Motion Control Algorithms for Mobile Vehicles and Marine Crafts

Iurii Kapitaniuk



university of groningen

The research described in this dissertation has been carried out at the Engineering and Technology Institute Groningen (ENTEG), the Faculty of Science and Engineering, University of Groningen.



This dissertation has been completed in partial fulfillment of the requirements of the Dutch Institute of Systems and Control (DISC) for graduate study.

Printed by ProefschriftMaken
De Bilt, The Netherlands

ISBN (book): 978-94-034-2735-5
ISBN (e-book): 978-94-034-2736-2



university of groningen

Motion Control Algorithms for Mobile Vehicles and Marine Crafts

PhD thesis

to obtain the degree of PhD at the
University of Groningen
on the authority of the
Rector Magnificus Prof. C. Wijmenga
and in accordance with
the decision by the College of Deans.

This thesis will be defended in public on

Friday 29 May 2020 at 12.45 hours

by

Iurii Kapitaniuk

born on 19 July 1988
in Langepas, USSR

Supervisors
Prof. dr. ir. M. Cao
Prof. dr. ir. J.M.A. Scherpen

Assessment committee
Prof. dr. R. Carloni
Prof. dr. ir. B. Jayawardhana
Prof. A. Lanzon

To my family

Acknowledgments

So Long, and Thanks for All the Fish[1]

Douglas Adams

I would like to express my deepest gratitude and appreciation to my advisors Ming Cao and Jacquelien Scherpen for academic guidance and patience during my PhD journey.

I would like to acknowledge all the members of my assessment committee, professor Raffaella Carloni, professor Bayu Jayawardhana and professor Alexander Lanzon, for evaluating the manuscript and returning me valuable comments.

My special admiration goes to Dr. Anton Proskurnikov, without whom this dissertation would not have happened. I appreciate our discussions full of constructive criticism based on your experience and expertise. Also, I am really thankful to my collaborators Dr. Héctor García de Marina Peinado for bringing my theory to practice and Weijia Yao for further development of the guiding vector field framework.

I thank Dr. Alain Govaert and Carlo Cenedese for being my paranymphs. Special thanks to Alain for translating the summary of this thesis into Dutch.

Finally, I would like to thank all my former colleagues and friends at DTPA and SMS groups for providing a friendly atmosphere for the exchange and discussion of ideas during 4 years of my PhD.

[1] It is the message left by the dolphins when they departed Planet Earth just before it was demolished to make way for a hyperspace bypass, as described in The Hitchhiker's Guide to the Galaxy.

Contents

1 Introduction **1**
- 1.1 Modern applications of mobile robots . . . 2
- 1.2 Main contributions and outline of the thesis . . . 7
- 1.3 List of publications . . . 8

2 Path following problem: preliminaries **11**
- 2.1 Unicycle: the basic kinematic model . . . 11
- 2.2 Path following: problem setup . . . 12
- 2.3 Technical assumptions on the desired path . . . 14
- 2.4 Guiding vector field design . . . 15
  - 2.4.1 Properties of the integral curves . . . 16
  - 2.4.2 The GVF and the "ideal" motion of the robot . . . 18

3 GVF-based path following **23**
- 3.1 Overview of path following algorithms . . . 23
- 3.2 The Path Following Controller Design . . . 25
  - 3.2.1 A technical lemma . . . 26
  - 3.2.2 The steering algorithm . . . 26
  - 3.2.3 Local existence and convergence of the solutions . . . 28
  - 3.2.4 An invariant set, free of critical points . . . 31
- 3.3 Discussion and Extensions . . . 34
  - 3.3.1 Comparison with other GVF algorithms . . . 34
  - 3.3.2 The GVF method vs. alternative path following algorithms . . . 35
- 3.4 Conclusions . . . 37

4 The moving path following **39**
- 4.1 Introduction . . . 39
- 4.2 Problem statement . . . 40

4.3 The steering algorithm . . . 43
4.3.1 Design of the vector field . . . 43
4.3.2 Path following controller design . . . 44
4.3.3 Properties of the solutions . . . 45
4.4 Numerical simulations . . . 46
4.4.1 The Cassini oval in the moving frame . . . 46
4.5 Conclusions . . . 48

5 Experimental validation **49**
5.1 Experiments with wheeled robot . . . 49
5.1.1 Circulation along the ellipse . . . 50
5.1.2 Circulation along the Cassini oval . . . 52
5.1.3 Moving path following . . . 54
5.2 Experiments with a fixed wing UAV . . . 55
5.2.1 Sinusoidal path . . . 56
5.2.2 Elliptical path . . . 56
5.3 Experiments with robotic fish . . . 59

6 Roll Damping for Marine Vessels **65**
6.1 Introduction . . . 65
6.2 Mathematical models . . . 67
6.2.1 The vessel's motion . . . 67
6.2.2 The disturbance model . . . 68
6.3 Linear-quadratic optimization in presence of uncertain polyharmonic signals . . . 70
6.3.1 A family of uncertain optimization problems . . . 71
6.3.2 A class of linear OUC . . . 72
6.4 Optimal Universal Roll Damping Controllers . . . 76
6.5 Transfer matrices of the ship-autopilot system . . . 80
6.6 Numerical simulation . . . 81
6.6.1 Wave disturbance . . . 82
6.6.2 The OUC design . . . 82
6.6.3 Comparison with alternative controllers . . . 83
6.6.4 Simulation results . . . 84
6.7 Conclusions . . . 85

7 Conclusions and future research **91**
7.1 Conclusions . . . 91
7.2 Recommendations for future research . . . 92

Bibliography **94**

Summary **105**

Samenvatting **107**

Chapter 1

Introduction

Mobile robots contribute to human, industrial, agricultural, technical, and social life improvements [88]; their positive impact on modern society is continuously increasing. They can have various appearances, dynamics and configurations of actuators: wheels for ground vehicles; legs for walking robots; propellers, waterjets, rudders and fins for marine robots; propellers, reactive engines and wings for aircraft. Nevertheless, many problems of their navigation and motion control can be solved in a unified manner. Nowadays, a vast variety of autopilots, both proprietary industrial-grade solutions and open-source projects [6, 74] are available for many types of robots: multi-copters, traditional helicopters, fixed-wing aircraft, ground rovers and underwater remote operated vehicles. The navigation modules implemented in those autopilots mainly support only very basic functionalities, e.g. traveling with a predefined heading at a prescribed speed. However, many practical applications require to fulfill more complicated control tasks, e.g. obstacle avoidance, monitoring, surveillance or patrolling of a given area, convoying and covert tracking of mobile targets. The relevant applications are summarized in Section 1.1.

Many of the aforementioned problems can be reduced to precise *path following* control. Path following is a classical problem in mobile robotics and control theory; nevertheless, many questions regarding the behavior of the nonlinear path-following algorithms still remain open. The problem becomes especially challenging for underactuated and constrained robots, e.g. a fixed-wing UAV, a high-speed marine craft or a car on a highway, since the class of trajectories a robot can follow is restricted by its dynamics. Another problem of motion control, addressed in this thesis, is active *oscillation damping*. We address this problem in the context of roll stabilization for marine crafts [77]; however, the proposed approaches can also be used for active vibration control of aircraft [76] and automobiles [48]. Oscillation damping is most challenging in situations where the damping controller and the autopilot share the same set of actuators.

1.1 Modern applