches can be used where GPS or INS systems are not available, or can be used with other sensors to obtain better estimates. The use of vision systems for unmanned systems increased considerably in the last 30 years. Visual odometry, stereo camera pairs, obstacle avoidance, automatic landing and take-off, navigation, combining vision with GPS and IMU sensors, visual servoing, inspection for pipelines or bridges, road traffic monitoring, target detection and tracking, and mapping are some of the fields that vision systems have been used for in unmanned aerial systems.

The problem of enabling a flying robot to navigate autonomously in unstructured and unknown indoor or outdoor environments draws great attention from the robotics community (Figure 3-1). The difficulties of operating in unknown environments are twofold; detecting the environments and detecting the ego motion. Additionally, having a sensor onboard the flying system that can sense the environment as well as the ego motion is a hard-to-solve problem. Such a system requires additional weight, additional energy consumption and considerable computational power onboard. These requirements could not be satisfied until recent technological developments.

Navigation consists of changing attitude as well as translational coordinates. Attitude is the most important parameter for an autonomous UAV during flight. In this chapter, we present a vision system that can be used to estimate vehicle attitude using only a monocular pinhole camera. Vision systems are readily available on various UAV platforms and can be used for this purpose. The use of vision for attitude estimation

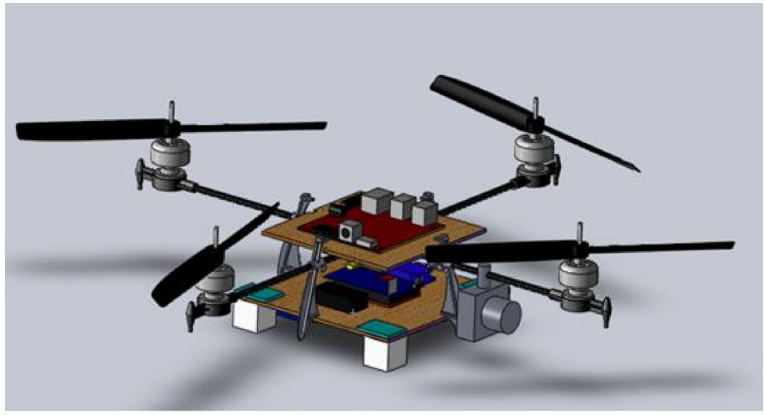


Figure 3-1 Robot in an unknown environment.

is reliable and affordable. Unlike previous work, the proposed approach does not require (natural or artificial) parallel lines on the environment. Moreover, vision processing and control are performed onboard the vehicle, as opposed to using a remote vision computer to process the images. The presented approach is based on the Visual SLAM algorithm [1], in order to locate the quadrotor using a single camera. Unlike previous work that uses external sensors (e.g., VICON), ground markers or landmarks, additional onboard sensors (e.g., IMU), additional or special purpose cameras or an integration of multiple sensors, the proposed algorithm relies on onboard vision only.

There are various different types of rotorcraft; helicopters, tandem rotorcrafts, co-axial rotorcrafts, quadrotor and tilt-rotor vehicles have been developed by various different studies [2]. The quadrotor platform is one of the flying robot platforms that gained the interest of the robotics and control community within last decade. Although the proposed vision algorithm and vision computer is equally applicable to any UAV type, in this work a quadrotor flying robot has been chosen due to its multiple advantages over other platforms.

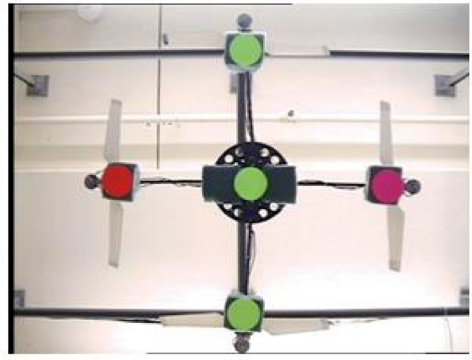
This chapter is organized as follows. In Section 3.2, we will present the use of Attitude Estimation with vision in UAVs. After an explanation of the algorithm that is used to estimate the attitude, we will present quadrotor helicopter modeling and control in Section 3.3. The details of the flying robot helicopter design are provided in Section 3.4. The experiments of the estimation and the control algorithms performed on

a model helicopter are presented in Section 3.5, followed by concluding remarks and future work suggestions in Section 3.6.

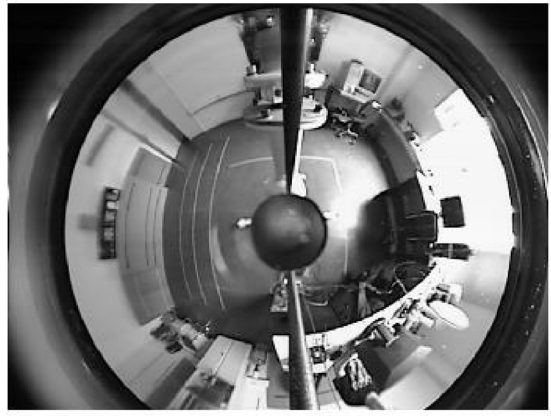
3.2 Attitude estimation with vision

Attitude (or pose) is the most critical parameter in a UAV. Successful operation of the UAVs requires fast and accurate estimation of attitude information. The usual approach in attitude estimation is to use a dedicated sensor (IMU). Recent studies on estimation of the attitude information use vision only, and some studies estimate this information using vision with other sensors. Visual sensors are reliable, and they can be used to determine attitude information. Recent studies use artificial features to help the detection procedure, as shown in Figure 3-2. Artificial features can be placed in the environment to determine the relative orientation or relative displacement, as well as velocity. Artificial features can also be placed on flying robots to determine them from known camera locations or by other flying robots.

Some studies use the capabilities of catadioptric imaging systems. The main advantage of a catadioptric camera is the increased field of view, enabling a more wide view that can be used to estimate attitude information. The estimation of roll and pitch angles of a UAV by determining the horizon has been studied [3, 4]. In these studies, the horizon was determined with Random Markow Fields or RGB based Mahalanobis distance. Unfortunately, this approach requires conditions where the horizon is visible, which is not always the case for a UAV. In addition, it cannot be used to estimate the yaw angle. A hybrid method that uses the horizon and the homography was also proposed [5]. To prevent the limitations of horizon detection, another approach was proposed that uses lines that are usually available in urban areas [6]. In this approach, it is still not possible to estimate the yaw angle; also it requires determining the sky. Therefore, this approach is not suitable in dense city environments or closed areas. A more recent paper proposes the use of vanishing points and infinite homography to estimate helicopter attitude in urban environments [7]. More recently, implementation of attitude estimation with a catadioptric system and implementation on a quadrotor s presented [8]. In this study, authors used an omnidirectional camera to estimate quadrotor attitude, improved it using a Kalman Filter and showed successful stabilization experiments.



Color Blobs



Artificial Lines

Figure 3-2 Using artificial features on flying robots or on the environment.

Recent research on UAV attitude information includes [9] who obtained vehicle localization using a vision based visual SLAM algorithm based on algorithm presented in [10]. Davison et al. [1], presented a real-time algorithm which can recover the 3D trajectory of a monocular camera. Achtelik et al. [11] presented on-board IMU and monocular vision-based control for MAVs. In Eberli et al. [12], two concentric circles were used as landmarks for UAV vision-based control. Achtelik et al. [13] used a laser range finder and stereo camera for autonomous helicopters. Hovering flight and vertical landing control of a VTOL

unmanned aerial vehicle using optical flow is presented in [14]. The problem of estimating the pose of an object in real time using visual servoing is presented in [15]. Some researchers used indoor positioning sensors [16], or ground-based sensor modules [17] for indoor localization, and ground markers [18] which limit the usability and mobility of the UAV systems. Some researchers used visual-inertial sensor fusion. Jones and Soatto [19] presented a visual-inertial navigation system for estimating the motion of a sensing platform. Kelly and Sukhatme [20] presented visual-inertial sensor fusion. Rondon et al. [21] used inertial sensors with vision for a two-rotor VTOL miniUAV. Fusion of inertial and visual data for position, velocity and attitude estimation was presented in [22]. Johnson [23] used vision with IMU information for indoor operations.

Recent studies on monocular vision-based pose estimation take two main directions. The first approach (e.g., [1]) tracks features on images and uses different Kalman filters to estimate the pose of the camera. This approach is generally called SLAM. The second approach uses the SFM reconstruction technique (e.g., [10]), and separates tracking and mapping operations.

In this study, the higher priority task is the localization; therefore the V-SLAM approach used in [1] was taken. The other reason for this selection is the limited computational power available. The goal of having all processing onboard the quadrotor requires the use of an embedded computer, beagle-board [24], on the system. Unfortunately, this selection limits the processing power available for attitude estimation.

The algorithm works as follows: images are captured by the camera as the quadrotor moves, and the images are transferred to the beagle-board for processing. The code is based on extended Kalman filter, and it involves 13 states of the quadrotor. An algorithm determines natural features on images, and tracks these images actively as the camera moves. An algorithm called "goodfeaturestotrack" determines the dominant corners to track. For each corner 6 states are added to the Kalman filter. At least 5 corners are determined from each image, and if the corners are below this number, new corners are added. The motion of the camera (and therefore the helicopter) is modeled and predicted before the motion, and as the camera moves the attitude information is updated (Figure 3-3). The camera is expected to move in constant velocity, but the constant speed model can deal with accelerations and treat them

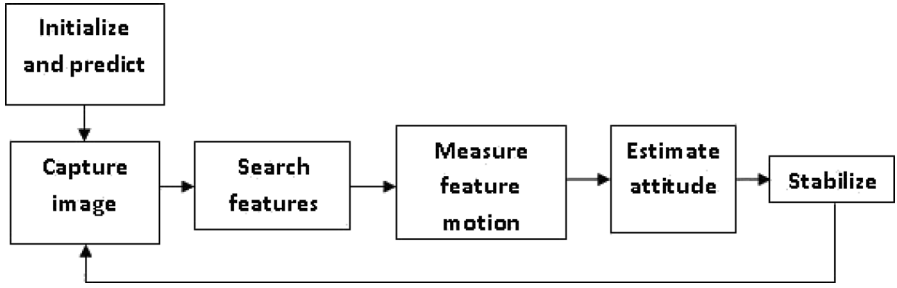


Figure 3-3 Simplified block diagram of the algorithm.

as noise. An initialization routine is required to provide initial attitude information, which can also be provided with a calibration pattern.

The algorithm has some limitations that should be considered in UAV applications. For accurate attitude estimation we assume that there are no moving objects in the field of view, helicopter velocity is constant during the experiment, excessive camera motion is not available, and there are no repeating big patterns on images (such as unique floor pavements), which may lead to attitude estimation errors.

The proposed UAV system is a mini-size quadrotor helicopter. The algorithm is implemented on Beagle-board onboard the quadrotor. Some samples of images captured by the camera as it is looking at a random scene are presented in Figure 3-4. The camera is moved through the experiment, as if it is a moving UAV. As the algorithm runs, determined and tracked features on the images are marked with a white box. Additionally, obtained attitude information is stamped on the images for easy verification. The estimated attitude information is very close to the real attitude information. This experiment verifies that the algorithm can be run on a small sized embedded computer successfully.

3.3 Quadrotor UAV modeling and control

There are various different types of UAV rotorcrafts: helicopters, tandem rotorcrafts, co-axial rotorcrafts, quadrotor and tilt-rotor vehicles have been developed by various different studies [2]. The quadrotor platform is one of the flying robot platforms that the gained interest of the robotics and control community in the last decade. The ease of control and superior hovering ability of this platform, as well as the simplicity of its design (as opposed to complex swash plate mechanisms and tail rotors of regular helicopters), leads to its use in many research

Figure 3-4 Images captured by the camera and obtained attitude information. Source: [25] IEEE, 2012 (with permission).

projects. In this work, the quadrotor flying robot has also been chosen due to these reasons.

A quadrotor is an under-actuated aircraft with four fixed pitch angle rotors [26, 27]. It contains four motors, located on the front, back, left and right of the airframe. The rotors connected to these motors provide the necessary lift forces. The helicopter has four input forces which are basically the thrust provided by each propeller. The front and rear motors rotate counterclockwise, while other motors rotate clockwise. So yaw command is derived by increasing (decreasing) the counter-clockwise

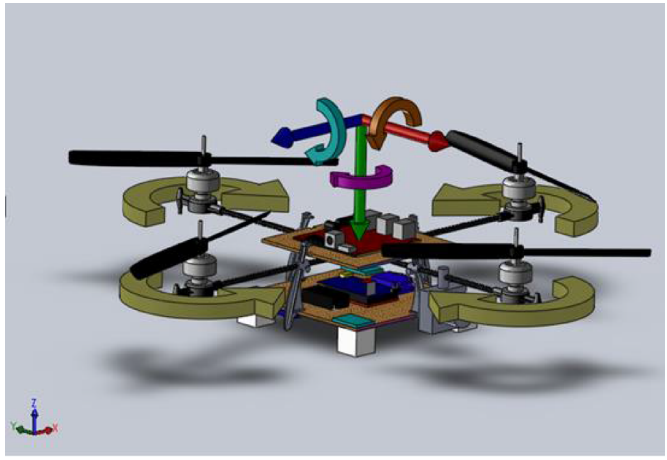


Figure 3-5 Robot and its axes.

motor speed while decreasing (increasing) the clockwise motor speeds. This also eliminates the need for a tail rotor (Figure 3-5).

In this section we will describe the mathematical model of the quadrotor. This four-rotor aerial vehicle has been modeled using Newton-Euler equations. The presented model is a slightly modified version of the quadrotor model presented in [26]. The coordinate axes, the rotation directions of the quadrotor, the lift forces and the Euler angle descriptions are given on Figure 3-6.

For the rigid body model of a 3D quadrotor given in Figure 3-6, a body fixed frame (frame B) is assumed to be at the center of gravity of

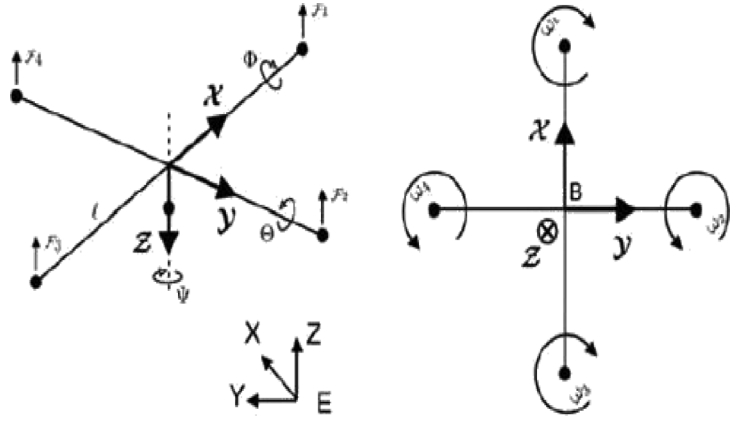


Figure 3-6 Forces acting on a quadrotor.

the quadrotor; an inertial frame (frame E) is located on the ground. The Euler angles corresponding to the rotations around the x, y, and z axes are given respectively as roll (ϕ), pitch (θ) and yaw (ψ). The forces acting on the quadrotor are F_1, F_2, F_3, F_4 and mg. The rotor spinning directions are determined to be clockwise for the first and third rotors and counterclockwise for second and the fourth rotors.

The 3D position of any point on axis E can be represented with a vector $P = [x\ y\ z]^T$. The linear velocity (v) and linear acceleration ($\dot{v}$) of the vehicle can be obtained by taking the derivative of this vector.

A rotation matrix $\mathbf{R} : \mathrm{E} \rightarrow \mathrm{B}$, where $R \in SO(3)$ is composed of three Euler angles (ψ, θ, ϕ), representing yaw, pitch and roll respectively. The axes have been chosen as N.E.D. (North, East, Down).

The rotation speed of the rotors is described by i, where i corresponds to the rotor number (i: 1,2,3,4). The lift forces generated by the rotation of the rotors will be $F_i = b\Omega_i^2$. The parameter b is a constant. The total lift force will be summation of the lift forces of all four rotors ($F_1+F_2+F_3+F_4$).

The dynamics of a free body subject to external forces can be presented using the Newton-Euler equations as:

$$\begin{bmatrix} mI_{3\times3} & 0 \\ 0 & I \end{bmatrix}\begin{bmatrix} \dot{V} \\ \dot{\omega} \end{bmatrix}+\begin{bmatrix} \omega\times mV \\ \omega\times I\omega \end{bmatrix}=\begin{bmatrix} F \\ \tau \end{bmatrix} \tag{1}$$

where $I \in R^{3\times3}$ is the inertia matrix, V is the linear body velocity, ω is the body angular velocity vector, F is the external force, τ is the external torque, and m is the mass of the system. One can obtain the following set of equations:

$$\begin{aligned} &\ddot{x} = \sum F_x/m \quad \ddot{y} = \sum F_y/m \quad \ddot{z} = g - \sum F_z/m \\ &\ddot{\phi} = \sum \tau_x/I_{xx} \quad \ddot{\theta} = \sum \tau_y/I_{yy} \quad \ddot{\psi} = \sum \tau_z/I_{zz} \end{aligned} \tag{2}$$

Let us assume the hub forces and the flapping moments can be ignored. By simplifying the model, we can represent the system in state space form, with U input vector as $U = \{U_1, U_2, U_3, U_4\}$ and X as the state vector, $X = \begin{bmatrix} x & \dot{x} & y & \dot{y} & z & \dot{z} & \phi & \dot{\phi} & \theta & \dot{\theta} & \psi & \dot{\psi} \end{bmatrix}^T$. Let us choose the inputs as:

$$U = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ b(-\Omega_2^2 + \Omega_4^2) \\ b(-\Omega_1^2 + \Omega_3^2) \\ d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \end{bmatrix} \tag{3}$$

where Ω_i is the speed of rotation of rotor i, b is the push factor, and d is the factor relating torques to forces. Full dynamics of the helicopter can then be obtained as:

$$f(X, U) = \begin{bmatrix} \dot{x} \\ (c_\phi s_\theta c_\psi + s_\phi s_\psi)\frac{U_1}{m} \\ \dot{y} \\ (c_\phi s_\theta s_\psi - s_\phi c_\psi)\frac{U_1}{m} \\ \dot{z} \\ -g + (c_\phi c_\theta)\frac{U_1}{m} \\ \dot{\phi} \\ \dot{\theta}\dot{\psi}\frac{I_{yy} - I_{zz}}{I_{xx}} - \frac{J_r}{I_{xx}}\dot{\theta}\Omega + \frac{lU_2}{I_{xx}} \\ \dot{\theta} \\ \dot{\theta}\dot{\psi}\frac{I_{zz} - I_{xx}}{I_{yy}} - \frac{J_r}{I_{yy}}\dot{\theta}\Omega + \frac{lU_3}{I_{yy}} \\ \dot{\psi} \\ \dot{\phi}\dot{\theta}\frac{I_{xx} - I_{yy}}{I_{zz}} + \frac{U_4}{I_{zz}} \end{bmatrix} \tag{4}$$

where $\Omega = (-\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4)$, g is the gravitational acceleration, J_r is the rotor inertia, and l is the distance between the rotor center and geometrical center of the helicopter.

In this study, we used the following PD controllers:

$$
\begin{aligned}
U_1 &= \frac{mg}{C_\theta C\phi} + \frac{m(Kp_z(z - z_d) + Kd_z(\dot{z} - \dot{z}_d))}{C_\theta C\phi} \\
U_2 &= K_{p\phi}(\phi_d - \phi) + K_{d\phi}(\dot{\phi}_d - \dot{\phi}) \\
U_3 &= K_{p\theta}(\theta_d - \theta) + K_{d\theta}(\dot{\theta}_d - \dot{\theta}) \\
U_4 &= K_{p\psi}(\psi_d - \psi) + K_{d\psi}(\dot{\psi}_d - \dot{\psi})
\end{aligned}
\tag{5}
$$

where ϕ_d, θ_d, ψ_d are the desired roll, pitch and yaw angles, and z_d is the desired altitude. Usually the desired roll and pitch angles are zero. The ϕ_d, θ_d, ψ_d are the desired derivatives of the roll, pitch and yaw angles. U_1 signal controls the motion along z axis, U_2 controls motion along y-axis (roll angle), U_3 controls the motion along the x-axis (pitch angle) and U_4 controls rotation along the z-axis (yaw angle). The designed controllers should set values to U_i parameters, which determine the four rotor speed parameters, Ω_i. A low-level controller is still needed to keep this speed constant for each motor. To control the motions along x-axis and y-axis, ϕ and ϕ angles, and their derivatives, should be controlled. The desired values of the roll and pitch values and the desired roll and pitch velocities will depend on the desired values of the x and y locations [27]. Once we can select these desired values, we can place them in Equation 5.

3.4 Robot design and manufacturing

The first step to test the estimation and stabilization algorithms was to develop a UAV. We decided to develop our customable flying robot. The idea behind this choice was to obtain the ability to adjust and tune as many parameters as possible. With the use of as much COTS elements as possible, the UAV can be developed rapidly, can be easy to repair, be reproducible by the robotics community and be low cost. The design has various elements; quadrotor body, motors, rotors, motor drivers, controller, vision computer, camera, batteries, and communication card. The block diagram of the developed system is shown in Figure 3-7.

The robot involves multiple parts as shown in Figure 3-8. Successful motors, rotors and motor drivers are selected based on various lift and

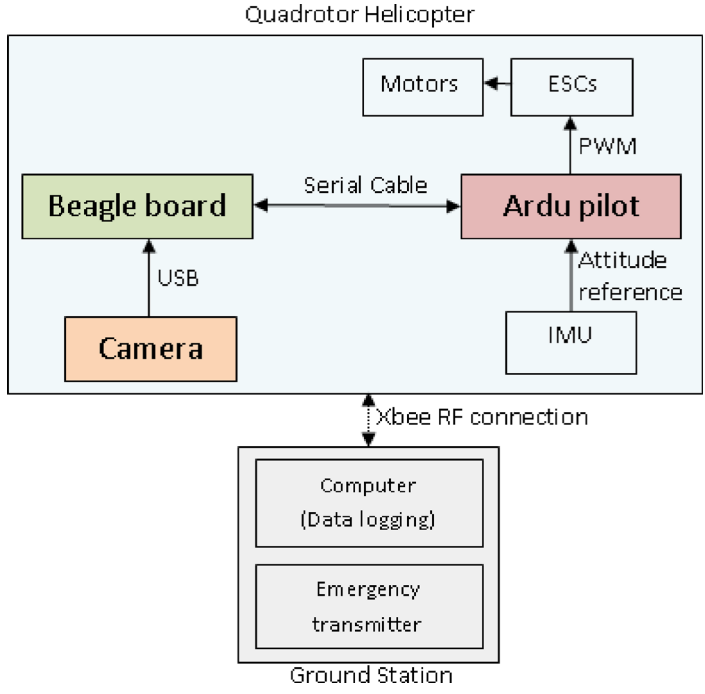


Figure 3-7 The system block diagram. Source: [25] IEEE, 2012 (with permission).

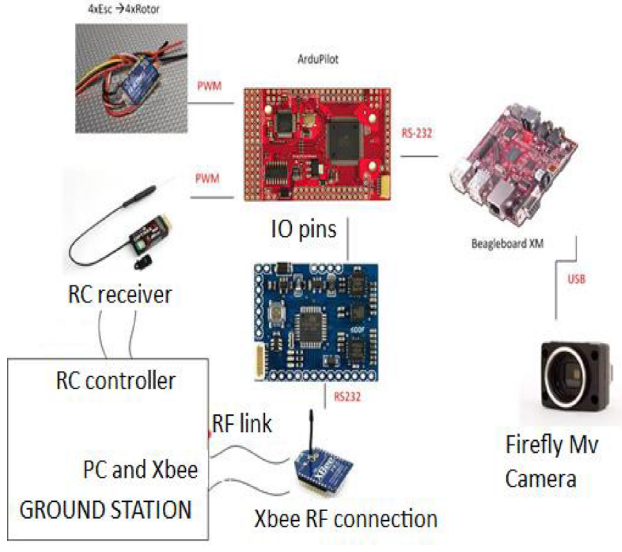


Figure 3-8 Parts of the robot.

power consumption tests. TowerPro brushless outrunner 2410-08T 890 kv motors with H-King 20A fixed-wing brushless speed controllers form the propulsion system on the vehicle.

The quadrotor controller is an ArduPilot Mega UAV controller. In addition to ArduPilot, a Mega IMU shield board was used for sensing. This board features a collection of sensors including three axis angular rotation and accelerations sensors, an absolute pressure and temperature sensor, and a 16 MBits data logger chip. Additionally a telemetry module was installed to obtain vehicle data remotely.

Vision is a critical part of the helicopter, since it is used as the principal source of UAV attitude. This requires an onboard computer to process the visual data captured by the camera. In this way, autonomy of the UAV is increased since it is no longer dependent on, and limited by, a communication link to a ground processing station. And so related time delays are eliminated. A Point Grey FireFly MV camera has been selected and installed on the quadrotor for image collection. It is connected to the Beagle-board through a USB connection.

On the Beagle-board there is an operating system built with the help of the http://narcissus.angstrom-distribution.org web site. The serial port communication and the OpenCV library have been enabled on the operating system. The code running on the ArduPilot is based on ArducopterNG v1.0 code. There have been many changes performed on the code, especially on communications and Beagle-board integration. The V-SLAM code of Davison et al. [1], has been buily using the OpenCV library on Beagle-board. A serial port was used for communication between Beagle-board and ArduPilot. Images captured by the camera are processed on the Beagle-board, and the three Euler angles obtained from the V-SLAM code are then transferred at 10 Hz to ArduPilot. The quadrotor uses this information for the stabilization. IMU available on ArduPilot is only used for comparison. Information transferred from Beagle-board to ArduPilot, IMU data, time stamp, and reference input signals are transferred to a ground computer through an RF link with zigbee. The ground computer saves this information for subsequent evaluation.

The developed quadrotor is shown in Figure 3-9, where 1 is the beagle-board, 2 is the ArduPilot and the IMU board, 3 is the RF module, and 4 is the camera. The developed system has been successfully tested in a laboratory test area.

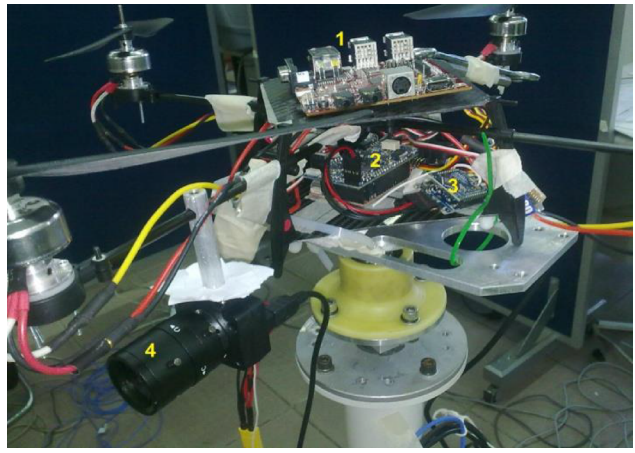


Figure 3-9 Developed quadrotor helicopter on the test stand. Source: [25] IEEE, 2012 (with permission).

In order to limit the flight vibrations on the vision system, some alterations were done to the robot (Figure 3-10). A developed quadrotor can be seen on a test flight in Figure 3-11. In this flight test, the robot was controlled with an RC controller in manual mode. The floors and walls of the laboratory area were covered with foam tiles to limit the potential damages to the robot in case of failure.

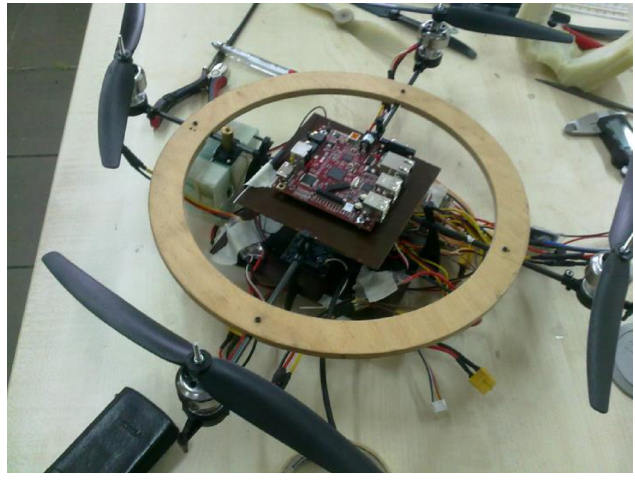


Figure 3-10 Developed quadrotor.

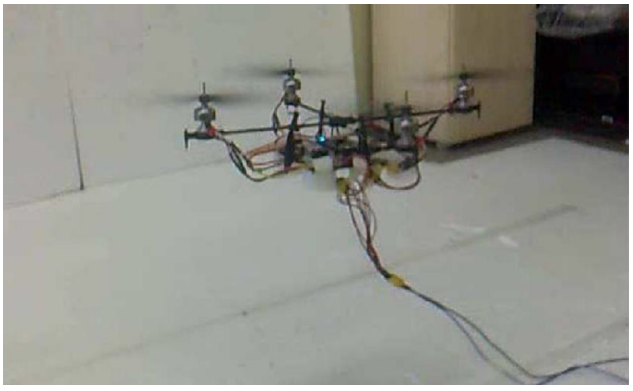


Figure 3-11 Remote controlled of the developed quadrotor robot in laboratory. Source: [25] IEEE, 2012 (with permission).

3.5 Experiments

In this section, we will present the attitude estimation and attitude stabilization experiments. In order to accomplish these experiments, two custom set-ups were developed. The first set-up was used to experiment a single degree of freedom of the quadrotor. In this set-up pitch or roll angles can be experimented on (Figure 3-12).

Later, a more advanced set-up was developed to test the full attitude control of the quadrotor. The experiment system shown in Figure 3-13 consists of a model quadrotor helicopter, a test stand and a camera. The test stand was developed in order to perform successful and secure

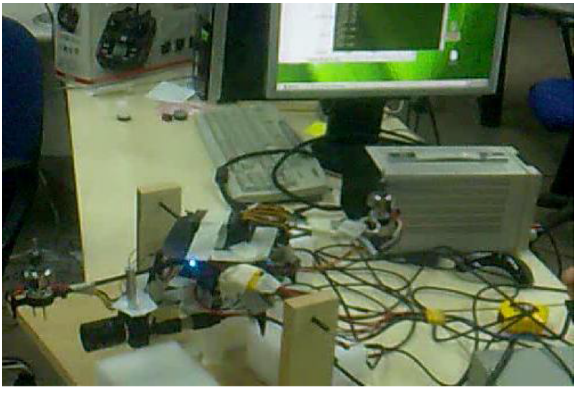


Figure 3-12 The experimental test stand that is used to experiment one axis of the quadrotor control.

Figure 3-13 The experimental test stand that is used to experiment on quadrotor attitude and altitude control.

experiments. The test stand allows the helicopter to perform yaw motions freely, and allows up to 2 meters of altitude, as well as up to ± 20° roll and pitch motion. The vertical motion of the helicopter will not be used in this experiment.

The first experiment involved estimation of the full attitude of the quadrotor. In this experiment, the quadrotor is moved by radio transmitter, and the attitude estimation algorithm is expected to estimate quadrotor attitude correctly. While VSLAM algorithm estimates the pitch and roll angles, IMU provides pitch and roll angles for comparison. The experiment results presented in Figure 3-14 show the full attitude variation of the quadrotor during the experiment. The IMU readings are very close to the VSLAM estimations. Mean and standard deviation of pitch and roll angles are found to be (0:80, 1:53) and (0:44, 2:36) degrees, respectively. The biggest reason for the mismatch is the

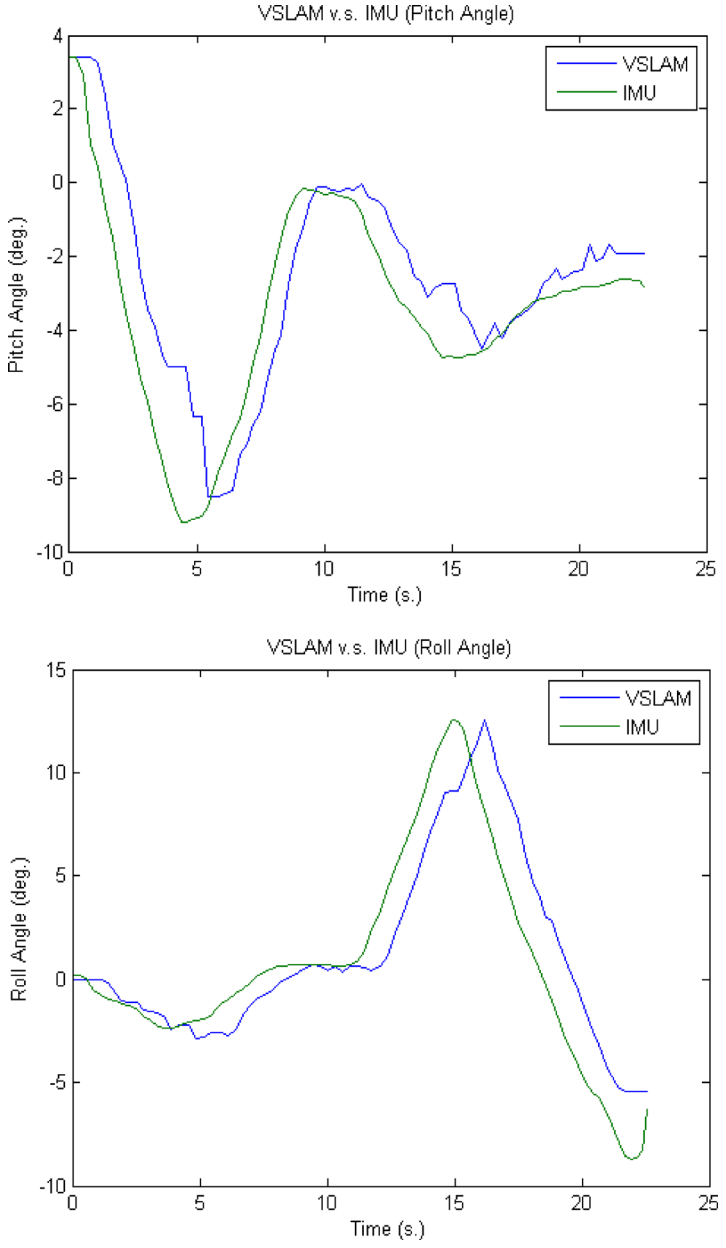


Figure 3-14 The first experiment was on full attitude estimation of quadrotor. Source: [25] IEEE, 2012 (with permission).

lag of about 1 second visible on the graphs. The experiment results show that the estimation using VSLAM algorithm is highly accurate.

The yaw angle estimation was performed in a separate experiment. In this experiment the VSLAM algorithm is used to estimate the yaw, and a magnetometer is used to provide quadrotor yaw. Experiment results show that the VSLAM based yaw estimation has estimation error mean of 0.5612 degrees and standard deviation of 0.83 degrees (Figure 3-15).

The second experiment is about attitude stabilization. In this experiment, the attitude values obtained from VSLAM algorithm are used to stabilize quadrotor using the proposed controllers. The quadrotor was subjected to a disturbance at about 7 seconds, which resulted in a yaw increase. Controllers were able to direct the quadrotor to zero yaw direction, and to keep zero pitch and roll angles during the motion (Figure 3-16). This experiment shows that the estimation and stabilization algorithms are successful.

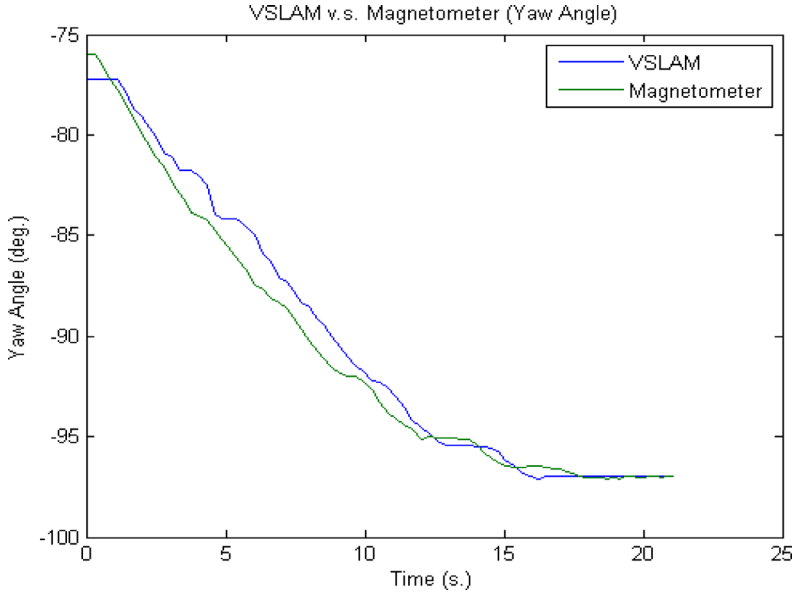


Figure 3-15 Yaw estimation with VSLAM and magnetometer resulted similar results. Source: [25] IEEE, 2012 (with permission).

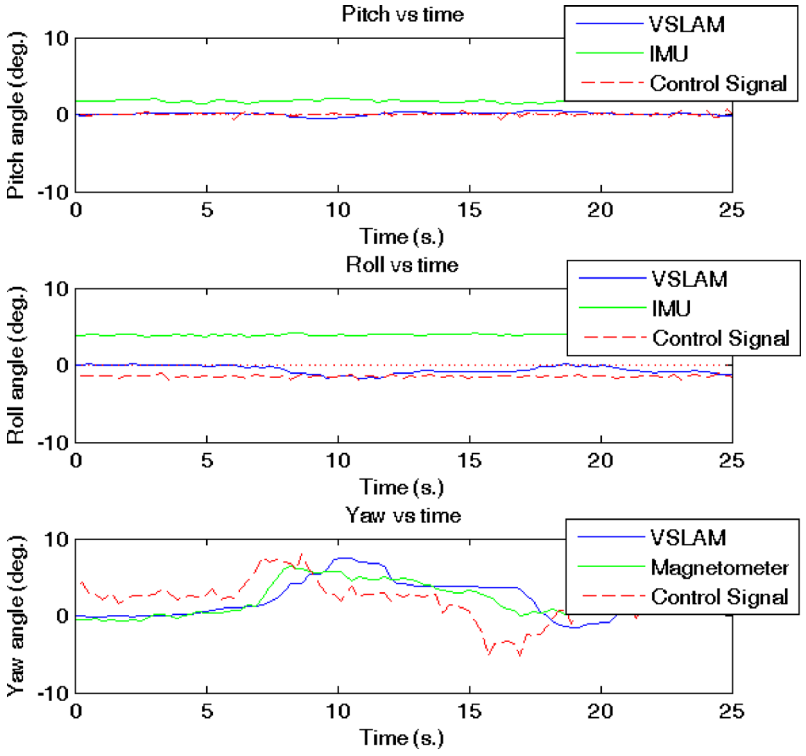


Figure 3-16 Attitude stabilization performed using VSLAM algorithm. Source: [25] IEEE, 2012 (with permission).

3.6 Closure

Attitude is one of the most important parameters for a UAV during flight. In this paper, attitude is estimated by an algorithm that uses a forward looking pinhole camera onboard the UAV. The algorithm uses readily available objects in the field of view, eliminating the need to use artificial or natural features such as parallel lines. This makes the algorithm applicable to different scenes. The proposed system can also be implemented on mobile robots. This algorithm has been tested in various different conditions. In this paper, a quadrotor UAV model and controller have also been introduced. Finally, we presented experiments and the results of the estimation and control algorithms on a real model helicopter. Two experiments were presented, where the proposed VSLAM attitude estimation algorithm gives close results to IMU readings. Furthermore, the controllers stabilize the helicopter rapidly.

The contributions of this chapter are:

- Implementation of a VSLAM code developed for the vision community for the UAV attitude estimation problem.
- Presented quadrotor modeling and control.
- Developed a COTS quadrotor system that has a novel onboard control computer as well as a vision computer.
- Showed successful attitude estimation and attitude stabilization experiments as the quadrotor was moving in laboratory experiments.

In experiments, the estimation and control rate of the system was about 10 fps; this is believed to be improvable. This frame rate and delay did not cause many problems in near hover stabilization experiments, but is expected to cause problems in autonomous flights (especially the delay related to image transfer to the Beagle-board, which will be minimized by optimizing the code). Our future work will also be implementing the algorithm to fully autonomous flight.

There is a trend to increase the level of autonomy of robotic systems, to make them more autonomous, more intelligent. One way to achieve this is to use visual information more intelligently. As this study showed, visual information can be used to determine the current attitude and relative motion of the robot through the tracking of the natural features on the scene. This capability can be added to other robotic systems easily. One limiting factor of this integration is the computing power available for smaller robotic systems. As the newer, more powerful yet less power-consuming micro-computers become available, the integration of the visual sensors for intelligent decision-making and navigation for robotics systems will be possible.

3.7 Acknowledgment

We would like to thank the scientific research projects coordination unit of *Istanbul Technical University* for partially funding this project. We would also like to thank Mr. Gökçe Tağlıoğlu for his help.

3.8 References

[1] Davison, A. J., Reid, I. D., Molton, N. D., Stasse, O., 2007, MonoSLAM: Real-Time Single Camera SLAM, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **29**(6).

[2] Castillo, P., Lozano, R., Dzul, A. E., 2005, *Modelling and Control of Mini-Flying Machines*, Springer-Verlag, London.

[3] Demonceaux, C., Vasseur, P., Pégard, C., 2006, Robust attitude estimation with catadioptric vision, *Proceedings of the International Conference on Intelligent Robots and Systems*, pp. 3448–3453.

[4] Demonceaux, C., Vasseur, P., Pégard, C., 2006, Omnidirectional vision on UAV for attitude computation, *Proceedings IEEE International Conference on Robotics and Automation*, pp. 2842–2847.

[5] Bazin, J. C., Kweon, I. S., Demonceaux, C., Vasseur, P., 2007, UAV attitude estimation by combining horizon-based and homography-based approaches for catadioptric image, *6th IFAC/EURON Intelligent Autonomous Vehicles*, Toulouse, France.

[6] Demonceaux, C., Vasseur, P., Pégard, C., 2007, UAV attitude computation by omnidirectional vision in urban environment, *IEEE Int. Conf. Robot. Autom.*, doi:10.1109/ROBOT.2007.363618.

[7] Bazin, J. C., Kweon, I. S., Demonceaux, C., Vasseur, P., 2008, UAV attitude estimation by vanishing points in catadioptric images, *IEEE International Conference on Robotics and Automation*, pp. 2743–2749.

[8] Tarhan, M., Altuğ, E., 2010, EKF Based Attitude Estimation and Stabilization of a Quadrotor UAV Using Vanishing Points in Catadioptric Images, *Journal of Intelligent and Robotic Systems*, **62**(3–4), pp. 587–607, doi: 10.1007/s10846-010-9459-y.

[9] Blosch, M., Weiss, S., Scaramuzza, D., Siegwart, R., 2010, Vision Based MAV Navigation in Unknown and Unstructured Environments, *IEEE International Conference on Robotics and Automation*, pp. 21–28.

[10] Klein, G., Murray, D., 2007, Parallel tracking and mapping for small ar workspaces, *Proc. 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 225–234.

[11] Achtelik, M., Achtelik, M., Weiss, S., Siegwart, R., 2011, Onboard IMU and Monocular Vision Based Control for MAVs

in Unknown In- and Outdoor Environments, *Proc. of the IEEE International Conference on Robotics and Automation*.

[12] Eberli, D., Scaramuzza, D., Weiss, S., Siegwart, R., 2011, Vision based Position Control for MAVs using one single Artificial Landmark, *Journal of Intelligent and Robotic Systems*, **61**, 495–512.

[13] Achtelik, M., Bachrach, A., He, R., Prentice, S., Roy, N., 2009, Stereo vision and laser odometry for autonomous helicopters in GPS-denied indoor environments, *Unmanned Systems Technology XI*, edited by Grant R. Gerhart, Douglas W. Gage, Charles M. Shoemaker, *Proc. of SPIE* Vol. 7332, 733219.

[14] Herisse, B., Russotto, F. X., Hamel, T., Mahony, R., 2008, Hovering fight and vertical landing control of a VTOL Unmanned Aerial Vehicle using Optical Flow, *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

[15] Janabi-Shari, F., Marey, M., 2010, A Kalman-Filter-Based Method for Pose Estimation in Visual Servoing, *Transactions on Robotics*, **26**(5), pp. 937–947.

[16] Ahrens, S., Levine, D., Andrews, G., How, J. P., 2009, Vision-Based Guidance and Control of a Hovering Vehicle in Unknown, GPS-denied Environments, *IEEE International Conference on Robotics and Automation*.

[17] Eckert, J., German, R., Dressler, F., 2011, An Indoor Localization Framework for Four-Rotor Flying Robots Using Low-Power Sensor Nodes, *IEEE Transactions on Instrumentation and Measurement*, **60**(2), pp. 336–344.

[18] Rudol, P., Wzorek, M. Doherty, P., 2010, Vision-based Pose Estimation for Autonomous Indoor Navigation of Micro-scale Unmanned Aircraft Systems, *IEEE International Conference on Robotics and Automation*.

[19] Jones, E. S., Soatto, S., 2011, Visual-inertial navigation, mapping and localization: A scalable real-time causal approach, *the International Journal of Robotics Research*, published online 17 January 2011.

[20] Kelly, J., Sukhatme, G. S., 2010, Visual-Inertial Sensor Fusion: Localization, Mapping and Sensor-to-Sensor Self-calibration, *the International Journal of Robotics Research*, published online 5 November 2010.

[21] Rondon, E., Salazar, S., Escareno, J., Lozano, R., 2010, Vision-based Position Control of a Two-rotor VTOL miniUAV, *J Intell Robot Syst*. **57**, pp. 49–64.

[22] Cheviron, T., Hamel, T., Mahony, R., Baldwin, G., 2007, Robust Nonlinear Fusion of Inertial and Visual Data for position, velocity and attitude estimation of UAV, *IEEE International Conference on Robotics and Automation*, pp. 10–14.

[23] Johnson, N. G., 2008, Vision-Assisted Control of a Hovering Air Vehicle in an Indoor Setting, Thesis, Brigham Young University.

[24] Beagle-board. [Online] Available: http://beagleboard.org/.

[25] Yiğit, B., Altuğ, E., 2012, Visual attitude stabilization of a unmanned helicopter in unknown environments with an embedded single-board computer, *IEEE International Symposium on Robotic and Sensors Environments*, pp. 49–54.

[26] Bouabdallah, S., Siegwart, R., 2007, Design and Control of a Miniature, *Advances In Unmanned Aerial Vehicles*, Springer, pp. 171–210.

[27] Altuğ, E., Ostrowski, J., Taylor, C. J., 2005, Control of a Quadrotor Helicopter Using Dual Camera Visual Feedback, *the International Journal of Robotics Research*, **24**(5), pp. 329–341.

4. Robot swarms: dynamics and control

Veysel Gazi, University, Mahmutbey, Turkey; Barış Fidan, University of Waterloo, Canada; Lino Marques, University of Coimbra, Portugal; and Raul Ordonez, University of Dayton, USA

Abstract: This chapter provides a brief overview of modeling and control of multi-robot systems or simply robot swarms, from a dynamics and control perspective. First some commonly used agent dynamic models are described and some swarm coordination and control problems are presented. Then, commonly used control design approaches are summarized and some related swarm robotic applications are discussed. The chapter provides an extensive list of references to help the readers find detailed information on most recent key studies in the field.

4.1 Introduction

The field of swarms and swarm robotics, and the more general field of multi-agent dynamic systems, is an active research field which has been popular for more than two decades now. Since the pioneering work of Reynolds [1] on simulation of a flock of birds in flight, extensive work has been done on modeling and analysis of the swarm behavior, as well as development of swarm control and coordination algorithms. A collection of some of the related results can be found in the recent special journal issues [2, 3], edited books [4–8] and monographs [9–14] on swarm robotics and multi-agent dynamic systems.

A mobile *robot swarm*, in general, can be defined as a network of mobile robots moving on a (two-dimensional) plane or a three-dimensional space to perform certain cooperative tasks. Within this context each individual robot in the swarm is considered a dynamic *agent*. There is no direct mechanical link between pairs of robot agents within the swarm, but rather some wireless sensing and communication links between certain assigned pairs. The main purpose of using mobile robot swarms is to collectively reach goals that are difficult to achieve by an individual robot or a monolithic robot system. A sample experimental robot swarm moving on plane is shown in Figure 4-1.

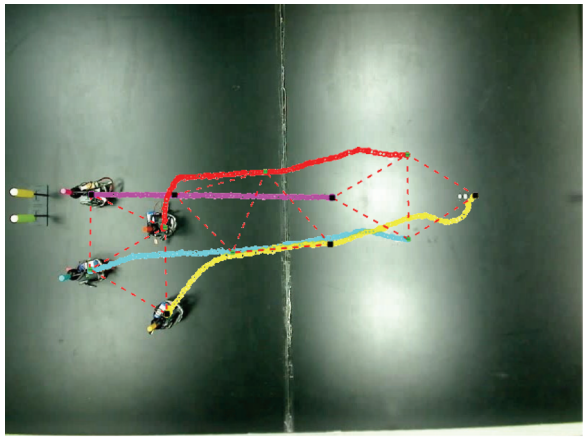


Figure 4-1 A robot swarm with four agents performing a rigid formation control experiment [15].

The swarm robotics field is interdisciplinary by nature, and there are many aspects with various modeling and analysis approaches [16]. Many studies on swarms get their inspiration from nature, and concepts like stigmergy [14], bio-mimicry [11, 14], and physico-mimicry [8] are getting increasingly popular. In this chapter, we focus on system dynamics and control aspects, and review studies on a set of coordination and control problems involving robot swarms.

The increased interest in multi-agent dynamic systems, and in particular in swarms of robots, is due to the rich portfolio of possible applications in various fields, including agriculture, health, defense and others. Of particular current interest are applications involving teams of unmanned aerial, ground, space or underwater vehicles, robots, mobile sensors, automatic life-stock control units, etc. [17–26]. From mathematical and system theoretical perspectives, these applications map to the problems of aggregation and foraging [27–29], formation control and coordinated tracking [30–32], distributed agreement and output synchronization [33–46], and source seeking [47–55]. We provide further details of these problems in Section 4.3. Note that this is not an exhaustive list and in addition to the aforementioned problems, there exist also various more specific ones considered in robot swarm literature, including coordinated search, deployment, map building, and olfactory navigation.

For the robot swarm coordination and control problems mentioned above, various multi-agent control approaches have been developed in the literature, including behavior-based methods [56], feedback linearization [57], virtual structures [58], leader-follower structure [15, 59–63], artificial potential functions [64], graph theory-based [15, 61, 65–68] methods, and extremum seeking control [55]. The details of these approaches are given in Section 4.4.

The chapter is organized as follows: Section 4.2 presents the agent (individual robot) models that are most widely used in the literature. Various types of robot swarm coordination and control problem definitions are covered in Section 4.3. Section 4.4 presents various approaches in the literature to these problems. A summary and concluding remarks are provided in Section 4.6.

4.2 Agent dynamics

For a robot swarm coordination and control problem one needs to consider high-level dynamic behavior of the whole swarm, including interactions between individual robot agents in the swarm, as well as the low-level dynamic characteristics of the agents. There are various mathematical models which can be used to describe the dynamics of the individual agents. Here we consider a classification based on actuation constraints, and present holonomic (fully actuated) and non-holonomic (involving velocity constraints) agent models in Subsections 4.2.1 and 4.2.2, respectively. Later, in Subsection 4.2.3, we discuss the introduction of simplified models for convenience of control design.

4.2.1 Fully actuated agent model

A general form for a fully-actuated (holonomic) agent dynamics model can be represented as:

$$M(p_i)\ddot{p}_i + f_i(p_i, \dot{p}_i) = u_i, \tag{1}$$

where $p_i, \dot{p}_i, \ddot{p}_i, u_i \in \mathrm{R}^n$, and $M(p_i) \in \mathrm{R}^{n\times n}$ denote the position, velocity, acceleration, control input, and mass (inertia) matrix of agent/robot i, respectively. The term $f_i(p_i, \dot{p}_i)$ represents the other effects (such as centripetal, Coriolis, gravitational effects, and additive disturbances). In case it is completely unknown, f_i can be thought of as the cumulative disturbance acting on the agent dynamics.

The fully-actuated model in (1) can be used to represent some omni-directional robots as well as some manipulators or spacecraft [69–72]. If f_i and M_i are known, with a control input of the form:

$$u_i = f_i(p_i, \dot{p}_i) + M(p_i)\overline{u}_i$$

the model in (1) can be easily reduced to the point mass model:

$$\ddot{p}_i = \overline{u}_i \tag{2}$$

Therefore, researchers usually consider either (2) or assume that the agent dynamics in (1) contain uncertainties and disturbances. One usual assumption is that:

$$f_i(p_i, \dot{p}_i) = f_i^k(p_i, \dot{p}_i) + f_i^u(p_i, \dot{p}_i), 1 \le i \le N,$$

where $f_i^k(\cdot,\cdot)$ represents the *known* part and $f_i^u(\cdot,\cdot)$ represents the *unknown* part. The unknown part is assumed to be bounded with a known bound, i.e.,

$$\left\| f_i^u(p_i, \dot{p}_i) \right\| \le \overline{f}_i(p_i, \dot{p}_i), 1 \le i \le N,$$

where $\overline{f}_i(p_i, \dot{p}_i)$ are known for all i. This in a sense incorporates model uncertainties and additive disturbances in the model. Another usual assumption is that the mass/inertia matrices $M_i(p_i)$ are unknown for all agents/robots i. However, it is also assumed that they are lower and upper bounded by known bounds, e.g., they satisfy:

$$\underline{M}_i \|y\|^2 \le y^{\mathrm{T}} M_i(p_i) y \le \overline{M}_i \|y\|^2, \; 1 \le i \le N$$

for any arbitrary $y \in \mathrm{R}^n$ and some known scalars $\underline{M}_i$ and $\overline{M}_i$ satisfying $0 < \underline{M}_i < \overline{M}_i < \infty$. Consideration of uncertainties and disturbances affecting the agent dynamics makes the agent model more realistic. To suppress such adverse effects one can use either robust or adaptive strategies. Sample robot swarm studies considering the fully actuated dynamics in (1) can be seen in [31, 32, 73, 74].

4.2.2 Non-holonomic agent dynamics

Another agent/robot model that is commonly used in the literature is given by:

$$\begin{aligned}
\dot{x}_i &= v_i \cos(\theta_i), \\
\dot{y}_i &= v_i \sin(\theta_i), \\
\dot{\theta}_i &= \omega_i, \\
\dot{v}_i &= \frac{1}{m_i}[F_i + f_{v_i}], \\
\dot{\omega}_i &= \frac{1}{I_i}[\tau_i + f_{\omega_i}]
\end{aligned} \tag{3}$$

where $p_i(t) = [x_i(t), y_i(t)]^T \in R^2$ denotes the position of agent i at time instant t in cartesian coordinates, θ_i is its steering angle, v_i is its linear speed, and ω_i is its angular speed. These terms are graphically shown in Figure 4-2. The quantities m_i and I_i are positive constants and represent the mass and the moment of inertia of agent i, respectively. The control input for agent i is $u_i = [F_i, \tau_i]^T$ where F_i is the force input and τ_i is the torque input.

The terms f_{vi} and $f_{\omega i}$ in (3) represent modeling uncertainties and additive disturbances for agent i. These terms can be assumed to be known or unknown. In case they are known they can easily be compensated for

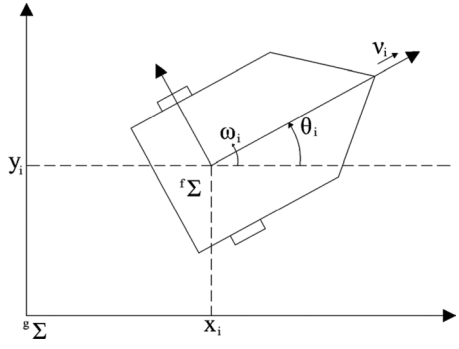


Figure 4-2 Illustration of the dynamics of a non-holonomic agent/robot.

with appropriate controller design. Therefore, in order to obtain more general and more practical results, they are usually assumed to be unknown but with known bounds. In other words, usually it is assumed that $|f_{vi}| < f_v^+$ and $f_{\omega i}| < f_\omega^+$, for known bounds f_v^+ and f_ω^+. Another common assumption is that the exact values of the mass m_i and the inertia I_i for agent i are unknown. Still, as in the case for the inertia matrix $M_i(p_i)$ for the fully actuated model in (1), it is assumed that they are lower and upper bounded in the form $0 < \underline{M} < m_i < \overline{M}$ and $0 < \underline{I} < I_i < \overline{I}$, where the bounds $\underline{M}$, $\overline{M}$, $\underline{I}$, and $\overline{I}$ are known.

The model (3) can be used to represent the dynamics of differentially driven mobile robots available in many research laboratories throughout the world. It can also be used to represent the dynamics of a unicycle and therefore sometimes is called the *unicycle model*. If the robots are moving with low speed, the acceleration dynamics can be removed and the model can be simplified to a *kinematic* agent model to include only the first three equations:

$$\begin{aligned} \dot{x}_i &= v_i \cos(\theta_i), \\ \dot{y}_i &= v_i \sin(\theta_i), \\ \dot{\theta}_i &= \omega_i, \end{aligned} \tag{4}$$

with the linear speed v_i and the angular speed ω_i as the control inputs, i.e., $u_i = [v_i, \omega_i]^T$. The model (4) is widely used in the literature as well.

The agent dynamics in (3 and 4) have a non-holonomic velocity constraint. In particular, the agents cannot move along the axis connecting the two actuated wheels in Figure 4-2. This constraint can be expressed in the form:

$$\dot{x}_i \sin(\theta_i) - \dot{y}_i \cos(\theta_i) = 0,$$

and needs to be taken into account in controller design. In fact, compared to controller development for the model (1), controller development for the model (3) is in general more difficult. It is known that systems such as (3) "cannot be stabilized via continuously differentiable, time invariant, state feedback control laws" [75, 76]. This brings an important restriction leading to the fact that controllers stabilizing (3) have to be either discontinuous or time varying.

Depending on the available agent/robot dynamics, one can usually use either the fully actuated holonomic model (1) or the non-holonomic model (3). However, there are exceptional problems where both models can be employed simultaneously. For example, as noted in [77], the formation control problem can be stated as the control of systems having both non-holonomic (vehicle dynamics) and holonomic (formation and tracking) constraints.

Another issue to mention about the non-holonomic model is that since the angles are 2π periodic it is possible to view θ_i as a continuous variable taking values from $(-\infty, +\infty)$, or as a discontinuous variable taking values in an interval of length 2π, such as $[-\pi, \pi)$. The second approach also requires special attention to be taken for operations with angles. In particular, one can perform all addition operations on the angles (mod 2π) with $-\pi$ radians shift. For example, $\theta_1 + \theta_2$ and $\theta_1 - \theta_2$ can be calculated as $[(\theta_1 + \theta_2 + \pi)(\text{mod } 2\pi) - \pi]$ and $[(\theta_1 - \theta_2 + \pi)(\text{mod } 2\pi) - \pi]$, respectively. Similarly, $\theta_i(t)$ can be defined as:

$$\dot{\theta}_i(t) = \lim_{\Delta t \to 0} \frac{(\theta_{\varepsilon,}(t) - \theta_i(t - \Delta t) + \pi)(\text{mod } 2\pi) - \pi}{\Delta t}$$

Then, the corresponding control laws can be modified accordingly. It is possible to use various strategies in order to steer the robotic agents in a desired direction [78].

4.2.3 Simplified or high-level agent models

The agent models discussed in preceding subsections are realistic models representing physical dynamics of agents/robots. They can be viewed as low-level agent dynamics models. In order to focus on overall behavior of the swarm and simplify the analysis, some studies further simplify or ignore the individual agent's dynamics. In other words, there are studies using high-level models for describing the motion of the agents, and they do not consider the low-level agent dynamics. Such models can be thought of as tools for planning agent paths and generating way-points for the agents to visit in order to achieve the desired behavior. Usually it is easier to analyze the qualitative overall behavior of the swarm using such simplified models. However, for implementation on real robotic swarms one still needs to consider the low-level agent dynamics based on robots used for experimentation.

One possible simplification is the use of a single integrator model instead of the double integrator model in (2). In other words, the agents can be assumed to move based on:

$$\dot{p}_i = \overline{u}_i \tag{5}$$

and perform the analysis accordingly. Then the results obtained for the swarms composed of agents with dynamic obeying (5) can easily be adapted to swarms composed of agents with dynamic obeying (2) with inclusion of appropriate damping to the developed control inputs, and without change in the overall qualitative behavior.

Other high-level or simplified models commonly used in the literature are discrete time models. Probabilistic Markovian models, evolutionary models, optimization-based deterministic or nature-inspired models are examples of such high-level discrete time representations [16]. Then, given the model, the overall swarm dynamics depend on the properties of the underlying motion coordination algorithm and relevant tools appropriate for the employed model/algorithm can be used to analyze the overall qualitative behavior.

4.3 Problem definitions

The control theoretical studies on swarms are concerned with developing appropriate individual control laws u_i for the agents/robots which will lead to achieving a desired swarm behavior for the given agent/robot and swarm settings. Various behaviors have been studied in the literature. We discuss some of these behaviors in more detail below. More detailed investigations can be found in [10, 16]. Note that the problems presented here provide just a glimpse of the enormous potential research issues and agent behaviors, and the potential applications of swarms.

4.3.1 Aggregation and social foraging

Cooperative collective behavior usually occurs in aggregated swarms in nature. Therefore, *aggregation* is an important fundamental behavior for swarms in nature and for swarm robotic systems. In swarms in nature, aggregation usually also occurs during *social foraging*, which is a behavior with many advantages, such as increasing probability of success for the individuals [79, 80]. The aggregation and social foraging behaviors have also been studied in engineering literature, using various approaches.

Given a swarm of N agents with positions $p_i(t)$, the objective in the swarm aggregation problem is to design the control inputs $u_i(t)$ for every agent i such that:

$$\lim_{t\to\infty}\left\|p_i(t)-p_j(t)\right\|\le\varepsilon,\ \ \forall i,j\in\{1,\ldots,N\},i\neq j \tag{6}$$

for some $\varepsilon > 0$ [10] (see Figure 4-3(a) for illustration). A commonly used approach to solve this problem is potential function modeling [27, 29]. The studies in [27, 29] rigorously investigate the dynamics of an artificial potential function-based model of swarm aggregation, and derive bounds on the swarm size for swarms composed of agents with single integrator dynamics. Double integrator agents are considered in [81, 82], with similar results to those in [27, 29] about the qualitative properties of the swarm dynamics. The article [31] considers swarms

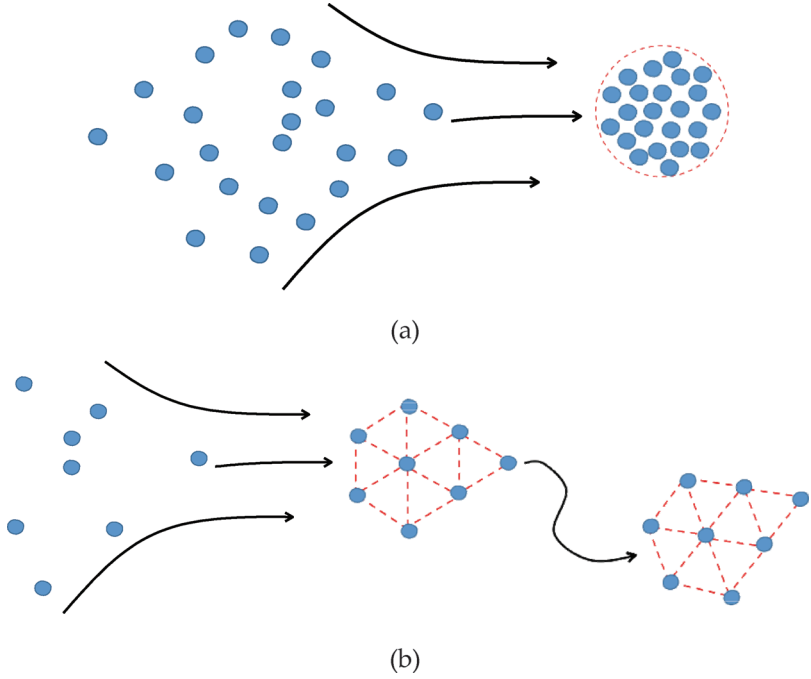


Figure 4-3 Illustration of typical robot swarm coordination and control problems: (a) Aggregation. (b) Formation acquisition and maintenance.

composed of agents with dynamics obeying the fully actuated model (1) with uncertainties, and develops a sliding mode controller to achieve swarm aggregations and suppress the uncertainty effects.

Social foraging is similar to aggregation in the sense that usually aggregation also occurs during social foraging. However, it is different since in social foraging the environment affects the motion dynamics of the individuals. The objective in social foraging is to increase the energy intake by the individuals and the swarm. Therefore, in nature the swarm moves towards regions with higher food/nutrient concentration (which can be referred to as favorable regions) and away from regions containing toxic or hazardous substances (which can be referred to as unfavorable regions). These concepts can be easily extended to swarm robotic systems such that favorable regions represent targets or goals, whereas unfavorable regions represent threats or obstacles.

In the light of the above observations, given a swarm of N agents with positions $p_i(t)$ and a function σ: $\mathrm{R}^n \to \mathrm{R}$ representing the environment (usually called the resource profile), the objective in the social foraging problem is to design the control inputs $u_i(t)$ for every agent i such that:

$$\lim_{t\to\infty} \left\| p_i(t) - c_{\sigma_j} \right\| \leq \varepsilon_j, \forall i \in \{1,\ldots,N\} \tag{7}$$

for some $\varepsilon_j > 0$ and $c_{\sigma_j} \in \mathrm{R}^n$ such that $\sigma(c_{\sigma_j}) \leq \sigma(x)$ for all x in some neighborhood of c_{σ_j}. In other words, defining the resource profile function σ such that the regions with lower σ values are more favorable than those with larger σ values, the swarm is required to converge into the vicinity of a local minimizer of σ. Note that satisfaction of (7) by all agents in the swarm results in satisfaction of (6) as well.

Social foraging for robot swarms using potential functions has been studied in [28, 83] using single integrator and double integrator dynamics with noise. Similarly, the works [84, 85] study aggregation as well as social foraging using double integrator dynamics. The article [85] considers an energy-based Lagrangian approach for investigating swarm dynamics. It brings an alternative strategy for analyzing the dynamics of swarms, and unifies the analysis strategies used for biological swarms and engineering multi-agent dynamic systems. The article

[86] considers swarms composed of non-holonomic agent dynamics described by (3), and achieves similar qualitative results.

Inspired by the social foraging of different species, researchers have developed various optimization algorithms. Examples of such algorithms include the *ant colony optimization* method [14], the *particle swarm optimization* method [10, 87, 88], and the *bacterial foraging* optimization method [10, 11, 89]. As mentioned in the preceding section, such algorithms can also be effectively used for high-level path planning for robot swarms, for such applications as cooperative search, contaminating source localization, chemical concentration mapping [90, 91], and others. In addition, they are powerful, general-purpose optimization methods.

Probabilistic [92] and evolutionary [93] models and studies for swarm aggregations can also be found in the literature. In particular, probabilistic strategies can be combined with rule based approaches, with each rule being applied with some probability to achieve aggregation. Evolutionary strategies, on the other hand, can effectively be used to tune controllers with unknown parameters (such as neural network controllers) for achieving desired swarm behavior, such as aggregation.

4.3.2 Formation control and swarm tracking

The formation of geometric patterns by robot swarms is an important engineering problem. This behavior can be seen in swarms in nature during cooperative behaviors such as migration, object transportation and others. The formation control problem can be divided into various stages, such as formation acquisition, formation maintenance and formation reconfiguration (see Figure 4-3(b) for illustration). These can be viewed also as different behaviors exhibited by the swarm robotic systems. Formation acquisition is concerned with achieving a predefined geometric shape from any initial position and orientation of the swarm members. Similarly, formation maintenance is concerned with keeping the acquired formation during motion of the swarm. Formation reconfiguration, on the other hand, is concerned with various behaviors, such as splitting, joining formations or changing the geometric shape of the formation. Such behaviors or tasks might need to be performed by a swarm robotic system while performing higher level missions or goals.

To express the formation control problem more formally consider a swarm of N agents with positions $p_i(t)$, the specific agent motion dynamics, and control inputs $u_i(t)$. Then the problem of formation control concerns designing the control inputs $u_i(t)$ for every agent i such that:

$$\lim_{t\to\infty}\left\|p_i(t)-p_j(t)\right\| = d_{ij,}\forall i \neq j \in \{1,\ldots N\} \tag{8}$$

for a given set of desired inter-agent distances $\{d_{ij}|i, j \in \{1,\ldots,N\}, i \neq j\}$, where d_{ij} denotes the desired distance between agents i and j. Sometimes achieving/satisfying the condition (8) exactly might be very difficult, and small tolerances might be allowable. In such cases, the condition can be stated as:

$$\lim_{t\to\infty}\left|\left\|p_i(t)-p_j(t)\right\|-d_{ij}\right| \leq \varepsilon, \forall i \neq j \in \{1,\ldots,N\}, \tag{9}$$

for some $\varepsilon > 0$.

The above problem definitions involve only the agents, and there are no external influences affecting the motion of the swarm. External environmental effects similar to those in the foraging problem can also be included in the definition of the model. Sometimes an external effect of particular interest is the case of external agents (agents not belonging to the swarm) which can represent friendly or hostile entities. Pursuit-evasion by the swarm and unwanted intruders under various scenarios are interesting research directions. Here we would like to emphasize the problem that we refer to as the swarm tracking problem, in which a swarm of agents is required to capture/enclose a moving target, achieve a geometric formation around the target, and continue tracking it (in a sense escort it) while keeping the formation. It is related to the formation control problem in the sense that the condition (8 or 9) needs to be satisfied. However, in addition to (8 or 9), the agent controllers are required to ensure that:

$$\lim_{t\to\infty} = p_T(t) \in \text{conv}\{p_1(t),\ldots,p_N(t)\}, \tag{10}$$

is satisfied. Here $p_T(t)$ denotes the position of the mobile target and conv$\{p_1, \ldots, p_N\}$ denotes the convex hull of $p_1, \ldots, p_N$. Note that the

swarm tracking behavior can be thought of as being composed of two parallel subtasks – formation acquisition/maintenance and cooperative enclosing/tracking the maneuvering target.

The formation control and/or swarm tracking problems have been investigated in many works in literature including [32, 74, 85, 86, 94, 95]. The work in [85] considers double integrator dynamics and uses an energy approach. The articles [32, 74] use the fully actuated model, whereas the works in [86, 94, 95] use the non-holonomic agent model for individual agent dynamics. Other articles on target tracking and/or formation control of swarms of agents include [17, 57, 64, 96–99].

4.3.3 Source seeking

Source seeking, or source localization, i.e., the design of control algorithms for autonomous agents to seek a source with unknown spatial distribution, is of great interest, where techniques such as extremum seeking control [55] have been used in the design. Source seeking is an application which has great theoretical interest, and it also has a significant impact on engineering applications: for instance, developing vehicles with more autonomy, such as the situation where no GPS information is available, or reducing costs due to position sensors. Some of the direct applications of source seeking can be found in contaminant plume control, autonomous odor sensing or toxic gas leakage localization. Source seeking for fish models is discussed in [100, 101] and an overview of source seeking can be found in [102]. Sliding mode extremum seeking control for non-holonomic vehicle source seeking is investigated in [103, 104].

In the application of source seeking, the task of the vehicle is to find a source that emits a signal that decays as a function of distance away from the source, where the signal field is unknown and only the measurement of the signal at the current agent location is available.

A basic diagram of source seeking using extremum seeking control is provided in Figure 4-4, where the control goal is for the agent to seek an extremum of an unknown signal field based on the measurement of the signal only. Source seeking is first discussed for systems with moderately unstable single poles in [47], where the autonomous vehicle is modeled as a single or a double integrator, such as in (2). A formal study on source seeking for the unicycle model (4) first appeared in [48], where

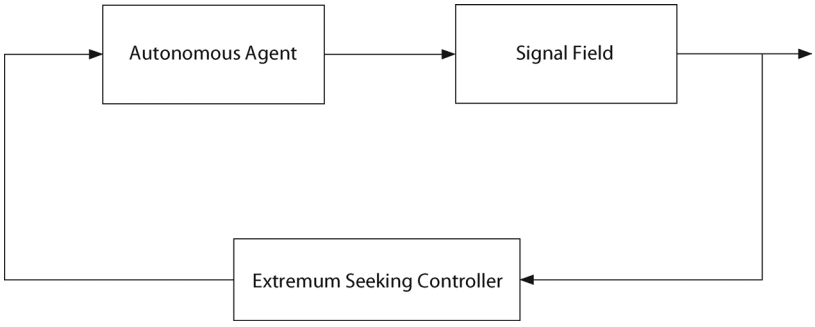


Figure 4-4 Source seeking [55].

the design keeps the angular velocity constant and tunes the forward speed by extremum seeking, which generates triangle or star pattern vehicle motions that drift towards the source.

A more challenging analysis is performed with a different strategy in [49], where the approach is to keep the forward speed constant and tune the angular velocity by extremum seeking. The resulting motion sinusoidally converges to the source and settles in a ring around the source as the forward speed is constant. Based on these two strategies, adding a simple derivative-like feedback to the forward speed in the angular velocity tuned by the extremum seeking loop [52] allows the vehicle to slow down as it gets closer and converges closer to the source, without giving up convergence speed. Source seeking for slow or drifting sensors is explored in [53]. Rather than sinusoidal perturbations, a stochastic source seeking control law is used to tune the forward velocity in [105, 106] and the angular velocity in [107].

When source seeking is performed using a group of robots, the problem may become mathematically equivalent to swarm tracking, as described in Section 4.3.2. The main difference is conceptual, in that when performing swarm tracking together with, say, artificial potentials, the entire potential is typically assumed known. In contrast, in group source seeking (or swarm seeking), at least part of the potential function is assumed unknown, and is maximized or minimized collectively by the swarm. In this case, the maximization or minimization results in the localization of the unknown source, while also, for instance, attaining and maintaining a desired formation. Various strategies based on extremum seeking control have been explored in [55].

4.4 Control design approaches

There are various approaches for high-level control, or path planning, and low-level agent control in multi-agent dynamic systems. It is possible to design high-level path planning or high-level control strategies without considering the individual agent dynamics, although taking them into account can also be beneficial. In contrast, low-level control of individual agents by its nature deals with the agent dynamics. Since the interest in multi-agent dynamic systems is in the collective behavior of the agents, the inter-agent distances play an important role and are of particular interest. In the preceding section, the swarm control problems are defined in terms of constraints on the inter-agent or agent-extremum point distances. Therefore, the control strategies developed need to take into account the distance requirements on the agents.

4.4.1 Artificial potential functions

Artificial potential functions are commonly used in multi-agent dynamic systems in order to specify inter-agent interactions and interactions of the agents with their environment. They can be used to construct attractive and repulsive forces between agents and between the agents and the environment [17, 31, 32, 86, 94, 99, 108, 109]. Artificial potentials can be thought of as the functions representing the forces on the agents due to the potential energy arising, due to the positions of the agents relative to each other, or relative to targets or threats in the environment [85]. In particular, these forces can be thought of as acting so as to decrease the overall potential energy, and thereby move the swarm towards a configuration with lower potential energy. Depending on the function that they perform, the potential functions can be categorized in different ways, including swarm potential, environmental potential, target or threat potential, and others [85]. Such potentials can be used to represent interactions in both biological swarms and swarm robotic systems. While the resultant behavior in biological systems needs to be biologically realistic, no such requirement exists in engineering multi-agent dynamic systems. Instead, engineering swarms should safely satisfy other requirements/constraints, such as achieving the system objectives or mission goals, including the problems defined in the preceding section.

The potential functions constitute, in a sense, a high-level control or path planning approach. Therefore, once the interaction requirements

are specified by the potential functions and given the agent dynamics, low-level control algorithms need to be designed in order to force the agents to move based on the potential function requirements.

Aggregation and social foraging

The aggregation potential is usually composed of two terms – one for attraction and the other for repulsion between individuals. The attraction term dominates on long distances in order to keep the individuals together, whereas the repulsion term dominates on short distances in order to avoid collision when agents are in close proximity. The potential energy due to the aggregation potential can be represented in the form:

$$J_{agg}(p) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left[J_a\left(\left\|p_i - p_j\right\|\right) - J_i\left(\left\|p_i - p_j\right\|\right)\right] \tag{11}$$

where J_a:$R^+ \rightarrow$ R represents the *attraction component*, and whereas J_r:$R^+ \rightarrow$ R represents the *repulsion component*. Various potentials can serve as aggregation potentials. One particular potential is the so-called *Morse potential*:

$$J_{agg}(p) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left[-a\exp\left(-\frac{\left\|p_i - p_j\right\|}{c_a} \right) + b\exp\left(-\frac{\left\|p_i - p_j\right\|}{c_r} \right) \right] \tag{12}$$

where a, b, c_a, $c_r > 0$, $b > a$, $c_a > c_r$, and $\left(\frac{b}{a}\right)\left(\frac{c_r}{c_a}\right)^2 < 1$ are satisfied. There are many other possible potentials that can lead to aggregation behavior [27–29, 85, 86, 109–114].

In the case of social foraging, in addition to the interactions between individuals, interaction of the individuals with the environment needs to be represented as well. This can be done by introducing a resource profile to model the environment, or simply an environmental potential J_{env}:$R^n \rightarrow$ R, which represents the potential energy due to the position of the agents in the resource profile. The resources in the resource profile include goals and targets that are attractive to the agents and obstacles, and threats that are repulsive to the agents. An example of a resource profile is the multimodal Gaussian resource profile:

$$J_{env}(y) = -\sum_{i=1}^{M} \frac{A_{ei}}{2} \exp\left(-\frac{\|y - c_{ei}\|^2}{l_{ei}} \right) + b_e$$

where $c_{ei} \in \mathrm{R}^n$, $l_{ei} \in \mathrm{R}^+$, $A_{ei} \in \mathrm{R}$, $\forall\ i = 1, \ldots, M$, and $b_e \in \mathrm{R}$. See Figure 4-5 for an example plot of such a profile. Depending on the properties of the environment, one may use other resource profile functions as well [28, 83, 85, 86]. Note that the potential function $J_{env}(y)$ discussed here correspond to the resource profile function σ mentioned in the preceding section. There could also be alternative non-potential function-based methods for representing the environment.

Formation control and swarm tracking

Potential functions can also be used to specify the inter-agent distances in the formation control problem and to achieve geometric formations. This can easily be achieved by a simple modification of the aggregation potential to have pair-dependent attraction repulsion parameters, depending on the desired distances between the pairs of agents in the

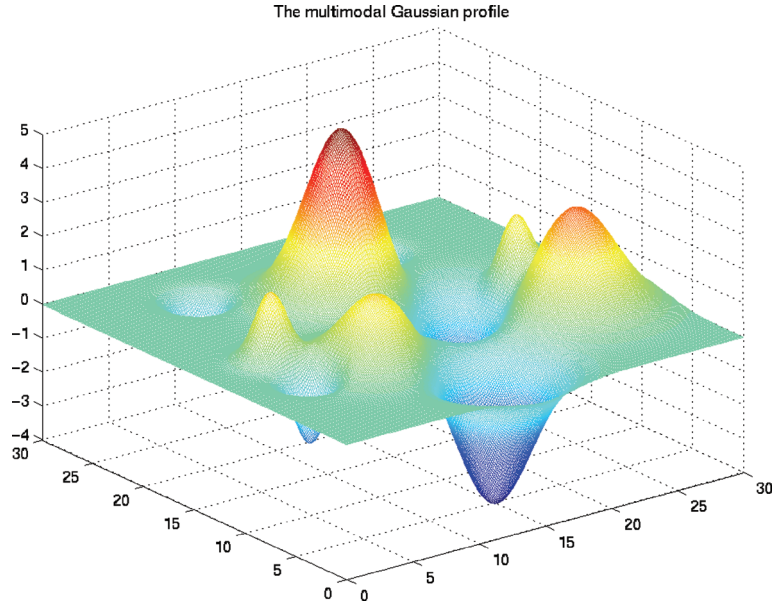


Figure 4-5 Example resource profile (Multimodal Gaussian profile [28]).

geometric formation. For example, the Morse potential (12) can be modified as:

$$J_{frm}(p)=\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\left[-a_{ij}\exp\left(-\frac{\|p_i-p_j\|}{c_{aij}}\right)+b_{ij}\exp\left(-\frac{\|p_i-p_j\|}{c_{rij}}\right)\right]$$

where a_{ij}, b_{ij}, c_{aij}, $c_{rij} > 0$, $b_{ij} > a_{ij}$, $c_{aij} > c_{rij}$, and $\left(\frac{b_{ij}}{a_{ij}}\right)\left(\frac{c_{rij}}{c_{aij}}\right)^2 < 1$ are satisfied. These parameters can have different values for different (i, j) pairs, depending on their relative location in the desired geometric formation.

In the problem of swarm tracking the agents are required to track a moving target, in addition to acquiring and maintaining a geometric formation. Therefore, in the case of swarm tracking, the overall potential consists of two terms – one for formation control and one for target tracking. The target tracking potential can be a simple quadratic potential or any other attractive potential. An example of such a potential is:

$$J_{tar}(p,p_t)=\sum_{i-1}^{N}\frac{1}{4}\|p_i-p_t\|^4 \qquad (13)$$

where p_t denotes the position of the target. Note that in this formulation there is an attraction to every agent from the target. Using a target tracking potential of the form (13), it is possible to show that the swarm will enclose the target as desired. In the meantime the formation potential will guarantee that the agents acquire and keep the desired geometric shape. In case there is a need to evade the target then the attractive potential can be replaced with a repulsive potential.

One issue to mention about potential functions is that they can suffer from the problem of existence of local minima. In other words, the dynamics can get stuck at a local minimum, and convergence to a global minimum cannot be guaranteed. This can be critical in particular in the case of formation control and swarm tracking tasks, since achieving the desired geometric shape cannot be guaranteed globally. In that case, in order to achieve convergence to the desired formation, the agents must start their motion with an initial configuration that is in the attraction

region of the desired configuration. Nevertheless, there are also strategies, such as simulated annealing, which can be used to overcome local minima. There are various studies in the literature considering the formation control or the swarm tracking problems using potential functions [29, 32, 85, 86].

4.4.2 Neighborhood topologies

The neighborhood structure or topology is very important for the information flow in multi-agent dynamic systems [10, 13, 115]. It determines how quickly and how broadly the information is spread within the swarm. Therefore, the topology also has an effect on the stability and performance of the control algorithms. One possible neighborhood strategy is to use all-to-all interactions. However, as the number of agents increases or the operational area of the swarm gets larger, all-to-all interaction becomes impractical, or simply infeasible. Therefore, other structures such as nearest neighbors rules and leader-follower type structures have also been considered in the literature.

Two distinct neighborhoods can be defined for a given multi-agent system: one for sensing or communication between the agents and one for satisfying control or task constraints. In other words, the set of agents from which a given agent obtains information (either via sensing or communication) and the set of agents with which it has to perform a common task (such as keeping a distance) might be different. The interactions in both neighborhoods can be defined as unidirectional or bidirectional. Moreover, it is possible to have static and dynamic (i.e., time-varying) topologies.

A common approach to represent or model the neighborhood structure/topology in multi-agent dynamic systems is to use tools from algebraic graph theory. This approach can be used together with potential functions or with other methods alike. Introducing graphs to model neighborhoods allows usage of techniques from graph theory for analyzing the connectivity properties of the swarm and derivation of conditions for the necessary or sufficient information flow for achieving the desired task.

4.4.3 Gradient-based, Lyapunov, and sliding mode methods

Many approaches using potential functions, and in particular the extremum seeking control strategies, utilize the gradient of the potential function (or the related other functions) in order to minimize (or

maximize) the potential (the objective function). Since the potential functions constitute a high-level method for modeling agent interactions and path planning, it is easier to develop the control/motion strategy assuming that the motion dynamics of the agents are governed by the point-mass single integrator model in (5), where $\bar{u}_i \in \mathrm{R}^n$ is the control input for the *ith* agent. This allows the designer to focus on the higher level task of coordinated control design. Then the obtained results for the single integrator model in (5) can be extended to more realistic models, such as the double integrator point-mass dynamics in (2) and the non-holonomic agent dynamics in (3), using techniques such as phase lead compensator design [47] and sliding mode control [94].

The potential function, which can be viewed as a signal to be tracked by the agents, can have many unknown minima. Its value/strength is assumed to be measured by the agents. Based on the application, this in a sense means that the agents can measure their distance to their neighbors, their distance to a possibly moving target, the value of the environment/resource profile (i.e., their distance to the obstacles/threats and/or targets/goals in the environment), etc. Then the purpose of the design is to minimize the potential and to achieve the desired goal, which can be aggregation, social foraging, source seeking, formation control, collision avoidance, obstacle avoidance, etc.

As mentioned in the preceding sections, depending on the application, the potential function can be composed of several components. For example, in the swarm tracking problem, it can be composed of an inter-agent interaction component and a tracking component. The inter-agent interaction component puts a constraint on the agents, based on their neighbors' positions, in order to maintain a group structure, and includes functions of the relative distance between each pair of neighbors. Its specific form is defined according to the desired geometric formation. The tracking component, on the other hand, contains constraints with respect to the target(s) or the source(s) of the particular tracking problem. It is typically defined in terms of a scalar signal to be tracked in order to direct the group's behavior for target tracking or source seeking. This signal can be an artificial potential function given the knowledge of target position, or an actual signal depending on the particular application, such as concentration of a chemical source, an electromagnetic signal emitted by the source of interest, an acoustic signal or a thermal diffusion field.

The choice of the potential function is important in the control design, because different potentials might result in different performances, even with the same control algorithm. In particular, the existence of multiple local minima in the potential function results in only being able to guarantee local convergence to the desired formation. Nevertheless, it is possible to show that by appropriate choice of the potential function, one can always guarantee that eventually the target will be surrounded or "enclosed" by the tracking agents.

It is possible to combine the potential functions using different methods. The simplest method is to take a linear combination of the potential functions for the subtasks. For example, for the swarm tracking problem the overall potential function can be combined as:

$$J(p, p_t) = K_{tar}J_{tar}(p, p_t) + K_{frm}J_{frm}(x) \tag{14}$$

where $K_{tar} > 0$ and $K_{frm} > 0$ are the weights of the potential components.

A block diagram of the overall control structure for the multi-agent dynamic system can be found in Figure 4-6 [55]. In the scheme in Figure 4-6 the extremum seeking controller is assumed to be a gradient-based controller, although it can be any controller. Then with appropriate

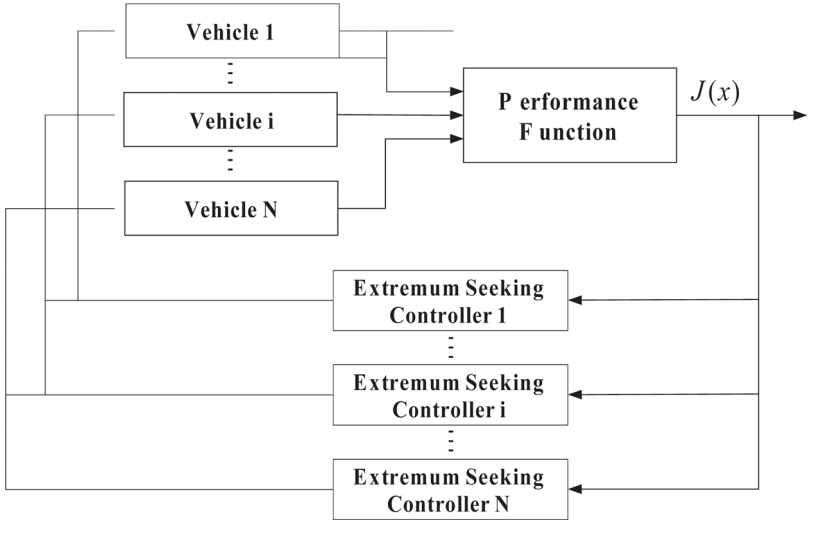


Figure 4-6 Swarm tracking for N agents/vehicles [55].

choice of the potential functions, if the extremum seeking controller for each agent can minimize the performance function (14), we can achieve our objective of source seeking, formation control and collision avoidance (this objective is achieved by adding a repulsive potential between the agents and obstacles into (14)). This control design will be decentralized if each agent has its own performance function.

If the velocity $\dot{p}_t$ of the target is known then for every agent i the control input in the form:

$$\overline{u}_i = \dot{p}_t - k_i \nabla_{pi} J(p, p_t) \tag{15}$$

where $k_i > 0$ guarantees that the Lyapunov function $V = J(p, p_t)$ satisfies

$$\dot{V} = -\sum_{i=1}^{N} k_i \left\| \nabla_{pi} J(p, p_t) \right\|^2 \leq 0 \tag{16}$$

Then using the LaSalle invariance principle it is possible to show that the trajectory of p asymptotically converges to values for which $\|\nabla_{p_i} J(p, p_t)\| = 0$ and $\|\nabla_{p_t} J(p, p_t)\| = 0$. Moreover, it is also possible to show that as $t \to \infty$ we will have:

$$p_t \to conv\{p_1, p_2, \ldots, p_N\},$$

where $conv\{p_1, p_2, \ldots, p_N\}$ is the convex hull of the positions of the agents. In other words, the agents will "surround" or "enclose" the target.

In case $\dot{p}_t$ is not known then under the assumption that $\left\| \dot{p}_t \right\| \leq \gamma_t$ for some known $\gamma_t > 0$ the controller:

$$\overline{u}_i = -k_i \nabla_{pi} J(p, p_t) - \beta_i \operatorname{sgn}(\nabla_{pi} J(p, p_t)) \tag{17}$$

where $k_i > 0$, $\beta_i > \gamma_t$ and $\operatorname{sgn}(\cdot)$ is the signum function, guarantees again that (16) is satisfied.

If only the sign of the gradient $\nabla_{pi} J(p, p_t)$ is known the controller can further be relaxed to:

$$\overline{u}_i = -(k_i + \beta_i) \operatorname{sgn}(\nabla_{pi} J(p, p_t)) \tag{18}$$

guaranteeing the same result.

4.4.4 Adaptive control approaches

Various adaptive techniques have been utilized for formation control and coordination in multi-agent dynamic systems [116–124], mainly for estimation and/or suppression of uncertainties and unknown disturbances present in the system dynamics.

When the uncertainties are parametric or the disturbance can be expressed in a parametric form, various identifier-based adaptive control techniques [125] can be applied. In contrast, if it is not possible to properly parameterize the uncertainties and/or the controller, then it is possible to use universal approximators, such as polynomial-based, neural networks, and fuzzy systems, to estimate and cancel the uncertainty effects [126–131].

A particular application of the identifier-based adaptive control for cooperative target localization and tracking by robot swarms is presented later in Subsection 4.5.3. Next we summarize another series of studies on adaptive formation control using universal approximation and fuzzy techniques. As an example application of this approach: in [132], a distributed adaptive fuzzy control scheme was developed for formation control and target tracking, in the existence of nonlinear and uncertain agent dynamics which are transformable to normal form. This scheme requires the agents to know not only their relative positions to the other agents and the target, but also the time derivatives of these positions up to the order of their relative degrees. Later in [133], this requirement was relaxed by employing high gain observers to estimate the derivatives of the combined agent formation and tracking errors. In [133], an application of the revised scheme involving high gain observers was demonstrated for various formation maneuvers, such as contraction/expansion, rotation, and reconfiguration of the formation. The overall adaptive scheme involved sliding mode rules and bounding terms as well, in order to guarantee boundedness and asymptotic stability of the formation and tracking errors.

4.4.5 Other nonlinear methods

In addition to the methods mentioned in the previous subsections, there exist various other nonlinear control techniques used in the coordination and control of robot swarms. Among the most popular and promising approaches, we see *feedback linearization* [57, 134, 135], *neural network* [136, 137], *model prediction* [134, 138, 139], *output regulation*

(*linear or nonlinear servomechanism*) [30, 39–46, 140–143], and *passivity-based* [9, 119] approaches.

4.5 Swarm robotic applications

Although no commercial application using swarm robotics approaches exists so far, these approaches have been proposed during the past fifteen years or so for a wide variety of potential applications, and a corresponding number of research projects have demonstrated their applicability, either with physical or simulated robots for all domains: ground [144], aerial [145], and marine [146] (the provided references contain examples to the respective domains).

Generally speaking, we can say that swarm robotics concepts are appropriate for spatially distributed tasks benefiting from easy and robust *scalability*, where adding or removing elements to the swarm increases or decreases (respectively) its performance, but does not compromise the global ability to perform the task. Examples of such tasks are static or dynamic coverage and transportation. *Static coverage* is the problem of positioning robots in a stationary configuration that maximizes the probability of detecting an event of interest. Usually, in this type of coverage every point in the environment is under the robots' detection range (i.e., covered) at every instant of time. A related problem is the deployment of communication nodes in a sensor network, guaranteeing that all nodes are reachable (i.e., under communication range). Static coverage is particularly relevant for monitoring dynamic environments or intermittent events that may happen anywhere and anytime, requiring global coverage. References [147–149] provide some proposals to solve this problem.

Dynamic coverage, on the other hand, is the problem of sequentially covering all the workspace. This is particularly relevant for monitoring static or slowly changing environments with scarce re-sources, being physically impossible to cover the entire environment with the existing robots [150]. The dynamic coverage of unknown environments is often called *exploration* [151], while a dynamic coverage aiming to regularly visit different places within the environment is called *patrolling* [152]. Exploration with swarm robotic systems is frequently accomplished with dispersive behaviors or by means of formations that increase the global sensing width. Both types of coverage may be applied to areas (area coverage) or to area boundaries (boundary coverage).

Searching for a given target inside an environment may require sensing all points of that environment (i.e., exploring the environment) or it may be a faster task if the target provides some kind of long range clues that may be used to guide the searching agents (e.g., a chemical source generating a chemical plume) [153]. If the found targets should be transported to a given collecting point, then we will be talking about a foraging problem [154]. Some items may not be transportable by a single robot and require the cooperation of multiple ones. This raises the problem of coordinating individuals in order to perform *collective transportation* [155].

4.5.1 Static coverage

Static area coverage by a robot swarm in order to optimally detect some kind of intermittent event, source of energy or chemical requires the positioning of the swarm elements in a given formation. This formation may correspond to a regular pattern for omnidirectional sensors positioned in open environments, or to distorted patterns for directional sensors or environments with avoiding areas such as obstacles. Example applications for static coverage are the maintenance of the communication infrastructure, environmental and wild-life monitoring, detection of chemical leaks, hot-spots, sound [156] or radio frequency sources, and surveillance. Most of these applications have been addressed within a sensor network framework (i.e., fixed nodes), but some works employ a robotic swarm approach, allowing the adjustment of proper formations to changing environments or swarm disturbances.

The SMAVNET project provides an example of static coverage applied to real-world applications. This project aims to use swarms of Unmanned Aerial Vehicles (UAVs) in disaster scenarios to create and maintain a communication network for first responders. The motivation behind flying robots was their mobility and the benefit of providing line-of-sight communication [157]. A very different example is the detection of chemical traces. Odor sources release chemical substances that are transported by the fluid flow and dispersed by turbulent phenomena, generating a flow-biased Gaussian plume [153] whose orientation may change in time. The optimal detection of such plumes may be achieved by the positioning of a sensor array aligned with the flow, at lateral and longitudinal distances that depend on the flow speed,

turbulence, and sensors' sensitivity. A swarm equipped with sensors to measure such variables may adjust itself to be always optimally positioned to detect a chemical release inside the workspace [158].

Another interesting approach is changing the swarm distribution in order to capture a monitored phenomenon with different levels of detail, minimizing the global uncertainty. Lynch et al. [159] considered the problem of modeling environmental fields, such as the temperature or salinity of a region of an ocean. The authors propose a scalable and decentralized approach to fuse sensor data into a global model of the environment. Additionally, the mobile sensors are moved to maximize their sensory information relative to current uncertainties in the model. On the other hand, the most common situation is dealing with static environments and more or less radial phenomena. Cortes et al. [149] surveyed the coverage with radial sensors, and proposed an asynchronous distributed implementation for coverage control with mobile sensing networks. Dirafzoon and Lobaton [160] employed a minimalist approach to infer the topology of an environment using agents without explicit localization abilities. The agents moved at constant velocity, following walls to explore the environment (like cockroaches). When a pair of agents interacts near a wall, that interaction contains information allowing the inferring of global topological information of the physical environment (i.e., a map).

4.5.2 Dynamic coverage

In a dynamic coverage framework, the swarm has no physical capacity to cover all the workspace at once, covering it sequentially according to some criteria. This is a common approach when searching stationary targets across large search spaces with small detection width sensors (e.g., for de-mining [161]) or small actuators (e.g., for cleaning [162]).

Some common examples of dynamic area coverage are floor and ocean cleaning, de-mining, and inspection of structures. Rutishauser et al. [163] studied and proposed an online coverage algorithm for inspection of turbine blades with a swarm of miniature robots (less than one cubic inch). Larionova et al. [162] showed cooperative floor cleaning with mobile robots equipped with smell sensors. The robots sensed cleaning chemicals released from the ground, being able to automatically coordinate their motion in order to perform complete coverage of dirty areas without explicit or centralized coordination. A related approach, but

validated only in simulation, was proposed by Jin and Ray for oil spill cleaning in a harbor environment [164].

Perimeter detection and boundary coverage approaches are also frequently employed in oil spill monitoring and other applications involving well-defined frontiers, such as forest fire surveillance [165], crop monitoring, and border or facility surveillance [166]. After some severe oil spill accidents occurred in the past years, like the recent Deepwater Horizon oil spill in the Gulf of Mexico, a growing interest has emerged in developing technologies and methods to mitigate the effects and cost of such events. Some examples of these are the EU-MOP [167] and the SeaSwarm projects [168], as well as several research projects dealing with boundary detection and tracking, using simple reactive motion control [169] or a hybrid hierarchical control, based on three behaviors: perimeter searching, pursuing and tracking [170].

Extending the previous example of odor plume detection to a complete formulation of plume search, plume tracking and odor source localization in a dynamic coverage approach, we may consider a swarm in optimal formation sweeping the environment in a crosswind-biased direction while searching for odor plumes [171]. If a swarm element senses an odor clue (i.e., it gets in contact with a plume), the swarm, or a part of it, may self-organize, tracking the plume in the upwind direction. This approach has the benefit of providing spatial filtering to the measured concentrations, providing a more robust estimate of the concentration gradient [172]. When a group of elements cover the source area, a vector divergence operator may be employed to detect the existence of a source [173]. Later, its properties may be sensed by a swarm element staying in the neighborhood and propagated to the other swarm elements, so they keep searching other odor sources [174].

When searching targets with reduced probability of being detected, like an odor source, it provides similar performance whether searching the environment systematically or randomly. In this case, metaheuristic search algorithms, which demonstrate explorative behavior when no clues about the target location is detected but turn exploitative when some are detected, have been demonstrated to perform better than conventional searching algorithms for similar conditions [153, 175, 176]. A limitation of these algorithms used to be dealing with multiple sources. This problem has been addressed by Menon and Ghose with

a Glowworm-inspired algorithm for simultaneous source localization and boundary mapping [177].

4.5.3 Cooperative target localization and tracking

The term *target localization* refers to estimation of the precise location of a *target* T by a set $S = \{S_1, \ldots, S_N\}$ of *sensory agents* from multiple, typically noisy and nonlinear geometrical measurements related to relative positions of $S_1, \ldots, S_N$ to T [50, 178]. The sensory agent set S may be composed of a single sensory agent or multiple agents working cooperatively. The target T is either a foreign entity to be identified and probably captured, an agent S_i within the set S which needs self-localization, or a beacon signal source to be used for self-localization of the whole sensor set S.

For localization, the agents within S are typically equipped with one particular type of measurement unit. Most common measurement techniques include bearing measurements based on signal angle-of-arrival, range measurements based on received signal strength or signal time-of-arrival, range difference measurements based on signal time difference-of-arrival, and scan-based measurements [115, 179].

In [178], relative sensor-target geometries are investigated resulting in minimization of various measures of uncertainty in the location estimate of the target produced by the sensors cooperatively. A similar study with focus on target tracking is presented in [180]. Such studies justify the importance of maintaining certain geometric formations for a given sensor agent set S in *target localization and tracking tasks*.

The aforementioned cooperative *target localization and tracking* tasks will require the use of distributed dual estimator/adaptive formation control algorithms. The distributed algorithms can be designed integrating various formation control schemes, some of which are mentioned in the previous sections, and suitable optimal cooperative localization algorithms, such as the one studied in [50]. The first step of such a design is presented in [51].

Target localization and target tracking methods are particularly useful for emergency scenarios' first responding teams. Typical examples of such scenarios are firefighting and search and rescue after earthquakes. The GUARDIANS EU project provides some interesting examples of using robot swarms to support human firefighters during the early phase of industrial fires, searching for hot spots, fire flames and

chemical sources while assessing the risk level inside areas filled with smoke, toxic gases, and inflammable materials [144]. Furthermore, the system provides human-swarming interaction, letting a firefighter guide a group of agents during the exploration phase, or in a reverse behavior, being helped by a swarm of robots through escaping trajectories [181]. An additional service provided by the swarm is maintaining communication links among the operating agents and an external supervising and commanding station [182].

4.6 Concluding remarks

In this chapter, a brief review of robot swarm systems is provided mainly from the aspects of dynamics and control. Within this context the most popular robot agent dynamic models, swarm coordination and control problems, as well as various control design approaches to these problems, were presented together with some applications. Due to space limitations, the presentation is kept very brief and the discussed literature is not exhaustive. Nevertheless, an extensive list of references has been provided for readers to search for details of the briefly introduced models, problems, control designs and applications. The field of robot swarm systems, or multi-agent dynamic systems, is still an active research field. Beside the significant number of problems solved in the literature, there still exist many potential fruitful research directions. Future research in the field can concentrate on extending the research on robust and adaptive strategies for decentralized coordination and control under model and sensor uncertainties. Developing hardware-specific, more realistic sensor/interaction models may also be a useful approach. Communication-constrained swarm coordination and control is also an important topic of present interest. Moreover, developing models and strategies for human-swarm interaction, intelligent learning strategies and various levels of autonomy/intelligence, for individual agents and the swarm as a whole, are important topics of future research.

4.7 References

[1] C. W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. *Comp. Graph.*, **21**(4):25–34, 1987.

[2] M. Dorigo and E. Sahin, editors. *Special Issue on Swarm Robotics*, volume **17**. Autonomous Robots, September 2004.

[3] V. Gazi, editor. *Special Issue on Swarm Robotics*, volume **15**. Turkish Journal of Electrical Engineering and Computer Sciences, July 2007.

[4] E. Sahin and W. M. Spears, editors. *Swarm Robotics, A State of the Art Survey*. Lecture Notes in Computer Science 3342. Springer-Verlag, Berlin Heidelberg, 2005.

[5] E. Sahin, W. M. Spears, and A. F. T. Winfield, editors. *Proceedings of the SAB06 Workshop on Swarm Robotics Swarm Robotics*. Lecture Notes in Computer Science. Springer-Verlag, Berlin Heidelberg, 2007.

[6] V. J. Kumar, N. E. Leonard, and A. S. Morse, editors. *Cooperative Control: 2003 Block Island Workshop on Cooperative Control*, volume 309 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, 2005.

[7] J. Shamma, editor. *Cooperative Control of Distributed Multi-Agent Systems*. Wiley-Interscience, 2008.

[8] W. M. Spears and D. F. Spears, editors. *Physicomimetics: Physics-Based Swarm Intelligence*. Springer Verlag, January 2012.

[9] H. Bai, M. Arcak, and J. Wen. *Cooperative Control Design: A Systematic, Passivity-Based Approach*. Communications and Control Engineering. Springer-Verlag, New York, 2011.

[10] V. Gazi and K. M. Passino. *Swarm Stability and Optimization*. Springer Verlag, January 2011.

[11] K. M. Passino. *Biomimicry for Optimization, Control, and Automation*. Springer-Verlag, London, UK, 2005.

[12] W. Ren and R. Beard. *Distributed Consensus in Multi-vehicle Cooperative Control*. Communication and Control Engineering Series. Springer Verlag, New York, 2007.

[13] F. Bullo, J. Cortes, and S. Martinez. *Distributed Control of Robotic Networks*. Series in Applied Mathematics. Princeton University Press, 2009.

[14] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, New York, 1999.

[15] B. Fidan, V. Gazi, S. Zhai, N. Cen, and E. Karataş. Single view distance estimation based formation control of robotic swarms. *IEEE Tr. Industrial Electronics*, **60**(12):5781–5791, December 2013.

[16] V. Gazi and B. Fidan. Coordination and control of multi-agent dynamic systems: Models and approaches. In E. Sahin, W. M. Spears, and A. F. T. Winfield, editors, *Proceedings of the SAB06 Workshop on Swarm Robotics Swarm Robotics*, Lecture Notes in Computer Science (LNCS). Springer-Verlag, Berlin Heidelberg, 2007.

[17] H. Yamaguchi. A cooperative hunting behavior by mobile-robot troops. *The International Journal of Robotics Research*, **18**(8):931–940, September 1999.

[18] J. P. Desai, J. Ostrowski, and V. Kumar. Modeling and control of formations of nonholonomic mobile robots. *IEEE Trans. on Robotics and Automation*, **17**(6):905–908, December 2001.

[19] J. M. Fowler and R. D'Andrea. A formation flight experiment. *IEEE Control Systems Magazine*, **23**(5):35–43, October 2003.

[20] W. Ren and R. W. Beard. A decentralized scheme for spacecraft formation flying via the virtual structure approach. *AIAA Journal of Gudiance, Control and Dyanmics*, **27**(1):73–82, 2004.

[21] D. J. Stilwell, B. E. Bishop, and C. A. Sylvester. Redundant manipulator techniques for partially decentralized path planning and control of a platoon of autonomous vehicles. *IEEE Tr. Systems Man and Cybernetics Part B-Cybernetics*, **35**(4):842–848, 2005.

[22] J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Trans. on Robotics and Automation*, **20**(2):243–255, 2004.

[23] I. F. Akyildiz, W. Su, Y. Sankarasubramniam, and E. Cayirci. A survey on sensor networks. *IEEE Commununications Magazine*, **40**(8):102–114, August 2002.

[24] R. Vaughan, N. Sumpter, J. Henderson, A. Frost, and S. Cameron. Experiments in automatic flock control. *Robotics and Autonomous Systems*, **31**:109–117, 2000.

[25] R. C. Kube and E. Bonabeau. Cooperative transport by ants and robots. *Robotics and Autonomous Systems*, **30**(1/2):85–101, 2000.

[26] R. L. Raffard, C. J. Tomlin, and S. P. Boyd. Distributed optimization for cooperative agents: Application to formation flight. In *Proc. Conf. Decision Contr.*, **3**:2453–2459, December 2004.

[27] V. Gazi and K. M. Passino. Stability analysis of swarms. *IEEE Trans. on Automatic Control*, **48**(4):692–697, April 2003.

[28] V. Gazi and K. M. Passino. Stability analysis of social foraging swarms. *IEEE Trans. on Systems, Man, and Cybernetics: Part B*, **34**(1):539–557, February 2004.

[29] V. Gazi and K. M. Passino. A class of attraction/repulsion functions for stable swarm aggregations. *Int. J. Control*, **77**(18):1567–1579, December 2004.

[30] V. Gazi. Formation control of a multi-agent system using nonlinear servomechanism. *Int. J. Control*, **78**(8):554–565, 20 May 2005.

[31] V. Gazi. Swarm aggregations using artificial potentials and sliding mode control. *IEEE Trans. on Robotics*, **21**(6):1208–1214, December 2005.

[32] J. Yao, R. Ordóñez, and V. Gazi. Swarm tracking using artificial potentials and sliding mode control. *Journal of Dynamic Systems, Measurement and Control*, **129**(5):749–754, September 2007.

[33] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. on Automatic Control*, **48**(6):988–1001, June 2003.

[34] L. Moreau. Stability of multiagent systems with time-dependent communication links. *IEEE Trans. on Automatic Control*, **50**(2): 169–182, February 2005.

[35] W. Ren and R. W. Beard. Consensus seeking in multi-agent systems under dynamically changing interaction topologies. *IEEE Trans. on Automatic Control*, **50**(5):655–661, May 2005.

[36] V. Gazi. Stability of a discrete-time asynchronous swarm with time-dependent communication links. *IEEE Tr. Systems, Man, and Cybernetics: Part B*, **38**(1):267–274, February 2008.

[37] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. on Automatic Control*, **49**(9):1520–1533, September 2004.

[38] Z. Lin, B. Francis, and M. Maggiore. Necessary and sufficient graphial conditions for formation control of unicycles. *IEEE Trans. on Automatic Control*, **50**(1):121–127, January 2005.

[39] P. Wieland and F. Allgöwer. An internal model principle for consensus in heterogeneous linear multi-agent systems. In *1st IFAC Workshop on Estimation and Control in Networked Systems*, pages 7–12, Venice, Italy, September 2009.

[40] P. Wieland and F. Allgöwer. An internal model principle for synchronization. In *IEEE International Conference on Control and Automation (ICCA)*, pages 285–290, Christchurch, New Zealand, December 2009.

[41] P. Wieland, R. Sepulchre, and F. Allgöwer. An internal model principle is necessary and sufficient for linear output synchronization. *Automatica*, **47**(5):1068–1074, 2011.

[42] J. Xiang, W. Wei, and Y. Li. Synchronized output regulation of linear networked systems. *IEEE Transactions on Automatic Control*, **54**(6):1336–1341, June 2009.

[43] H. Kim, H. Shim, and J. H. Seo. Output consensus of heterogeneous uncertain linear multi-agent systems. *IEEE Tr. Automatic Control*, **56**(1):200–206, January 2011.

[44] H. F. Grip, T. Yang, A. Saberi, and A. A. Stoorvogel. Output synchronization for heterogeneous networks of non-introspective agents. *Automatica*, **48**(10):2444–2453, October 2012.

[45] Z. Ding. Consensus output regulation of a class heterogeneous nonlinear systems. *IEEE Tr. Automatic Control*, **58**(10):2648–2653, October 2013.

[46] C. D. Persis, H. Liu, and M. Cao. Robust decentralized output regulation for uncertain heterogeneous systems. In *American Control Conference*, pages 5214–5219, Montreal, Canada, June 2012.

[47] C. Zhang, A. Siranosian, and M. Krstić. Extremum seeking for moderately unstable systems and for autonomous target tracking without position measurements. *Automatica*, **43**:1832–1839, 2007.

[48] C. Zhang, D. Arnold, N. Ghods, A. Siranosian, and M. Krstić. Source seeking with nonholonomic unicycle without position measurement and with tuning of forward velocity. *Systems and Control Letters*, **56**:245–252, 2007.

[49] J. Cochran and M. Krstić. Nonholonomic source seeking with tuning of angular velocity. *IEEE Tr. Automatic Control*, **54**(4):713–731, 2009.

[50] B. Fidan, S. Dasgupta, and B. D. O. Anderson. Guaranteeing practical convergence in algorithms for sensor and source localization. *IEEE Tr. Signal Processing*, **56**(9):4458–4469, September 2008.

[51] B. Fidan, S. Dasgupta, and B. D. O. Anderson. Adaptive range-measurement-based target pursuit. *International Journal of Adaptive Control and Signal Processing*, **27**(1):66–81, January 2013.

[52] N. Ghods and M. Krstić. Speed regulation in steering-based source seeking. *Automatica*, **46**:452–459, 2010.

[53] N. Ghods and M. Krstić. Source seeking with very slow or drifting sensors. *ASME Journal of Dynamic Systems, Measurement, and Control*, **133**, 2011.

[54] J. Cochran and M. Krstić. 3-d source seeking for underactuated vehicles without position measurement. *IEEE Tr. Robotics*, **25**(1):117–129, 2009.

[55] C. Zhang and R. Ordóñez. *Extremum Seeking Control and Applications – A Numerical Optimization Approach*. Springer, 2011.

[56] T. Balch and R. C. Arkin. Behavior-based formation control for multirobot teams. *IEEE Trans. on Robotics and Automation*, **14**(6):926–939, December 1998.

[57] J. R. T. Lawton, R. W. Beard, and B. J. Young. A decentralized approach to formation maneuvers. *IEEE Tr. Robotics and Automation*, **19**(6):933–941, December 2003.

[58] R. W. Beard, J. Lawton, and F. Y. Hadaegh. A coordination architecture for spacecraft formation control. *IEEE Tr. Control Systems Technology*, **9**(6):777–790, 2001.

[59] J. P. Desai, J. P. Ostrowski, and V. Kumar. Modeling and control of formations of nonholonomic mobile robots. *IEEE Tr. Robotics and Automation*, **17**(6):905–908, 2001.

[60] A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and C. J. Taylor. A vision-based formation control framework. *IEEE Tr. Robotics and Automation*, **18**(5):813–825, October 2002.

[61] B. Fidan, B. D. O. Anderson, C. Yu, and J. M. Hendrickx. *Persistent autonomous formations and cohesive motion control*, pages 247–275. Taylor & Francis, 2007.

[62] H. G. Tanner, G. J. Pappas, and V. Kumar. Leader-to-Formation Stability. *IEEE Tr. Robotics and Automation*, **20**(3):443–455, June 2004.

[63] F. Xie and R. Fierro. Stabilization of Nonholonomic Robot Formations: A First State Contractive Model Predictive Control Approach. *Journal of Computing and Information Technology*, **1**:37–50, 2009.

[64] P. Ogren, E. Fiorelli, and N. E. Leonard. Cooperative Control of Mobile Sensor Networks: Adaptive Gradient Climbing in a Distributed Environment. *IEEE Tr. Automatic Control*, **49**(8):1292–1302, August 2004.

[65] J. A. Fax and R. M. Murray. Information flow and cooperative control of vehicle formations. *IEEE Tr. Automatic Control*, **49**(9):1465–1476, 2004.

[66] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Tr. Automatic Control*, **48**(6):988–1001, June 2003.

[67] O. A. A. Orqueda, X. T. Zhang, and R. Fierro. An output feedback nonlinear decentralized controller for unmanned vehicle coordination. *Int. J. Robust Nonlinear Control*, **17**(12):1106–1128, 2007.

[68] W. Dong and J. A. Farrell. Decentralized cooperative control of multiple nonholonomic dynamic systems with uncertainty. *Automatica*, **45**(3):706–710, March 2009.

[69] C. M. Saaj, V. Lappas, and V. Gazi. Spacecraft swarm navigation and control using artificial potential field and sliding mode control. In *IEEE International Conference on Industrial Technology*, pages 2646–2652, Mumbai, India, December 2006.

[70] J. Guldner and V. I. Utkin. Sliding mode control for gradient tracking and robot navigation using artificial potential fields. *IEEE Trans. on Robotics and Automation*, **11**(2):247–254, April 1995.

[71] G. Campion, G. Bastin, and B. Dandrea-Novel. Structural properties and classification of kinematic and dynamicmodels of wheeled mobile robots. *IEEE Tr. Robotics and Automation*, **12**(1):47–62, February 1996.

[72] B.-J. Yi and W. K. Kim. The kinematics for redundantly actuated omnidirectional mobile robots. *Journal of Robotic Systems*, **19**(6):255–267, 2002.

[73] J. Yao, R. Ordóñez, and V. Gazi. Swarm tracking using artificial potentials and sliding mode control. In *Proc. Conf. Decision Contr.*, pages 4670–4675, San Diego, CA, USA, December 2006.

[74] V. Gazi and R. Ordóñez. Target tracking using artificial potentials and sliding mode control. *Int. J. Control*, **80**(10):1626–1635, October 2007.

[75] R. W. Brockett. Asymptotic stability and feedback stabilization. In R. S. Millman and H. J. Sussmann, editors, *Differential Geometric Control Theory*, pages 181–191. Birkhauser, 1983.

[76] A. Astolfi. Discontinuous control of nonholonomic systems. *Systems & Control Letters*, **27**:37–45, 1996.

[77] D. Lee. Passive Decomposition and Control of Nonholonomic Mechanical Systems. *IEEE Tr. Robotics*, **26**(6):978–992, December 2010.

[78] A. T. Samiloglu, V. Gazi, and A. B. Koku. Comparison of three orientation agreement strategies in self-propelled particle systems

with turn angle restrictions in synchronous and asynchronous settings. *Asian Journal of Control*, **10**(2):212–232, March 2008.

[79] D. Grünbaum. Schooling as a strategy for taxis in a noisy environment. In J. K. Parrish and W. M. Hamner, editors, *Animal Groups in Three Dimensions*, pages 257–281. Cambridge University Press, 1997.

[80] D. Grünbaum. Schooling as a strategy for taxis in a noisy environment. *Evolutionary Ecology*, **12**:503–522, 1998.

[81] D. Jin and L. Gao. Stability analysis of swarm based on double integrator model. In D.-S. Huang, K. Li and G. W. Irwin, editors, *Proceedings of ICIC 2006*, Lecture Notes in Bioinformatics (LNBI) 4115, pages 201–210. Springer-Verlag, Berlin Heidelberg, 2006.

[82] D. Jin and L. Gao. Stability analysis of a double integrator swarm model related to position and velocity. *Transactions of the Institute of Measurement and Control*, **30**(3):275–293, 2008.

[83] Y. Liu and K. M. Passino. Stable social foraging swarms in a noisy environment. *IEEE Tr. Automatic Control*, **49**(1):30–44, 2004.

[84] V. Gazi and K. M. Passino. Swarm stability. In W. S. Levine, editor, *The Control Handbook: Control System Applications*. CRC Press, December 2010.

[85] V. Gazi. On lagrangian dynamics based modeling of swarm behavior. *Physica D: Nonlinear Phenomena*, **260**:159–175, October 2013.

[86] V. Gazi, B. Fidan, Y. S. Hanay, and M. İ. Köksal. Aggregation, foraging, and formation control of swarms with non-holonomic agents using potential functions and sliding mode techniques. *Turkish Journal of Electrical Engineering and Computer Sciences*, **15**(2):149–168, July 2007.

[87] J. Kennedy and R. C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Publisher, 2001.

[88] M. Clerc and J. Kennedy. The particle swarm—explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. on Evolutionary Computation*, **6**(1):58–73, February 2002.

[89] K. M. Passino. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine*, **22**(3): 52–67, June 2002.

[90] S. B. Akat, V. Gazi, and L. Marques. Asynchronous particle swarm optimization based search with a multi-robot system: Simulation and implementation on a real robotic system. *Turkish Journal of Electrical Engineering and Computer Sciences (ELEKTRIK)*, **8**(5):749–764, 2010.

[91] M. Turduev, G. Cabrita, M. Kırtay, V. Gazi, and L. Marques. Experimental studies on chemical concentration map building by a multi-robot system using bio-inspired algorithms. *Journal of Autonomous Agents and Multi-Agent Systems*, **28**(1):72–100, 2014.

[92] O. Soysal and E. Sahin. Probabilistic aggregation strategies in swarm robotic systems. In *Proc. of the IEEE Swarm Intelligence Symposium*, Pasadena, California, June 2005.

[93] E. Bahceci and E. Sahin. Evolving aggregation behaviors for swarm robotic systems: A systematic case study. In *Proc. of the IEEE Swarm Intelligence Symposium*, Pasadena, California, June 2005.

[94] V. Gazi, B. Fidan, R. Ordóñez, and M. İ. Köksal. A target tracking approach for non-holonomic agents based on artificial potentials and sliding mode control. *Journal of Dynamic Systems, Measurement and Control*, **134**(6), November 2012.

[95] M. Defoort, T. Floquet, A. Kokosy, and W. Perruquetti. Sliding-mode formation control for cooperative autonomous mobile robots. *IEEE Tr. Industrial Electronics*, **55**(11):3944–3953, November 2008.

[96] R. Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Trans. on Automatic Control*, **51**(3):401–420, March 2006.

[97] R. Olfati-Saber and R. M. Murray. Distributed cooperative control of multiple vehicle formations using structural potential functions. In *Proc. IFAC World Congress*, Barcelona, Spain, June 2002.

[98] M. Egerstedt and X. Hu. Formation constrained multi-agent control. *IEEE Trans. on Robotics and Automation*, **17**(6):947–951, December 2001.

[99] J. H. Reif and H. Wang. Social potential fields: A distributed behavioral control for autonomous robots. *Robotics and Autonomous Systems*, **27**:171–194, 1999.

[100] J. Cochran, E. Kanso, and M. Krstić. Source seeking for a three-link model of fish locomotion. In *Proceedings of the American Control Conference*, pages 1808–1813, 2009.

[101] J. Cochran, E. Kanso, S.D. Kelly, H. Xiong, and M. Krstić. Source seeking for two nonholonomic models of fish locomotion. *IEEE Tr. Robotics*, **25**(5):1166–1176, 2009.

[102] M. Krstić and J. Cochran. Extremum seeking for motion optimization: From bacteria to non-holonomic vehicles. In *Proceedings of the Chinese Control and Decision Conference*, pages 18–27, 2008.

[103] Lina Fu and Ümit Özgüner. Sliding mode in constrained source tracking with non-holonomic vehicles. In *Variable Structure Systems, 2008. VSS'08. International Workshop on*, pages 30–34, 2008.

[104] L. Fu and U. Ozguner. Extremum-seeking control in constrained source tracing with nonholonomic vehicles. *IEEE Tr. Industrial Electronics*, **56**(9):3602–3608, 2009.

[105] M. S. Stankovic and D. M. Stipanovic. Stochastic extremum seeking with applications to mobile sensor networks. In *Proceedings of the American Control Conference*, pages 5622–5627, 2009.

[106] M. S. Stankovic and D. M. Stipanovic. Discrete time extremum seeking by autonomous vehicles in a stochastic environment. In *Proceedings of the Conference on Decision and Control*, pages 4541–4546, 2009.

[107] S. Liu and M. Krstić. Stochastic source seeking for nonholonomic unicycle. *IEEE Tr. Automatic Control*, **46**:1443–1453, 2010.

[108] N. E. Leonard and E. Fiorelli. Virtual leaders, artificial potentials and coordinated control of groups. In *Proc. Conf. Decision Contr.*, pages 2968–2973, Orlando, FL, December 2001.

[109] Y. L. Chuang, Y. R. Huang, M. R. D'Orsogna, and A. L. Bertozzi. Multi-vehicle flocking: Scalability of cooperative control algorithms using pairwise potentials. In *Proc. of IEEE International Conference on Robotics and Automation*, pages 2292–2299, Roma, Italy, April 2007.

[110] A. Mogilner and L. Edelstein-Keshet. A non-local model for a swarm. *Journal of Mathematical Biology*, **38**:534–570, 1999.

[111] H. Levine and W.-J. Rappel. Self-organization in systems of self-propelled particles. *Physical Review E*, **63**(1):017101-1–017101–4, January 2001.

[112] M. R. D'Orsogna, Y. L. Chuang, A. L. Bertozzi, and L. S. Chayes. Self-propelled particles with soft-core interactions: Patterns, stability, and collapse. *Physical Review Letters*, **96**, March 2006.

[113] Y. L. Chuang, M. R. D'Orsogna, D. Marthalerc, A. L. Bertozzi, and L. S. Chayes. State transitions and the continuum limit for a 2d interacting, self-propelled particle system. *Physica D*, **232**:33–47, 2007.

[114] R. C. Fetecau, Y. Huang, and T. Kolokolnikov. Swarm dynamics and equilibria for a nonlocal aggregation model. *Nonlinearity*, **24**:2681–2716, 2011.

[115] G. Mao and B. Fidan, editors. *Localization Algorithms and Strategies for Wireless Sensor Networks*. IGI Global – Information Science Publishing, 2009.

[116] S. S. Joshi. Multilevel adaptation for autonomous mobile robot formations. In *Guidance, Navigation, and Control Conference*, Denver, CO, USA, August 2000.

[117] F. Hadaegh, W. M. Lu, and P. Wang. Adaptive control of formation flying spacecraft for interferometry. In *Large Scale Systems: Theory and Applications*, Patras, Greece, July 1998.

[118] H. C. Lim and H. Bang. Adaptive control for satellite formation flying under thrust misalignment. *Acta Astronautica*, **65**:112–122, July–August 2009.

[119] H. Bai, M. Arcak, and J. T. Wen. Adaptive motion coordination: Using relative velocity feedback to track a reference velocity. *Automatica*, **45**:1020–1025, April 2009.

[120] X. Chen and Y. Li. Stability on adaptive nn formation control with variant formation patterns and interaction topologies. *International Journal of Advanced Robotic Systems*, **5**(1), 2008.

[121] J. Lawton, R. B. Beard, and F. Y. Hadaegh. An adaptive control approach to satellite formation flying with relative distance constraints. In *American Control Conference*, volume **3**, pages 1545–1549, San Diego, CA, USA, June 1999.

[122] G. Lee and N. Y. Chong. Adaptive flocking of robot swarms: Algorithms and properties. *IEICE Transactions on Communications*, **E91-B**(9):2848–2855, 2008.

[123] S. Güler, N. Köksal, B. Fidan, and V. Gazi. Indirect adaptive formation control with nonlinear dynamics and parametric uncertainty. In *Asian Control Conference*, Istanbul, Turkey, June 2013.

[124] V. Gazi. Distributed adaptive output agreement in a class of multi-agent systems. In *Asian Control Conference*, Istanbul, Turkey, June 2013.

[125] P. A. Ioannou and B. Fidan. *Adaptive Control Tutorial*. SIAM Society for Industrial and Applied Mathematics, 2006.

[126] C. Y. Su and Y. Stepanenko. Adaptive control of a class of nonlinear systems with fuzzy logic. *IEEE Transactions on Fuzzy Systems*, **2**(4):285–294, November 1994.

[127] F. C. Chen and H. K. Khalil. Adaptive control of nonlinear systems using neural networks. *International Journal of Control*, **55**:1299–1317, 1992.

[128] Y. Diao and K. M. Passino. Adaptive neural/fuzzy control for interpolated nonlinear systems. *IEEE Transactions on Fuzzy Systems*, **10**(5):583–595, October 2002.

[129] J. T. Spooner and K. M. Passino. Stable adaptive control using fuzzy systems and neural networks. *IEEE Transactions on Fuzzy Systems*, **4**(3):339–359, August 1996.

[130] J. T. Spooner, M. Maggiore, R. Ordez, and K. M. Passino. *Stable Adaptive Control and Estimation for Nonlinear Systems: Neural and Fuzzy Approximator Techniques*. John Wiley and Sons, Inc., New York, 2002.

[131] K. M. Passino and S. Yurkovich. *Fuzzy Control*. Addison-Wesley Longman, Menlo Park, CA, 1998.

[132] S. Duran and V. Gazi. Adaptive formation control and target tracking in a class of multi-agent systems. In *Proc. American Control Conference*, pages 75–80, Baltimore, MD, USA, June–July 2010.

[133] V. Gazi and B. Fidan. Adaptive formation control and target tracking in a class of multi-agent systems. In *Proc 13th Int. Conference on Control, Automation and Systems*, Gwangju, Korea, October 2013.

[134] H. Fukushima, K. Kon, and F. Matsuno. Model predictive formation control using branch-and-bound compatible with collision avoidance problems. *IEEE Tr. Robotics*, **29**(5):1308–1317, 2013.

[135] R. Vidal, O. Shakernia, and S. Sastry. Following the flock [formation control]. *IEEE Robotics Automation Magazine*, **11**(4): 14–20, 2004.

[136] T. Dierks, B. Brenner, and S. Jagannathan. Neural network-based optimal control of mobile robot formations with reduced information exchange. *IEEE Tr. Control Systems Technology*, **21**(4): 1407–1415, 2013.

[137] H. Wu and S. Jagannathan. Adaptive neural network control and wireless sensor network-based localization for UAV formation. In *Proc. 14th Mediterranean Conference on Control and Automation*, Ancona, Italy, June 2006.

[138] E. Scholte and M. E. Campbell. Robust nonlinear model predictive control with partial state information. *IEEE Tr. Control Systems Technology*, **16**(4):636–651, 2008.

[139] J. Shin and H. J. Kim. Nonlinear model predictive formation flight. *IEEE Tr. Systems, Man and Cybernetics, Part A: Systems and Humans*, **39**(5):1116–1125, 2009.

[140] V. Gazi and K. M. Passino. Decentralized output regulation of a class of nonlinear systems. *Int. J. Control*, **79**(12):1512–1522, December 2006.

[141] E. Gül and V. Gazi. Adaptive internal model based formation control of a class of multi-agent systems with switched exosystems. In *Chinese Control and Decision Conference*, pages 6–13, Taiyuan, China, May 2012.

[142] E. Gül and V. Gazi. Adaptive internal model based formation control of a class of multi-agent systems. In *American Control Conference*, pages 4800–4805, Baltimore, MD, USA, June–July 2010.

[143] V. Gazi. Distributed output agreement in a class of uncertain linear heterogeneous multi-agent dynamic systems. In *European Control Conference*, Strasbourg, France, June 2014.

[144] J. Penders, E. Cervera, U. Witkowski, L. Marques, J. Gancet, P. Bureau, V. Gazi, and R. Guzman. Guardians: A swarm of autonomous robots for emergencies. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07) Workshop on Multirobotic Systems for Societal Applications*, pages 8–16, 2007.

[145] A. Kushleyev, D. Mellinger, C. Powers, and V. Kumar. Towards a swarm of agile micro quadrotors. *Autonomous Robots*, **35**(4):287–300, 2013.

[146] A. Amory, B. Meyer, C. Osterloh, T. Tosik, and E. Maehle. Towards fault-tolerant and energy-efficient swarms of underwater robots. In *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2013 IEEE 27th International*, pages 1550–1553. IEEE, 2013.

[147] J. O'Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, 1987.

[148] A. Howard, M. J. Matarić, and G. S. Sukhatme. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *Distributed Autonomous Robotic Systems* 5, pages 299–308. Springer, 2002.

[149] J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *Robotics and Automation, IEEE Transactions on*, **20**(2):243–255, 2004.

[150] H. Choset. Coverage for robotics–a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, **31**(1–4):113–126, 2001.

[151] A. Marjovi and L. Marques. Multi-robot topological exploration using olfactory cues. In *Distributed Autonomous Robotic Systems*, pages 47–60. Springer, 2013.

[152] D. Portugal and R. P. Rocha. Multi-robot patrolling algorithms: Examining performance and scalability. *Advanced Robotics*, **27**(5):325–336, 2013.

[153] L. Marques, U. Nunes, and A. T. de Almeida. Particle swarm-based olfactory guided search. *Autonomous Robots*, **20**(3):277–287, 2006.

[154] A. F. T. Winfield. Foraging robots. In *Encyclopedia of Complexity and Systems Science*, pages 3682–3700. Springer, 2009.

[155] C. R. Kube and E. Bonabeau. Cooperative transport by ants and robots. *Robotics and Autonomous Systems*, **30**(1):85–101, 2000.

[156] G. Simon, M. Maróti, Á. Lédeczi, G. Balogh, B. Kusy, A. Nádas, G. Pap, J. Sallai, and K. Frampton. Sensor network-based countersniper system. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, pages 1–12. ACM, 2004.

[157] S. Hauert, J.-C. Zufferey, and D. Floreano. Evolved swarming without positioning information: An application in aerial communication relay. *Autonomous Robots*, **26**(1):21–32, 2009.

[158] A. Marjovi and L. Marques. Optimal spatial formation of swarm robotic gas sensors in odor plume finding. *Autonomous Robots*, **35**:1–17, 2013.

[159] K. M. Lynch, I. B. Schwartz, P. Yang, and R. A. Freeman. Decentralized environmental modeling by mobile sensor networks. *Robotics, IEEE Transactions on*, **24**(3):710–724, 2008.

[160] A. Dirafzoon and E. Lobaton. Topological mapping of unknown environments using an unlocalized robotic swarm. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 5545–5551. IEEE, 2013.

[161] T. M. Cheng and A. V. Savkin. Decentralized coordinated control of a vehicle network for deployment in sweep coverage. In *Control and Automation, 2009. ICCA 2009. IEEE International Conference on*, pages 275–279. IEEE, 2009.

[162] S. Larionova, N. Almeida, L. Marques, and A. T. de Almeida. Olfactory coordinated area coverage. *Autonomous Robots*, **20**(3):251–260, 2006.

[163] S. Rutishauser, N. Correll, and A. Martinoli. Collaborative coverage using a swarm of networked miniature robots. *Robotics and Autonomous Systems*, **57**(5):517–525, 2009.

[164] X. Jin and A. Ray. Coverage control of autonomous vehicles for oil spill cleaning in dynamic and uncertain environments. In *American Control Conference (ACC), 2013*, pages 2594–2599. IEEE, 2013.

[165] D. W. Casbeer, D. B. Kingston, R. W. Beard, and T. W. McLain. Cooperative forest fire surveillance using a team of small unmanned air vehicles. *International Journal of Systems Science*, **37**(6):351–360, 2006.

[166] S. Susca, F. Bullo, and S. Martinez. Monitoring environmental boundaries with a robotic sensor network. *Control Systems Technology, IEEE Transactions on*, **16**(2):288–296, 2008.

[167] N. M. P. Kakalis and Y. Ventikos. Robotic swarm concept for efficient oil spill confrontation. *Journal of Hazardous Materials*, **154**(1):880–887, 2008.

[168] SeaSwarm. SeaSwarm project. http://senseable.mit.edu/seaswarm/, 2014.

[169] J. Clark and R. Fierro. Mobile robotic sensors for perimeter detection and tracking. *ISA Transactions*, **46**(1):3–13, 2007.

[170] G. Zhang, G. K. Fricke, and D. P. Garg. Spill detection and perimeter surveillance via distributed swarming agents. *Mechatronics, IEEE/ASME Transactions on*, **18**(1):121–129, 2013.

[171] A. Marjovi and L. Marques. Optimal swarm formation for odor plume finding. *IEEE Transactions on Cybernetics*, **44**(12):2302–2315, 2014.

[172] A. Marjovi and L. Marques. Swarm robotic plume tracking for intermittent and time-variant odor dispersion. In *Proceedings of 6th European Conference on Mobile Robots (ECMR 2013)*, Barcelona, Spain, 2013.

[173] G. Cabrita and L. Marques. Divergence-based odor source declaration. In *Control Conference (ASCC), 2013 9th Asian*, pages 1–6. IEEE, 2013.

[174] G. Cabrita, L. Marques, and V. Gazi. Virtual cancelation plume for multiple odor source localization. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 5552–5558. IEEE, 2013.

[175] L. Marques, U. Nunes, and A. T. De Almeida. Odour searching with autonomous mobile robots: An evolutionary-based approach. In *Proceedings of the IEEE Int. Conf. on Advanced Robotics*, pages 494–500, 2003.

[176] M. Turduev, G. Cabrita, M. Kırtay, V. Gazi, and L. Marques. Experimental studies on chemical concentration map building by a multi-robot system using bio-inspired algorithms. *Autonomous Agents and Multi-Agent Systems*, **28**(1):72–100, 2014.

[177] P. P. Menon and D. Ghose. Simultaneous source localization and boundary mapping for contaminants. In *American Control Conference (ACC), 2012*, pages 4174–4179. IEEE, 2012.

[178] A. N. Bishop, B. Fidan, B. D. O. Anderson, K. Doğancay, and P. N. Pathirana. Optimality analysis of sensor-target localization geometries. *Automatica*, **46**(3):479–492, March 2010.

[179] R. Zekavat and M. Buehrer, editors. *Handbook of Position Location: Theory, Practice and Advances*. Wiley-IEEE Press, 2011.

[180] S. Martinez and F. Bullo. Optimal sensor placement and motion coordination for target tracking. *Automatica*, **42**(4):661–668, April 2006.

[181] J. Gancet, E. Motard, A. Naghsh, C. Roast, M. M.Arancon, and L. Marques. User interfaces for human robot interactions with a swarm of robots in support to firefighters. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2846–2851. IEEE, 2010.

[182] A. Marjovi, L. Marques, and J. Penders. Guardians robot swarm exploration and firefighter assistance. In *Workshop on NRS in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), St Louis, USA*, 2009.

5. Mobile robots for earth exploration: applications, technologies and image processing techniques for navigation

Filippo Bonaccorso, Carmelo Donato Melita, Giovanni Muscato, Università degli Studi di Catania, Italy, Istituto Nazionale di Geofisica e Vulcanologia (INGV), Italy; and Michele Prestifilippo, Istituto Nazionale di Geofisica e Vulcanologia (INGV), Italy

Abstract: This chapter gives an overview of the applications of mobile ground robots for earth exploration. An introduction of the main examples and projects is given, followed by a short survey on the necessary related technologies. The chapter concludes with an example of an image processing algorithm adopted for autonomous navigation in complex outdoor environments.

5.1 Introduction

Exploration of earth has been always one of the most exciting human activities. Since prehistoric times man has had the desire to discover unexplored lands and to better understand the natural phenomena around him [1, 2]. However, these discoveries have often entailed great risks, and many explorers have paid with their lives for their thirst for knowledge.

Nowadays robotics can be of help in exploring our planet and at the same time minimizing the risks for humans. This chapter will give an overview of the robotic systems that have been adopted in the last years for applications related to Earth exploration.

It should be emphasized that a great number of research groups have also investigated the related problem of robots for planetary exploration. The larger interest in these systems was further encouraged by the NASA rovers sent and operated on Mars. However, the problems solved in planetary exploration are very different than those that may be encountered by robots on earth. The weight in a planetary robot is crucial because each kilo in a spacecraft increases the cost. The autonomy in terms of operation is another important point, due to the delay in telecommunications that make telecontrol difficult. Moreover, the robot must also be autonomous from the point of view of energy, since it cannot be easily refueled or recharged. Other important factors are

the capability to resist high temperature variations, strong accelerations during travel and high levels of radiation. The budgets available for such missions is usually very high, compared to those available for service robotics applications intended for wider use. On the other hand, the ability to overlook these important considerations allows an earth robot to be adopted for many other applications.

The following sections will give a short outline of the main applications of robots for earth exploration. It should be emphasized that a complete survey of the state of the art is not the purpose of this chapter, where just a few examples of the different applications and of the related technologies are briefly explained. A detailed analysis of each solution is not easy, and the cited references will help the interested reader to go into the details of each system.

In this chapter we will not examine underwater and aerial robots, since these are rather different systems with respect to terrestrial robots, and should be specifically covered. A more detailed analysis of underwater robots can be found in Chapter 4, while an overview of the particular case of aerial systems lighter than air, is given in Chapter 5.

The sub-section on volcanic exploration will illustrate with details two examples of mobile robots developed in our department. Another section is devoted to related technologies useful for the implementation of a robot for earth exploration. Sun-synchronous systems, traversability analysis, traction control, localization and map-building are some important aspects to be investigated during the design of a real system for unstructured environments. The last section will concentrate on a method developed by our group for road detection and obstacle avoidance. It uses a stereo camera and can be really useful in allowing the autonomous navigation of a robot in the exploration of unknown environments.

5.2 Applications of robots for earth explorations

5.2.1 Volcanic explorations

Among the most interesting environments on earth for robotic exploration are undoubtedly volcanoes. The exploration of volcanoes allows us to better understanding the nature of our planet by analyzing the geological mechanisms. Moreover it enables us to assess situations of potential hazard to people living near an active volcano.

The study of volcanoes becomes fundamental in close proximity of an eruption to better measure the phenomena precursors. However, these

are also conditions of great risk for scientists. Hence, the use of robots becomes a good solution to minimize the hazard for volcanologists.

Our research group has been involved in the last fifteen years in various projects on the use of robots for the exploration of volcanoes. The two main platforms developed are shown in Figures 5-1 and 5-2: Robovolc [3, 4] and U-Go [5, 6]. These have been specifically designed to be used in rough terrains and hard environments.

Robovolc is a robot with special features developed for volcano exploration. For locomotion it has six independently actuated wheels mounted on a passively articulated chassis [7]. The robot has a Science Package which includes a pan-tilt turret, a manipulator and a gas sampling system. The pan-tilt turret, that can be oriented toward the measurement region, has a digital video camera recorder, an infrared camera for thermal measurement, a high resolution still image camera, a video camera, and a radar Doppler for lava and gas-jet speed measurement. A manipulator with 5 degrees of freedom and a gripper are adopted to collect samples of rocks and to get gas samples through the gas sampling system.

The robot U-Go has been mainly developed to solve problems like transportation, navigation and inspection in harsh outdoor environments [8]. In this case, two rubber tracks actuated by two DC motors have been adopted for locomotion. This robot has a weight of about 150 kg and is capable of carrying a payload of 200 kg.

Figure 5-1 The Robovolc system on the Mt. Etna volcano.

Figure 5-2 The U-Go Robot.

Although U-Go and Robovolc have different low-level hardware and architectures, they share a common custom interface allowing them to adopt the same navigation and localization system [5]. The latter is composed of an onboard computer interfaced via a wireless link with the remote base station, and internally with the robot and the localization and navigation sensors. The sensors comprise a stereo camera, an Inertial Measurement Unit, a laser range finder, the motor encoders and a RTK-DGPS receiver. In this way both systems can be teleoperated or programmed to move autonomously. Other relevant volcanic exploration robots are the Dante II [9] and the mixed Aerial and ground system developed by Tohoku University in Japan [10]. A survey of unmanned aerial systems for volcanic applications is reported in [11] while for ground system in the already mentioned references [3] and [4].

5.2.2 Meteorite search

In 1999 the robot NOMAD was developed by the Robotics Institute of Carnegie Mellon within a project funded by NASA [12]. This robot was successfully adopted for the autonomous search and classification of Antarctic meteorites [13]. The NOMAD robot had four independently actuated wheels connected to an articulated-frame averaging suspension system. The sensing system included a DGPS, a stereo camera, a laser range finder and a panoramic camera. During the missions several meteorites were found and successfully classified.

5.2.3 Search and rescue

The use of robots has been proposed to help in search and rescue operations, in particular following environmental disasters [14]. Examples are robots adopted to search for victims after an earthquake or building collapse, or robots for use in fires, floods or post-nuclear accidents [15]. Important examples are represented from the tests performed by the groups of Robin Murphy after the World Trade Center collapse in New York City [16], and the explorations performed by robots following the accident at the Fukushima Daiichi nuclear power plants [17, 18].

A specific conference is held each year on this topic (Safety Search and Rescue Robotics) by IEEE, and IEEE has a technical committee on Safety, Security and Rescue Robotics. The ROBOCUP competition also established the Robocup-Rescue category following the Great Hanshi-Awaji earthquake which hit Kobe City in 1995, causing more than 6500 casualties. The main aim is to promote research and development in this socially significant domain at various levels. There are several categories regarding real or simulated robots, and a junior competition for young students is also organized [19].

5.2.4 Humanitarian demining

Anti-personnel mines have been used in military conflicts since the sixteenth century and are still used by several countries. As a consequence, all around the world there are still hundreds of millions of landmines buried that cause thousands of victims each year. The actual manual methods adopted to neutralize minefields are expensive, dangerous for the operators and very slow. For these reasons several groups have proposed the adoption of robotic systems to find and remove landmines [20–24]. The problem is not easy because minefields can be in really unstructured terrains, very difficult for an autonomous robot [25]. In Japan many research groups have investigated this problem [26], while in Europe the TIRAMISU project has among its aims the development of robotic technologies to help de-miners speed-up landmine field clearing operations [27].

5.2.5 Underground explorations

There are only a few examples of robots developed for underground exploration [28]. Although the applications could be fascinating and important, a robot that can autonomously excavate and explore under the earth is really challenging. Among these examples there is

the underground explorer robot based on the peristaltic crawling of earthworms, shown in [29]. Other applications include robots adopted for mining operations [30, 31] and drilling [32].

5.3 Related technologies

This section concerns some of the related technologies useful for the implementation of a robot for earth exploration. The reader is also referred to other chapters in the book which contain details of additional useful technologies for a mobile ground robot.

5.3.1 Sun synchronous robots

Energy independence is obviously one of the great obstacles to the possibility of adopting a robot for long exploration missions. Sun-synchronous exploration robots solve this problem using particular routes that allow the robot to be always exposed to the sun in order to ensure an adequate exposure to sunlight [33]. In this way solar panels can be adopted to power the system and the robot could virtually move indefinitely. The speed of the vehicle should match the speed of rotation of the planet, so on the Earth for example, this can be achieved on very high latitudes in the summer. Several experiments have been performed by the Robotics Institute at Carnegie Mellon with the Hiperion robot [34–36]. The trials were performed in the Artic Circle in the southernmost latitude at which the sun does not set on the summer solstice. The experiments performed demonstrated the feasibility of the idea. The robot was capable to navigate for 24 hours almost always autonomously, and the energy consumed was practically equal to that generated by the solar panels.

5.3.2 Traversability analysis

Traversability analysis is important in allowing a robot operating on natural terrain to select the safest path autonomously. The traversability index defined in [37, 38] is expressed by linguistic fuzzy sets that represent the suitability of the terrain to be traversed on the basis of several physical properties [39].

Several research activities have been developed on the traversability concept on the basis of the sensor adopted. Some examples include the use of laser scanners [40], stereo cameras [41], image sensors [42] and RGB-D cameras [43].

5.3.3 Localization and map building

An autonomous robot designed to explore the earth must know where he is at all times. This allows scheduling the explorations, geo referencing of the measurements and retracing the path traveled. In unknown environments it is also important to build local maps, to be able to relatively localize the robot with respect to them [44]. Since the statement of the SLAM (Simultaneous Localization and Mapping) problem [45, 46], a huge amount of research has been published concerning different applications and methodologies. As regard the localization task, satellite tracking systems today, such as the GPS, GLONASS or GALILEO, allow one to obtain sub-centimeter accuracies. And due to their simplicity, cost and availability, they are among the most used for the location of a robot in outdoor environments. Multisensor data fusion, combining data measurements from GNSS with odometer, laser scanner or vision system, further improves the localization capabilities in those environments where visibility of the satellite is problematic [47].

5.3.4 Traction control

Many research activities have investigated traction control strategies based on the regulation of a desired slip ratio decided by a road condition estimator [48]. Unfortunately these strategies are optimized for high-speed vehicles (i.e., cars) and for regular terrains. Nevertheless, some analogies can be made with a low-speed vehicle that moves on sandy, slippery terrains [49].

Other centralized strategies require the knowledge of the terrain morphology, and optimize the torques assigned to the various motors in order to obtain the maximum traction [50–52]. These algorithms are generally very efficient, but they require good estimations of both robot attitude and terrain morphology. However, the required sensors are often expensive; moreover, this kind of algorithm is often computationally very expensive, compared to other simpler strategies.

5.4 Current challenges

It should be clear at this point that the applications of mobile ground robots for earth exploration represent a great opportunity for robotics. However, this introduces several challenges regarding the mechanical structure of robots, the hardware involved, in terms of sensors and processing units, and the "intelligence," in terms of algorithms and software systems adopted.

From the mechanical hardware point of view, several attempts have gone beyond classical wheeled robots, such as [53, 54]. Even the Robovolc robot, here reported as a benchmark platform for robotic exploration, has a robust and innovative set up of wheels.

From a sensor point of view, on-board sensing devices are close to the ones adopted in indoor mobile robotics; the main difference is that they must be housed in a safer way, being used outdoors and in harsh environments. As a matter of fact, there will be GPS, laser range finders, sonars, cameras and so on. However, many vision systems and RGB-D sensors extensively used for indoor environments are more difficult to be adapted to sunlight or in continuously changing light conditions.

All those data coming from robots must be processed and used to perform desired tasks which can span among three different types: moving on an a-priori know environment, moving on a partially known one or, the more unusual and challenging topics, moving on an unknown location. Obstacles should be also considered in the surrounding environment: they can be fixed or moving.

Thus, to cope with earth exploration, robots have to deal with canonical topics of robotics: localization, map building, obstacle avoidance and drivable surface detection. Researchers have to always keep in mind an important tradeoff between algorithm capabilities and their computational effort. This is a crucial factor in order to have reliable and functional robots, because the main target is to have real-time processing algorithms allowing fast and efficient robots.

5.5 A road detection and obstacle avoidance method of using a stereo camera for autonomous navigation

This section deals with the autonomous navigation of mobile robots in complex outdoor environments by means of the presented computer-vision-based algorithms. The proposed software framework allows performing both road detection and obstacle avoidance. A drivable surfaces detection algorithm has been tested, even in complex unpaved and off-road terrain conditions, while the obstacle avoidance algorithm is also suitable to detect moving objects within the environment.

A first algorithm is deputed to extract a custom road model from the acquired images. This model, used as a marker for road morphology, is

then adopted to define a drivable surface in the proximity of the robot. Instead of using stationary or pre-processed road models, this algorithm adapts its model to changing environments, allowing the proposed approach to be more robust.

The adoption of a stereo cameras permits detecting obstacles and putting out the elevation data of objects in acquired images, thus allowing it to find drivable corridors.

These algorithms have been integrated into a navigation system based on the Potential Field Method [55, 56], which allows a robot following the path classified as road, and to avoid obstacles.

The navigation system has been tested using two robotic platforms, described previously in the volcanic exploration section Robovolc and U-Go. The reported experimental results show the robustness achieved in autonomous navigation.

5.5.1 Related works and overview

Autonomous navigation of mobile robots has always been one of the major research topics in robotics. Autonomous mobile robots should be able to drive while understanding the surrounding environment. To do so a robot should be aware of the environment, avoiding obstacles and detecting safe navigation paths.

Therefore, many research groups have been interested in computer vision as a method to recognize features in the environment. Several algorithms have been produced to fulfill the gap in robotic perception, facing most of the different tasks crucial to performing reliable autonomous navigation in dynamic environments [57, 58].

Moreover, robotic competitions among autonomous unmanned vehicles in realistic off-road scenarios and terrains, such as the DARPA Grand Challenge [59] and The European Land Robot Trial (ELROB) [60], have further stimulated interest in developing computer-vision-based algorithms for autonomous navigation. Among all possible applications of computer vision for outdoor robotic navigation, road detection and obstacle avoidance are among the most important.

Road detection algorithms may be grouped in two main categories, as they could be adaptive or not. The most commonly used are the non-adaptive algorithms, based on some a-priori knowledge of the environment and on visual features that are used both to define and identify the

road. One category of algorithms searches for image edges defining the roadway, such as lane markers [61] or road borders [62, 63], while the other adopts color information from the road surface [64, 65].

A few algorithms use adaptive learning processes, adopting supervised learning to train classifiers for the identification of road regions [66, 67]. Other methods achieve adaptability by using color information of recent known road regions to identify drivable surfaces [68–71].

Similar consideration can be given for computer-vision-based obstacle detection algorithms. These have gained great advantage from the adoption of stereoscopy [72]. Most of the obstacle-detection algorithms in literature are based on the adoption of laser range finders, due to their precision and reliability [71]. However, stereoscopic cameras are usually much cheaper than laser range finder devices and, in certain cases, can considerably reduce the cost of a road detection and obstacle avoidance system. Moreover, for some particular applications, their adoption, in conjunction with a laser range finder, can greatly improve detection performance.

This work deals with a perception-oriented software framework for autonomous navigation of mobile robots in complex outdoor environments, which allows performing both road detection and obstacle avoidance. Roads are classified by using a real-time adjusted model defined by color information: instead of relying on a static pre-computed set of possible terrains, the model is continuously updated by using the features of the terrain surrounding the robot. Obstacles are detected by means of depth information elaborated from the stereo cameras.

This framework is part of a more global control system [6, 8], developed to perform outdoor autonomous navigation in unknown environments: several data from different sensors are fused in order to provide localization and a navigation strategy. However, in this section only the computer vision algorithms will be described, and some results of autonomous navigation tests obtained by using exclusively the visual feedback will be presented.

5.5.2 Drivable surface detection outline

For drivable surface detection, an approach similar to those described in [68, 69, 71] is adopted. In these works a supervised algorithm permits creation of a database of the scenarios, allowing the online identification of possible safe patterns.

Algorithms like those reported in [8, 71] combine information from three sensors: a laser range finder, a pose estimation system and a color camera. On the contrary, the approach presented in this paper simply relies on the adoption of a stereo camera pair.

The steps involved in the algorithm for Drivable Surface Detection (DSD), shown in Figure 5-3, are:

1. Image Acquisition
2. Bird's eye transformation
3. Kernel Matrix Generation
4. Image processing

This subsection will introduce basic concepts of proposed algorithms, describing early work developed, while the next subsection will more deeply describe the implementation and optimization steps.

Image acquisition

An auto-calibration tool for stereo cameras has been developed in order to easily and quickly compute cameras' intrinsic, extrinsic and distortion parameters used to pre-process the images coming from the stereo cameras [73]. The values of the parameters are strictly related to the camera and are used to define the relationship between a point in space, the road plane in this approach, and its corresponding point on the image's plane. It is important to highlight a particular relation allowing construction, from a camera-acquired image, of an artificial image that is more suitable to analyze road characteristics, and to identify them. This relation is the *planar homography*, or *projective transformation*, which is a non-singular linear transformation between points defined in R^2. Figure 5-4 summarizes how planar homography transforms points from the road to the image planes. The planar homography can be represented by means of a relation h, defined as follows:

$$h : P \rightarrow \overline{P}$$

Figure 5-3 Steps performed in the Drivable Surface Detection Algorithm.

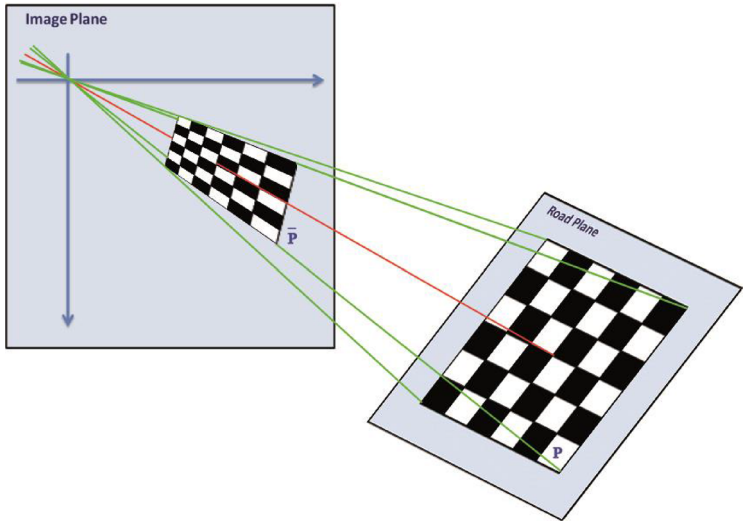


Figure 5-4 Points from the road plane are mapped into the image plane using planar homography transformation.

where P and $\bar{P}$, both defined in R_2, are the sets of points in the road plane and the image plane respectively. This relation defines a homography if h can be defined as:

$$h(x) = Hx \tag{2}$$

with H non-singular 3×3 matrix and $x = (x_1, x_2, 1)^T$ is a 3×1 vector defining a point in the road plane using homogeneous coordinates. The complete relation is defined by the following equation where λ is a scale factor:

$$\lambda \begin{bmatrix} \overline{x_1} \\ \overline{x_2} \\ 1 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} \tag{3}$$

At least four known points are needed to compute H parameters, except the scale factor [73, 74].

Bird's eye transformation

Inverse Perspective Mapping (IPM) can be used in order to distinguish objects and the road in an image [75]. The computed parameters

are used in the real-time image processing procedures so as to obtain a top-down "bird's eye" view, as shown in Figure 5-5, from which information about the path in front of the robot can be extracted.

IPM allows compensating for the well-known problem of *perspective distortion* transforming all pixels from an acquired image into a new representation, where information is stored and homogeneously distributed [61, 74]. IPM needs to know position and heading of the camera and its parameters, which are calculated in the calibration phase.

Being this kind of transformation, shown in Figure 5-6, a remapping of points from the image plane into the road plane, IPM can be classified as a particular type of planar homography. Supposing the road in front of the camera to be flat, IPM allows obtaining a top-down view, or bird's eye view, of the scene seen by the camera. The algorithm for road detection and obstacle detection shown in this work makes use of bird's eye view.

Kernel Matrix Generation

The *Road Detection* algorithm should be able to detect a suitable path for the robot motion without any a-priori knowledge of the environment. The robot could move on asphalted or paved street, on off-road track with several grades and types of grass, and so on. It is really hard to define robust classes identifying road types by any type of features such as geometry or color. Sometimes, during in-field navigation, road characteristics may change, increasing the difficulty to correctly identify drivable surfaces.

The core of the *Drivable Surface Detection Algorithm* (DSD) is the *Kernel Matrix Generation* Task (KMG): this assumes that the robot starts its motion from a drivable surface, thus allowing it to characterize this

Figure 5-5 Bird's eye view example: Original image (left) Transformed image (right).

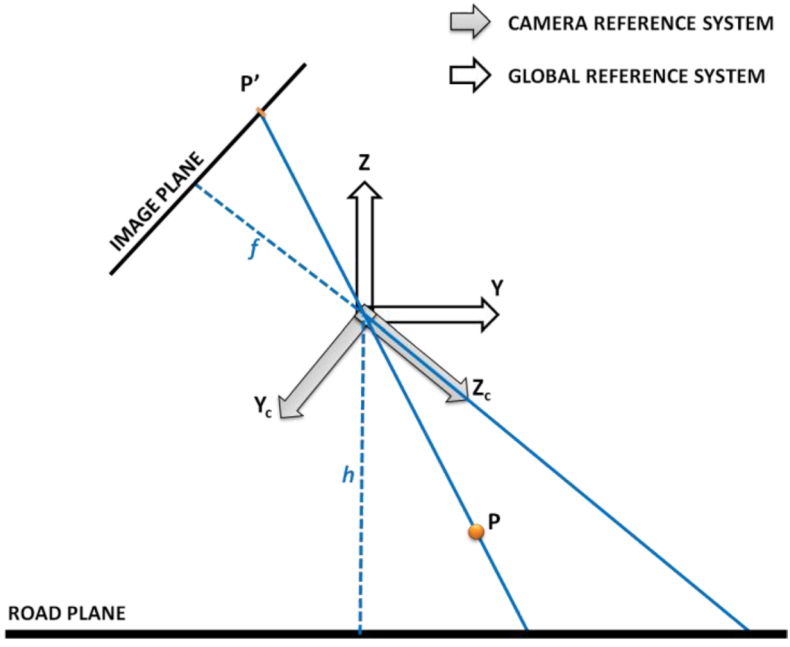


Figure 5-6 Coordinate systems representation for IPM.

drivable surface by means of features of the near-robot region. Features involved in this step are the RGB components of a particular *Region Of Interest* (ROI), shown in Figure 5-7, which is a rectangular portion of the bird's eye image of the scene in front of the robot. This ROI will be referred to as *Kernel*, and it is 120 pixels wide and 80 pixels high.

The Kernel is used to build the Kernel Matrix (KM), which is a 3D Matrix containing information regarding Kernel colors. The way the KM is defined allows associating each of its elements, defined by three coordinates, to one three-element vector representing RGB components of a pixel. The KMG process works as follows:

1. Each pixel of the Kernel is defined by its RGB components, expressed by the vector [r, g, b];
2. Each pixel in the Kernel increments by an amount D the value of the corresponding element in the KM, which is KM (r, g, b), and by a smaller amount d, ones nearby.

Because the Kernel should usually have homogeneous characteristics, the kernel matrix will show well-defined clusters.

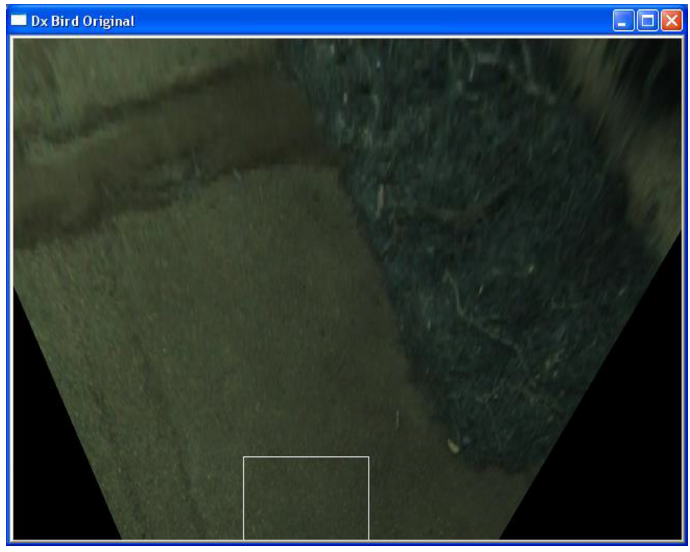


Figure 5-7 Kernel ROI in the bird's eye view.

Image processing

After the KM is built, each pixel of the whole image is analyzed in order to estimate if its color is present in the constructed KM, and so in the kernel. This step allows defining drivable surface. The process works as follows:

1. Each pixel of the image, by means of its RGB components, defines a vector [r, g, b];
2. The above mentioned vector identifies an element of the KM within a specific value;
3. The greater this value, the greater the probability that the evaluated pixel belongs to a drivable surface. Several thresholds have been experimentally tested to correctly classify pixels into distinct categories.

Initially the identification step was performed using one threshold, as shown in Figures 5-10 and 5-11 in the next section, while the final version makes use of two thresholds creating three categories. Figure 5-8 shows the result of this kind of classification, where three colors identify the following regions:

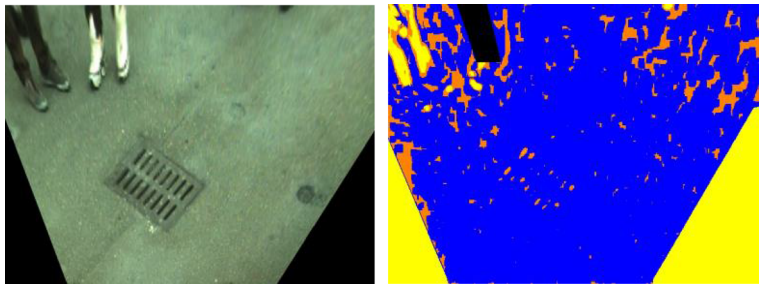


Figure 5-8 DSD example with two thresholds: original birds' eye image (left) and output image (right).

- Blue pixels represent Fully Drivable Surfaces
- Yellow ones represent Un-Drivable Surfaces
- Orange ones are in between the previous two cases
- Black pixels represent Obstacles

5.5.3 Drivable surface detection setup

The Robovolc and U-Go robots have been used to test and validate the proposed algorithm. The two robots were equipped with high quality stereo cameras with a resolution of 1.3 Megapixels and fixed-focus lens of 4.0 mm [76]. The images coming from the two cameras were synchronized with an 8 kHz clock generated by using an IEEE1394 interface. The image processing is based on the OpenCV open source software library [73].

Several experimental tests have been performed to refine the proposed algorithm in order to improve its performances and the ability to identify drivable surfaces.

Images from real navigation scenarios can be very noisy and affected by several disturbing elements. One problem could be the presence of non-drivable objects inside the ROI. For instance, if the robot is moving on grass, a small stone, viewed inside the ROI, may influence the kernel matrix, thus altering the way the rest of the picture is classified.

Moreover, as usual for computer vision algorithms, and in particular in outdoor applications, scene lighting is the greater problem to be faced. Various strategies have been used to ensure that the effect of any changes in brightness or small shadows could be reduced. In particular

three techniques have been developed and tested with the aim of reducing the effect of disturbances and noise:

1. Macro-pixels
2. Morphology operators
3. Normalization

Macro-pixels

In order to reduce the effect of irregular pixels, with respect to the drivable surface, inside the ROI or inside the rest of the image, *Macro-pixels* have been introduced. A Macro-pixel is defined as a region of 10×10 pixels. The mean RGB value, among pixels forming the macro-pixel, defines its RGB components. These new RGB components and the associated vector are both used to process the kernel, during the Kernel Matrix Generation stage and during the image processing step. Each macro-pixel acts both on the KMG step and on the image processing step the same way pixels do by means of their RGB components.

Macro-pixels have been used as moving windows of 10×10 pixels; so after a macro-pixel is processed, the next one is obtained shifting this window one pixel beyond.

Four modalities have been analyzed:

- Pixel to pixel
- Macro to pixel
- Pixel to macro
- Macro to macro

Comparative results of these possible modalities, applied to the road image shown in Figure 5-9, are reported in Figures 5-10 and 5-11.

Morphology operators

Morphology operators have been tested in order to remove irregularities in the image. Before the KMG step, combinations of the *Opening* and *Erosion* operators have been applied to the bird's eye image. These operators are provided by the OpenCV framework. The trade-off between the obtained benefits and the increased execution time, applying morphology operators, did not appear to be valuable, so morphology operators have been discarded.

Figure 5-9 Original bird's eye image used to test macro-pixels.

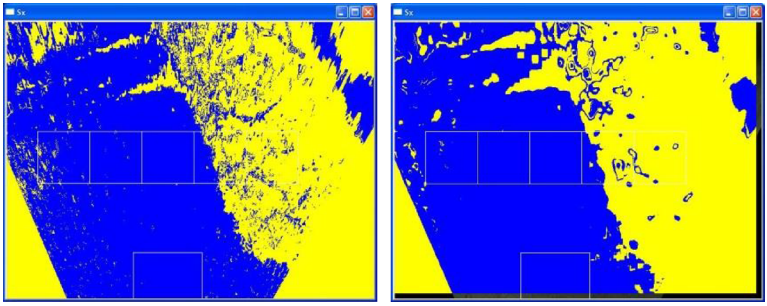


Figure 5-10 Transformation involving macro-pixels: Pixel to Pixel (left) and Pixel to Macro (right).

Normalization

Another operation performed after the image acquisition step is a normalization of the image. This normalization stage is applied to the RGB components of the pixels, defined by the vector $[r, g, b]$, and works as reported in (4). Each component is divided by the sum of all of them and then multiplied for a scale factor L representing a desired color level.

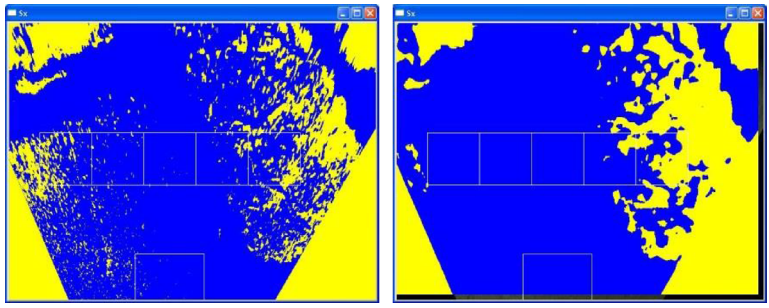


Figure 5-11 Transformation involving macro-pixels: Macro to Pixel (left) and Macro to Macro (right).

$$r' = \frac{r}{r+g+b} L \quad g' = \frac{g}{r+g+b} L \quad b' = \frac{b}{r+g+b} \qquad (4)$$

The new RGB components are defined by the vector $[r', g', b']$. This stage reduces the number of color levels and make the components complementary; this means that $r' + g' + b' = L$. Tests have analyzed two values for the scale factor, $L = 32$ and $L = 64$. In this stage a quantization is also performed, because integer values of the new RGB components are adopted. Thus they may assume integer values ranging from zero to L.

Moreover this normalization stage also allows for reducing the complexity of the kernel matrix that became a 2D matrix, instead of 3D, thus reducing the general computational cost. The components that have been used to construct the 2D kernel matrix are the green and the red ones.

Setup and optimization

Once several tests were performed it was possible to make some considerations.

Mixed techniques, such as pixel to macro or macro to pixel, have been discarded because they don't use the same method to process the image and the ROI. Both these techniques involve two stages: one working with pixels and the other using macro-pixels. As a result the noise onto images of the road is usually increased.

As was already mentioned, the use of morphological operators has been discarded too, because they increase the computational effort with a negligible reduction of noise on the output.

Pixel to pixel and macro to macro are good techniques, especially if combined with normalization, because a small loss of information is compensated by a smaller sensitivity to variations in brightness. Finally normalization makes the algorithm more robust.

Once the global behavior of the algorithms has been defined, in their implementation several code optimization techniques have been also used, in order to reduce the computational cost.

First a memory access management optimization, to speed up all the image processing algorithms, has been carried out. Another improvement was obtained by a suitable management of macro-pixels. As was said before, the mean of the components of the macro-pixels is used to generate the kernel matrix; to do so one hundred pixel components should be manipulated for each step and for each macro-pixel. To better explain this stage, consider a row of pixels, as shown in Figure 5-12, and a 1×3 macro-pixel.

The sum $S(1)$ of the components of the first macro pixel, the red rectangle represented by the pointer ptr(1), is $S(1) = A + B + C$ while $S(2)$. The sum for the second macro-pixel, the green rectangle represented by the pointer ptr(2), can be expressed with $S(2) = S(1) - A + D$. In this way the sum is iteratively evaluated. If this example is extended to greater macro-pixels, it's easy to understand that the greater the dimensions of the macro-pixels are, the more useful this technique is. Working with 10×10 macro-pixels, one hundred sum operations must be performed for the first macro-pixel, while for the following only ten sums and ten subtraction operations are needed. The navigation system, as it

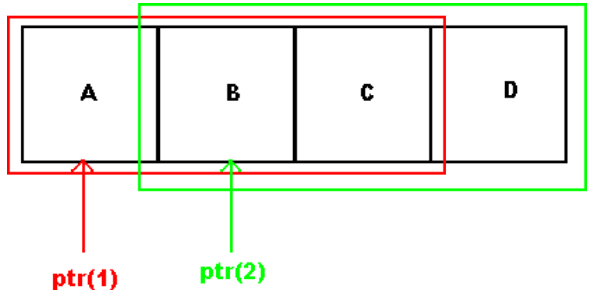


Figure 5-12 Macro-pixels mean calculation example.

will be presented later on, makes use of a portion of the image only, so the execution time is furthermore reduced.

Table 5-1 shows the execution times of the proposed DSD algorithm for the possible configurations discussed before and, in the last row, for the final optimized solution. Reported values are the average values of several tests involving different navigation scenarios. Tests have been performed on a laptop PC with a Pentium M @ 1.73 GHz and 1.5 GB of Ram.

The first row of Table 5-1 reports the fastest configuration which is related to the pixel-to-pixel image processing without normalization. Although this is the fastest configuration, the last one has been chosen. In fact the adoption of a macro-pixel with 64 color levels has demonstrated a greater immunity to small irregularities present on road images, and the optimization has led to a faster execution time.

On the basis of all these considerations, the technique chosen for the road detection algorithm works as follows:

- Builds the normalized image
- Adopts 10×10 macro-pixels to scan the image and the ROI

Table 5-1 Comparison of execution time of the DSD algorithm with different configurations.

Configuration	Execution time (ms)
Pixel to Pixel	**9,589**
Pixel to Macro	358,170
Macro to Pixel	**20,661**
Macro to Macro	381,090
Pixel to Pixel - 32 levels normalization	327,424
Pixel to Macro - 32 levels normalization	2,426,799
Macro to Pixel - 32 levels normalization	485,652
Macro to Macro - 32 levels normalization	3,454,011
Pixel to Pixel 64 - levels normalization	356,452
Pixel to Macro 64 - levels normalization	2,580,387
Macro to Pixel 64 - levels normalization	498,217
Macro to Macro 64 - levels normalization	3,764,042
Macro to Macro 64 - levels Optimized	**15,219**

- Builds a 2D Kernel matrix with $L = 64$, $D = 8$, $d = 1$
- Uses two thresholds, 1 and 2000 occurrences, for the DSD

5.5.4 Obstacle detection

As previously introduced, the adoption of stereo cameras allows for detecting obstacles and getting elevation data out from the images, in order to find drivable corridors. To do so, both left and right bird's eye images are used to create an "error image."

The error image, as shown in Figure 5-13, is the difference between left and right bird's eye images. As can be seen all those pixels belonging to the road plane produce darker pixels in the error image because those pixels in stereoscopic images, transformed with the homography, look similar. Otherwise, objects rising from the road are seen in a different way from the two stereo cameras, so the error image presents clearer areas especially along the sides of objects, because these regions, being seen by a single camera, cannot be superimposed.

Thus, in presence of obstacles, non-zero pixels are present in the error image; non-zero means brighter pixels. Obstacle borders can be so identified by means of the clearer triangular regions appearing in correspondence of their edges.

The obstacle detection algorithm uses a polar histogram to scan the error image, with respect to a focus point F placed on the horizontal plane, in order to locate triangles generated in the presence of obstacles. The histogram is constructed adding pixel values along the lines in the overlapping regions.

As shown in Figure 5-14, the histogram presents a certain level of noise that can be attenuated through the use of a low pass filter. A

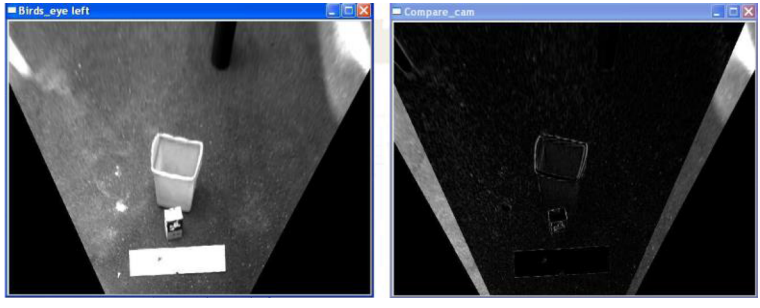


Figure 5-13 A bird's eye image (left) and error image (right).

threshold has been used to detect peaks in the histograms, which define obstacles borders.

Once the position of the borders of an obstacle is found, it is useful to calculate the distance of the obstacle from the camera. A cumulative histogram, along the directions of the obstacle's edges, is evaluated; even in this case a low pass filter has reduced the noise.

An example is reported in Figure 5-15, where the horizontal axis corresponds to the *y* coordinate of the pixel within the considered line. All these histograms start with a higher slope and end with a lower slope (near the robot). The breaking point, where the slope changes, corresponds to the contact point between the obstacle and the horizontal plane, thus providing the distance of the obstacle by the camera. Even in this case, evaluating the histogram slope, a threshold is used to detect the exact location of the obstacle. Some results are shown in Figures 5-16 and in 5-8.

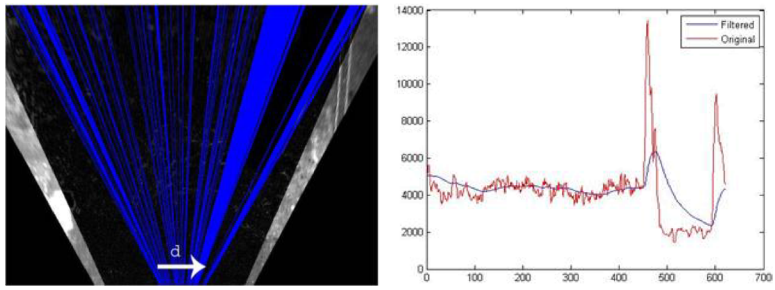


Figure 5-14 Polar histogram creation (left) and filtered and unfiltered output (right).

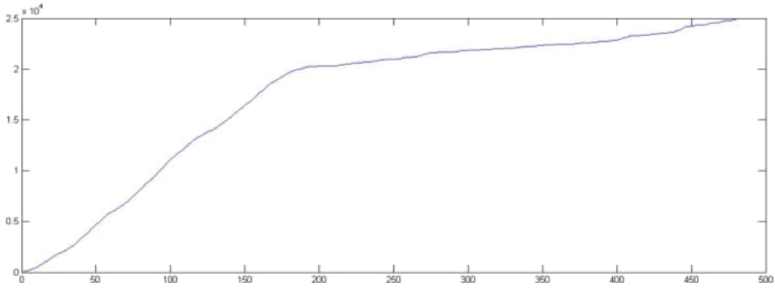


Figure 5-15 Cumulative histogram example, the right side is nearer the robot.

Figure 5-16(a) shows the image acquired by the stereo cameras while Figure 5-16(b) shows the relative bird's eye transformation. Figure 5-16(c) reports the error image. Figure 5-16(d) shows the output of the whole algorithm; black regions appear in correspondence to the obstacle placed in front of the robot.

5.5.5 Control of the robotic platform

Navigation algorithms have been implemented using the Microsoft Robotics Developer Studio (MRDS) tool [77]. MRDS is particularly useful for moving a project from simulation toward real robot implementation. It is in fact possible to replace each simulation entity with a corresponding tool of the real world.

The control module, therefore, receives and processes sensor information. Data are, first of all, processed by an Extended Kalman Filter in order to evaluate robot position and heading. The navigation algorithm takes care of generating the control reference for the robot, by means of the Potential Field Method, allowing it to avoid collisions with obstacles detected by sensors [56, 78].

The Potential Field Method (PFM) is a classical method that allows software development and operation with high modularity because,

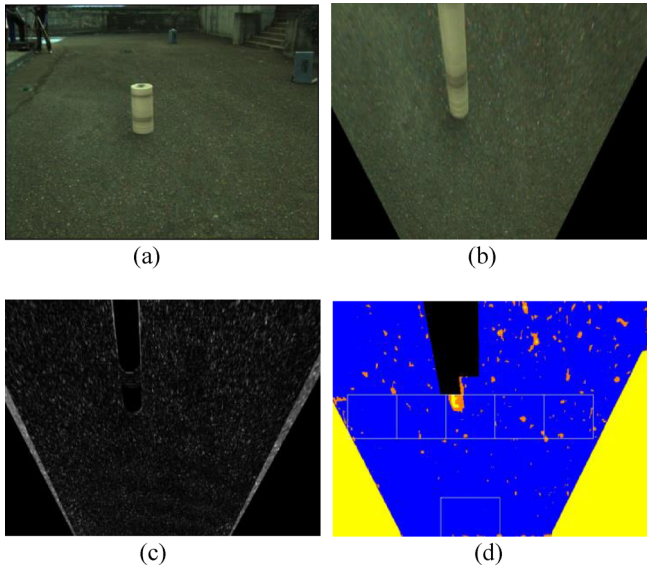


Figure 5-16 Obstacle detection stages: a) original image; b) bird's eye view; c) error image; d) output image.

being that sensors transform information into virtual forces acting on the robot, the presence or the absence and even the fault of a sensor doesn't affect the global control algorithm behavior. In most of the considered real cases the robot had to follow several given waypoints using a DGPS-based localization system, and at the same time following a drivable surface and avoiding obstacles. The result of this technique is a sequence of movements that lead to safely driving the vehicle towards the target without collisions.

Figure 5-17 shows the five squared portions of the image that, during normal operations, are the only areas processed in order to control the robot. Each of the five squares generates a virtual force applied to the robot. Equation (5) reports how the module of the attractive force generated by drivable surface is evaluated:

$$F_{Ai} = \frac{W_B \cdot n_B + W_0 \cdot n_0}{n} \tag{5}$$

where W_B and W_O are weight coefficients for the blue and orange pixels respectively, while n, n_B and n_O are the total number of pixels inside the square and the number of blue and orange pixels respectively. W_B and W_O are set to 1 and 0.5 respectively.

$$F_{Ri} = \frac{W_D \cdot n_D}{n} \tag{6}$$

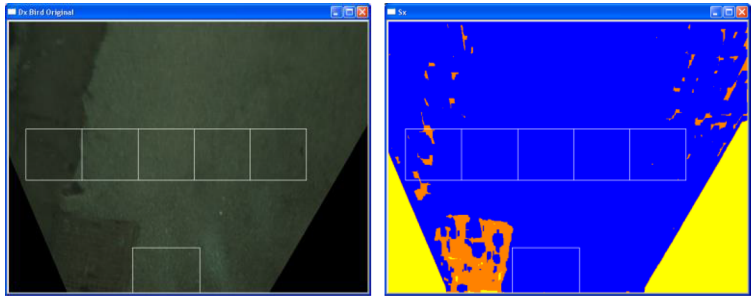


Figure 5-17 Control squares used to evaluate virtual forces.

Equation (6) reports how repulsive forces, generated from obstacles seen inside each square, are evaluated. W_D and n_D are the weight coefficients and the number of black pixels respectively; W_D is set to 1.

Figure 5-18 outlines, in a simplified simulated environment, how information coming from the computer vision algorithm is transformed into virtual forces, suitable to be used for the PFM. The small blue triangle represents the robot, while the above rectangle is the bird's eye image of the road. The gray rays represent a discretization of the image into sectors. The gray and black stars are obstacles which apply repulsive forces (red arrows) on the robot while small attractive forces are also generated by drivable surfaces (cyan arrows).

To control the robot the total force F_T must be evaluated as reported by (7):

$$F_T = \sum_i K_{A,i} F_{A,i} - \sum_i K_{R,i} F_{R,i} \tag{7}$$

where $K_{A,i}$ and $K_{R,i}$ are weighting vectors.

The lower control loops maintain the linear velocity constant and act on the angular speed command ω through the following steps:

$$\vartheta_{FT} = atan2(F_{Ty}, F_{Tx}) \tag{8}$$

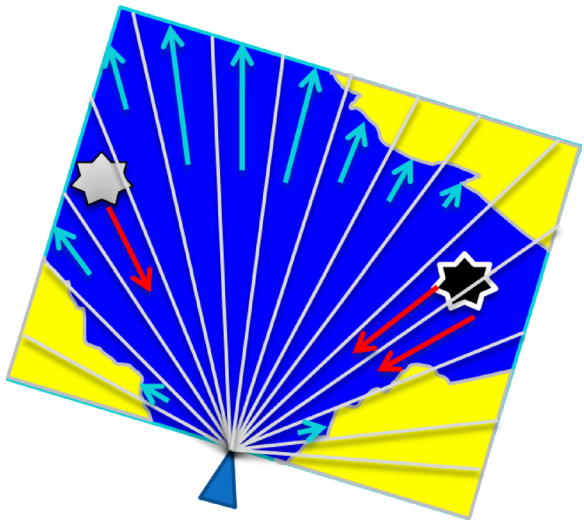


Figure 5-18 A representation of virtual forces generation.

$$\vartheta_E = \vartheta_{FT} - \vartheta_R \tag{9}$$

$$\omega = K_\omega \times \vartheta_E \tag{10}$$

Equation (8) is used to evaluate the orientation θ_{FT} of the total force acting on the robot; equation (9) is used to evaluate the error θ_E between θ_{FT} and the robot orientation expressed by θ_R. Errors greater than 180° are suitably reversed, as usual. The error is used to directly control the robot by means of a simple proportional action, as exposed by (10).

The five control squares, shown in Figure 5-17, used to evaluate virtual forces by means of equations (5)–(7), have been moved farther from the robot following the results of the initial experimental trials. Figure 5-19 shows the initial position on the left and the final position on the right.

Figure 5-20 shows a test with this new configuration while Figure 5-21 shows an image, generated off-line, where the homography, introduced by equations (1)–(3), is used to impose the Drivable Surface image onto the real image.

Several tests have been performed in outdoor navigation. Figure 5-22 shows a 140 meter path navigation test, where the robot has moved from the starting point on the left, has reached a waypoint represented by the end of the path on the right, and has come back to the starting position avoiding the obstacle detected with the proposed algorithm.

Figure 5-23 shows some off-line generated pictures of the test. Figure 5-23(a) has been acquired and processed at the starting position. As can be seen, there is an initial offset between the robot orientation and the road; indeed the grass on the left side is marked in orange and yellow

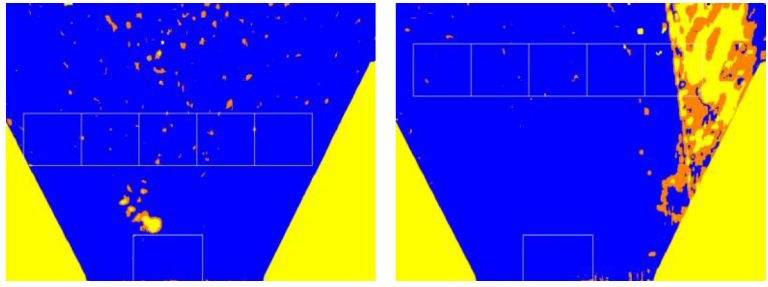


Figure 5-19 Squared Regions positions: before (left) and after optimization (right).

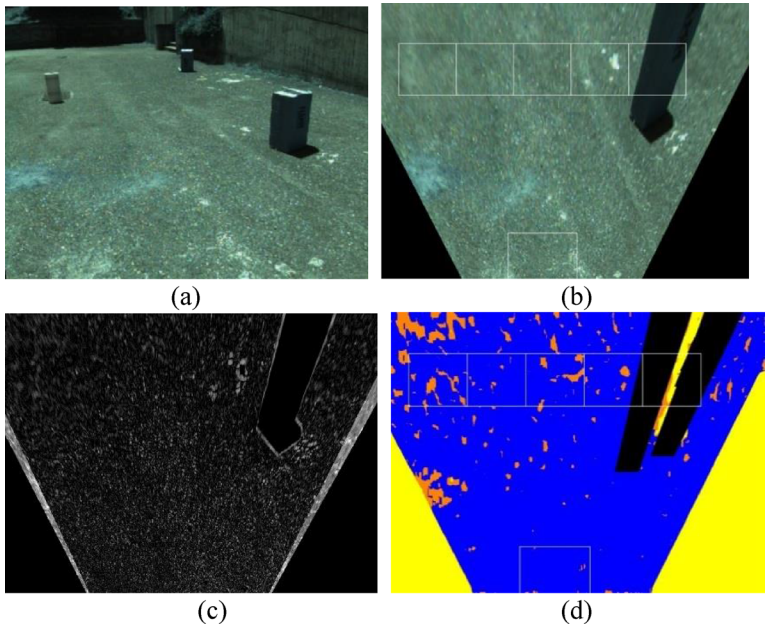


Figure 5-20 Experimental test: a) original image; b) bird's eye view; c) error image; d) output image.

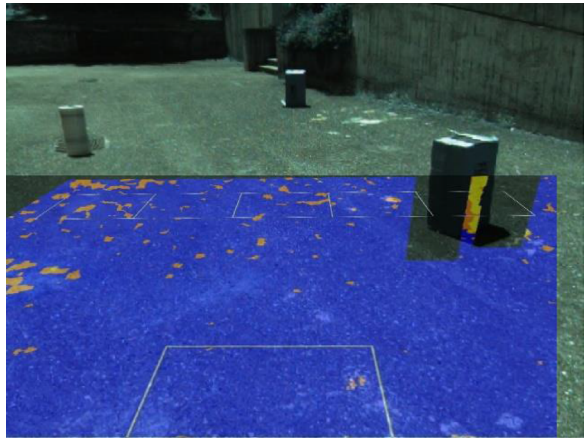


Figure 5-21 Off-line generated image of an experimental test.

Figure 5-22 Map of a navigation test: Copyright © 2012 Google (Google Earth) © 2012 Tele Atlas Image © 2012 GeoEye © Maplink/Tele Atlas.

because it is correctly seen as different from the sandy road inside the kernel. Figure 5-23(b) was acquired once the robot moved to its destination waypoint; it has correctly turned and moved towards the center of the road.

This initial path is slightly descendent. At the end of the slope there is a van, shown in Figure 5-23(c); even if there is some grass, the shadow of the vehicle is seen as undrivable surface before the vehicle is detected as obstacle. The results is the robot is forced to turn left.

In Figure 5-23(d) again the grass on the border is detected as undrivable surface while the plastic net on the right is detected as obstacle, so a darker region appears in the image. The robot has turned again left as shown in Figure 5-23(e).

After the robot reached its waypoint, it had to move back to the initial position, so in Figure 5-23(f) the plastic net and the van are viewed again. The net is seen again as an obstacle in Figure 5-23(g). Figures 5-23(h) to (j) report three other images acquired while the robot is going back.

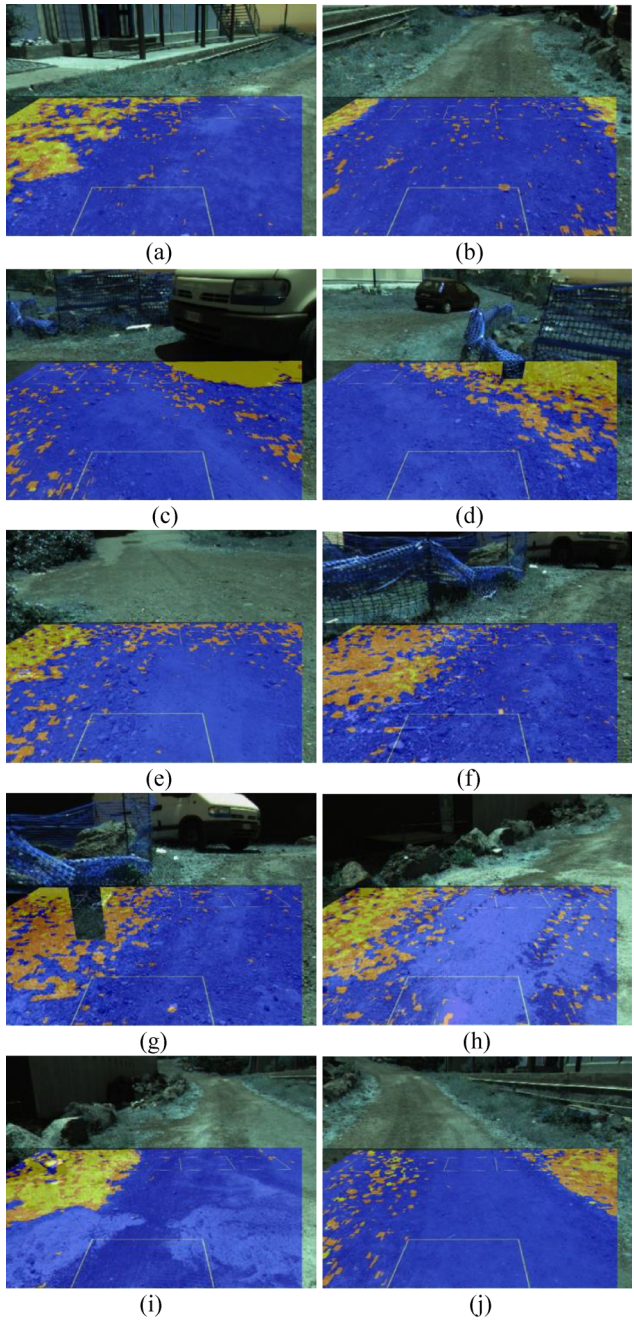


Figure 5-23 Time sequence of peculiar off-line generated pictures from a navigation test.

5.5.6 Final considerations

A new computer vision-based method suitable of performing road detection and obstacle avoidance was presented in this section. The proposed framework relies on fast and well-optimized algorithms, reaching a high performance that allowed real-time implementation with simple computer hardware. It has been tested in several on- and off-road scenarios, showing strong abilities in road scene understanding and obstacle avoidance. A preliminary implementation of this method was employed in the robot that participated in the ELROB 2010 contest [60].

The whole navigation system adopted integrates multiple sensor information, but the results presented here highlight how this framework has been used in controlling autonomous vehicles, reaching good results.

5.6 Conclusions and future work

The proposed software framework copes with two important tasks of robotics, road detection and obstacle avoidance, by means of the adoption of stereo cameras.

The robustness of the adopted algorithms has been deeply verified with several tests: drivable surfaces detection has been tested even in complex unpaved and off-road terrain conditions, while the obstacle avoidance algorithm has also shown the ability to detect moving objects within the surrounding environment. The drivable surface detection avoids the use of stationary or pre-processed road models, allowing it to work with changing environments; this guarantees robustness and reliability even in complex outdoor environments.

All the information coming from image processing algorithms is managed by the navigation system based on the potential field method which manages paths classified as road as attractive forces and, otherwise, detects obstacles as repulsive forces.

The navigation system, tested using the two robotic platforms presented above, has had good experimental results in terms of the robustness achieved in autonomous navigation.

Future work will deal with several aspects. The reliability could be increased by integrating information coming from other sensors, such as sonars or laser range finders. Even if the road detection algorithm has shown good performance, its robustness will be verified by testing the proposed image processing algorithms with harder and more variable road conditions.

5.7 References

[1] Moorehead S. J., "Autonomous surface exploration for mobile robots", 2001, PhD Thesis, Carnegie Mellon University.

[2] Bellingham, J. G., Rajan, K., "Robotics in remote and hostile environments", *Science*, 318.5853, 1098–1102, 2007.

[3] Muscato G., Bonaccorso F., Cantelli L., Longo D., Melita C. D., "Volcanic Environments: Robots for Exploration and Measurement," *Robotics & Automation Magazine, IEEE*, **19**(1), 40–49, March 2012.

[4] Guccione S., Muscato G., Nunnari G., Virk G. S., Azad A. K. M., Semerano A., Ghrissi M., White T., Glazebrook C., "Robots for volcanoes: The state of the art" in *Proc. 3rd Int. Conf. Climbing and Walking Robots (CLAWAR)*, Madrid, Spain, Oct. 2–4, 777–788, 2000.

[5] Aranzulla P., Bonaccorso F., Bruno C., Cantelli L., Lanteri G., Longo D., Muscato G., Pennisi A., Prestifilippo M., "An innovative autonomous outdoor vehicle based on Microsoft Robotic Studio," in *Proc. CLAWAR: 13th Int. Conf. Climbing and Walking Robots and the Support Technologies for Mobile Machines*. Nagoya, Japan: World Scientific Publishing, Aug. 31–Sept. 3, pp. 97–104, 2010.

[6] Bonaccorso F., Cantelli L., Longo D., Melita D., Muscato G., Prestifilippo M., "The U-Go Robot, a multifunction rough terrain outdoor tracked vehicle for R&D on autonomous navigation algorithms," in *Proc. 5th IARP Workshop Robots for Risky Interventions and Environmental Surveillance-Maintenance*, Leuven, Belgium, June 20–22, 2011.

[7] Caltabiano D., Muscato G., "A Robotic System for Volcano Exploration", pp. 499–519, in *Cutting Edge Robotics, Advanced Robotic Systems Scientific Book* July 2005, ISBN:3-86611-038-3.

[8] Bonaccorso F., Longo D., Muscato G., "The U-Go Robot, a multifunction rough terrain outdoor tracked vehicle," *Proceedings of the 2012 World Automation Congress WAC2012*, Puerto Vallarta, Mexico.

[9] Bares J., Wettergreen D., "Dante II: Technical Description, Results and Lessons Learned," *The International Journal of Robotics Research*, 621–649, July 1999.

[10] Nagatani K., Akiyama K., Yamauchi G., Otsuka H., Nakamura T., Kiribayashi S., Yoshida K., Hada Y., Yuta S., Fujino K., Izu T., Mackay R., "Volcanic Ash Observation in Active Volcano Areas using Teleoperated Mobile Robots – Introduction to Our Robotic-Volcano-Observation Project and Field Experiments," *Proceedings of the 11th International Symposium on Safety Security and Rescue Robotics*, Oct 21–26, Linkoping, Sweden, 2013.

[11] Giudice G., Longo D., Melita C. D., Muscato G., "Measurement and exploration in Volcanic environments," in *Handbook of Unmanned Aerial Vehicles*, K. P. Valavanis, G. J. Vachtsevanos (Eds.), Springer, 2014.

[12] Moorehead S. J., Simmons R., Apostolopoulos D., Whittaker W., "Autonomous Navigation Field Results of a Planetary Analog Robot in Antarctica," *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Noordwijk, Netherlands, 1999.

[13] Apostolopoulos D. S., Wagner M. D., Shamah B. N., Pedersen L., Shillcutt K., Whittaker, W. L., "Technology and field demonstration of robotic search for Antarctic meteorites," *The International Journal of Robotics Research*, **19**(11), 1015–1032, 2000.

[14] Murphy R. B., Tadokoro S., Nardi D., Jacoff A., Fiorini P., Choset H., Erkmen A. M., "Search and Rescue Robotics," Chapter 50 in *Springer Handbook of Robotics*, Siciliano and Khatib editors, Springer 2008.

[15] Qian H., Xu Y., Xu W., "The state-of-arts and key technologies for telerobotics in nuclear applications," *32nd IEEE Chinese Control Conference (CCC), 2013*, pp. 6016–6021, July 2013.

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

[17] Nagatani K., Kiribayashi S., Okada Y., Otake K., Yoshida K., Tadokoro S., ..., Kawatsuma, S., "Emergency response to the nuclear accident at the Fukushima Daiichi Nuclear Power Plants using mobile rescue robots," *Journal of Field Robotics*, **30**(1), 44–63, 2013.

[18] Kawatsuma S., Fukushima M., Okada T., "Emergency response by robots to Fukushima-Daiichi accident: Summary and lessons learned," *Industrial Robot: An International Journal*, 39.5, 428–435, 2012.

[19] ROBOCUP Web site http://www.robocup.org/.

[20] Nicoud J.-D., "Vehicles and robots for humanitarian demining," *Industrial Robot: An International Journal*, 24.2, 164–168, 1997.

[21] Baudoin Y., Colon E., "Humanitarian demining and robotics," *Proceedings of the 1998 IEEE International Conference on Control Applications*, Vol. **1**, IEEE, 1998.

[22] Santana P. F., Barata J., Correia L., "Sustainable robots for humanitarian demining," *International Journal of Advanced Robotic Systems*, 4.2, 207–218, 2007.

[23] Habib M. K., "Humanitarian demining: Reality and the challenge of technology-the state of the arts," *International Journal of Advanced Robotic Systems*, 4.2, 151–172, 2007.

[24] Tojo, Y., Debenest, P., Fukushima, E. F., Hirose, S., "Robotic system for humanitarian demining," *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, Vol. **2**, pp. 2025–2030, April 26–May 1, 2004.

[25] Bogue R., "Detecting mines and IEDs: What are the prospects for robots?," *Industrial Robot: An International Journal*, 38.5, 456–460, 2011.

[26] Furuta K., Ishikawa J., eds. *Anti-personnel Landmine Detection for Humanitarian Demining: The Current Situation and Future Direction for Japanese Research and Development*, Springer, 2009.

[27] TIRAMISU Project Website http://www.fp7-tiramisu.eu/.

[28] Morris A. C., *Robotic Introspection for Exploration and Mapping of Subterranean Environments*. ProQuest, 2007.

[29] Omori H., Nakamura T., Yada T., "An underground explorer robot based on peristaltic crawling of earthworms," *Industrial Robot: An International Journal* 36.4, 358–364, 2009.

[30] Shaffer G., Stentz A., "A robotic system for underground coal mining," *Robotics and Automation, 1992 Proceedings, 1992 IEEE International Conference on*, Vol. **1**, pp. 633–638, 12–14 May 1992.

[31] Corke P., Roberts J., Cunningham J., Hainsworth, D., Mining robotics. *Springer Handbook of Robotics*, 1127–1150, 2008.

[32] Glass B. J., Cannon H., Stoker C., Davis, K., "Robotic and human-tended collaborative drilling automation for subsurface exploration." *In Proc. International Astronautical Congress*, October 2005.

[33] Shillcutt K., "Solar Based Navigation for Robotic Explorers," Ph.D. thesis, CMU-RI-TR-00-25, October 2000.

[34] Urmson C. P., Dias M., Simmons B., Reid G., "Stereo vision based navigation for sun-synchronous exploration." In: *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on. IEEE* pp. 805–810, 2002.

[35] Wettergreen D., Shamah B., Tompkins P., Whittaker W., "Robotic Planetary Exploration by Sun-Synchronous Navigation," *Proc. International Symposium on Artificial Intelligence and Robotics & Automation in Space*, St-Hubert, Canada, June 2001.

[36] Wettergreen D., Dias B., Shamah B., Teza J., Tompkins P., Urmson C., ..., Whittaker W., "First experiment in sun-synchronous exploration," In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, Vol. **4**, pp. 3501–3507, IEEE, 2002.

[37] Seraji H., "Traversability Index: A new concept for Planetary Rovers, " *Proc. IEEE International Conference on Robotics and Automation*, Detroit, USA, May 1999.

[38] Howard A., Seraji H., Tunstel E., "A rule-based fuzzy traversability index for mobile robot navigation." *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, Vol. **3**. IEEE, 2001.

[39] Seraji H., Howard A., Tuustel E., "Safe navigation on hazardous terrain." In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, Vol. **3**, pp. 3084–3091, IEEE, 2001.

[40] Ugur E., Dogar M. R., Cakmak M., Sahin E., "The learning and use of traversability affordance using range images on a mobile robot," *IEEE International Conference on Robotics and Automation*, pp. 1721–1726, 2007.

[41] Bellutta P., Manduchi R., Matthies L., Owens K., Rankin A., "Terrain perception for DEMO III," *Proceedings of the IEEE Intelligent Vehicles Symposium*, IV, pp. 326–331, 2000.

[42] Kim J., Sun S. M., Oh J. M., Rehg, Bobick A., "Traversability classification using unsupervised on-line visual learning for outdoor robot navigation," in *IEEE Intl. Conf. on Robotics and Automation (ICRA 06)*, (Orlando, FL), May 2006.

[43] Bellone M., Reina G., Giannoccaro N. I., Spedicato L., "3D traversability awareness for rough terrain mobile robots," *Sensor Review*, **34**(2), 220–232, 2014.

[44] Castellanos J. A., Tardos J. D., *Mobile Robot Localization and Map Building: A Multisensor Fusion Approach*. Kluwer Academic Publishers, Norwell, MA, USA, 2000.

[45] Durrant-Whyte H., Bailey T., "Simultaneous localization and mapping: Part I," *Robotics & Automation Magazine, IEEE*, **13**(2), 99–110, June 2006.

[46] Bailey T., Durrant-Whyte H., "Simultaneous localization and mapping (SLAM): Part II," *Robotics & Automation Magazine*, IEEE 13.3, 108–117, 2006.

[47] Adams M., Zhang S., Xie L., "Particle filter based outdoor robot localization using natural features extracted from laser scanners," *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, Vol. **2**, pp. 1493–1498, April 26–May 1, 2004.

[48] Hori Y., "Future Vehicle Driven by Electricity and Control – Research on Four-Wheel-Motored UOT Electric March II,"

IEEE Transactions On Industrial Electronics, **51**(5), 954–962, October 2004.

[49] Reina G., Foglia M., "On the mobility of all-terrain rovers," *Industrial Robot: An International Journal*, **40**(2), pp. 121–131, 2013.

[50] Caltabiano D., Muscato G., "A Comparison Between Different Traction Control Methods for a Field Robot," *International Conference on Intelligent Robots and Systems (IROS2002)*, September 30–October 4, EPFL, Switzerland, 2002.

[51] Iagnemma K., Dubowsky S., "Mobile Robot Rough-Terrain Control (RTC) for Planetary Exploration," *Proceedings of the 26th ASME Biennial Mechanisms and Robotics Conference, DETC 2000*, 2000(b).

[52] Iagnemma K., Dubowsky S., *Mobile Robots in Rough Terrain – Estimation, Motion Planning, and Control with Application to Planetary Rovers*, Springer, pp. 81–96, March 2004.

[53] Siegwart R., Lamon P., Estier T., Lauria M., Piguet R., "Innovative design for wheeled locomotion in rough terrain," *J. Robot. Auton. Syst.*, 40, 151–162, 2003.

[54] Guccione S., Muscato G., "The wheeleg robot," *Robotics & Automation Magazine*, IEEE 10.4, 33–43, 2003.

[55] Siciliano B., Khatib O., Eds., *Springer Handbook of Robotics*. ISBN 978-3-540-30301-5, 2008.

[56] Khatib O., "Commande dynamique dans l'espace operationnel des robots manipulateurs en presence d'obstacles"; PhD thesis, L'Ecole Nationale Superieure de l'Aeronautique et de l'Espace, 1979.

[57] DeSouza G. N., Kak A. C., "Vision for Mobile Robot Navigation: A Survey," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**(2), 237–267, February 2002.

[58] Bertozzi M., Broggi A., Fascioli A., "Vision-based intelligent vehicles: State of the art and perspectives," *Journal of Robotics and Autonomous Systems*, 1–16, 2000.

[59] Defense Advanced Research Projects Agency (DARPA), DARPA Grand Challenge; Online source: http://www.darpa.mil/grandchallenge.

[60] The European Land Robot Trial (ELROB); Online source: http://www.elrob.org.

[61] Muad A. M., Hussain A., Samad S. A., Mustaffa M. M., Majlis, B. Y., "Implementation of inverse perspective mapping algorithm for the development of an automatic lane tracking system," *TENCON 2004. 2004 IEEE Region 10 Conference*, vol. **A**, pp. 207–210, Vol. **1**, 21–24 Nov. 2004.

[62] Taylor C., Malik J., Weber J., "A real-time approach to stereopsis and lane-finding." in *Proc. IEEE Intelligent Vehicles Symposium*, 1996.

[63] Takahashi A., Ninomiya Y., Ohta M., Tange K., "A Robust Lane Detection using Real-time Voting Processor," In *Proc. IEEE ITS*, 1999.

[64] Liu L. J., Ren M. W., Cao Y. L., Yang J., "Color road segmentation for ALV using pyramid architecture" in *Proc. SPIE* Vol. **2028**, pp. 396–404, Applications of Digital Image Processing XVI.

[65] He Y., Wang H., Zhang B., "Color-Based Road Detection in Urban Traffic Scenes" in *IEEE Transactions on Intelligent Transportation Systems*, **5**(4), pp. 309–318, December 2004.

[66] Pomerleau D. A., "ALVINN: An autonomous land vehicle in a neural network," *Advances in Neural Information Processing Systems 1*, Morgan Kaufmann Publishers Inc., San Francisco, CA, 1989.

[67] Hadsell R., Sermanet P., Scoffier M., Erkan A., Kavackuoglu K., Muller U., LeCun Y., "Learning Long-Range Vision for Autonomous Off-Road Driving," *Journal of Field Robotics*, **26**(2), 120–144, February 2009.

[68] Thorpe C., Kanade T., Shafer S. A., "Vision and Navigation for the Carnegie-Mellon Navlab," *Proc. Image Understand Workshop*, pp. 143–152, 1987.

[69] Thorpe C., Herbert M. H., Kanade T., Shafer S. A., "Vision and Navigation for the Carnegie-Mellon Navlab," *IEEE Trans. Pattern Analysis and Machine Intelligence*, **10**(3), 362–372, May 1988.

[70] Fernandez J., Casals A., "Autonomous Navigation in Ill-Structured Outdoor Environments," in *Proc. Int. Conf. Intelligent Robots and Systems*, 1997.

[71] Dahlkamp H., Kaehler A., Stavens D., Thrun S., Bradski G., "Self-supervised monocular road detection in desert terrain," In *Proceedings of the Robotics Science and Systems Conference*, Philadelphia, PA, 2006.

[72] Kriegman D. J., Triendl E., Binford, T. O., "Stereo vision and navigation in buildings for mobile robots," *IEEE Trans. Robotics and Automation*, **5**(6), 792–803, 1989.

[73] Open Source Coputer Vision Library (OpenCV); Online source: http://code.opencv.org/projects/OpenCV/wiki/WikiStart.

[74] Faugeras O., *Three-Dimensional Computer Vision: A Geometric Viewpoint*, The MIT Press, 1993.

[75] Storjohann K., Zielke T., Mallot H. A., von Seelen W., "Visual obstacle detection for automatically guided vehicles," *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, Vol. **2**, pp. 761–766, 13–18 May 1990.

[76] Videredesign, User's manual STH-MDCS3-Var Stereo Head; Online source: http://www.videredesign.com.

[77] Microsoft Robotics Developer Studio; Online source: http://www.microsoft.com/Robotics/.

[78] Tilove R.B., "Local obstacle avoidance for mobile robots based on the method of artificial potentials," *IEEE Int. Conf. Robot. Autom.*, pp. 566–571, 1990.

THIS TITLE IS FROM OUR AUTOMATION AND CONTROL COLLECTION. MORE TITLES THAT MAY BE OF INTEREST INCLUDE…

Flexible Test Automation: A Software Framework for Easily Developing Measurement Applications
By Pasquale Arpaia, Ernesto De Matteis, and Vitaliano Inglese

Measurement and Monitoring
By Vytautas Giniotis and Anthony Hope

Momentum Press is one of the leading book publishers in the field of engineering, mathematics, health, and applied sciences. Momentum Press offers over 30 collections, including Aerospace, Biomedical, Civil, Environmental, Nanomaterials, Geotechnical, and many others.

Momentum Press is actively seeking collection editors as well as authors. For more information about becoming an MP author or collection editor, please visit http://www.momentumpress.net/contact

Announcing Digital Content Crafted by Librarians

Momentum Press offers digital content as authoritative treatments of advanced engineering topics by leaders in their field. Hosted on ebrary, MP provides practitioners, researchers, faculty, and students in engineering, science, and industry with innovative electronic content in sensors and controls engineering, advanced energy engineering, manufacturing, and materials science.

Momentum Press offers library-friendly terms:

- perpetual access for a one-time fee
- no subscriptions or access fees required
- unlimited concurrent usage permitted
- downloadable PDFs provided
- free MARC records included
- free trials

The **Momentum Press** digital library is very affordable, with no obligation to buy in future years.

For more information, please visit **www.momentumpress.net/library** or to set up a trial in the US, please contact **mpsales@globalepress.com.**

Mobile Robots for Dynamic Environments

Emin Faruk Kececi • Marco Ceccarelli

For several decades now, mobile robots have been integral to the development of new robotic systems for new applications, even in nontechnical areas. Mobile robots have already been developed for such uses as industrial automation, medical care, space exploration, demining operations, surveillance, entertainment, museum guides and many other industrial and non-industrial applications. In some cases these products are readily available on the market. A considerable amount of literature is also available; not all of which pertains to technical issues, as listed in the chapters of this book and its companion.

Readers will enjoy this book and its companion and will utilize the knowledge gained with satisfaction and will be assisted by its content in their interdisciplinary work for engineering developments of mobile robots, in both old and new applications. This book and its companion can be used as a graduate level course book or a guide book for the practicing engineer who is working on a specific problem which is described in one of the chapters.

The companion volume for this book, *Designs and Prototypes of Mobile Robots,* is also available from Momentum Press.

MOMENTUM PRESS
ENGINEERING

ISBN: 978-1-60650-821-3
90000
9 781606 508213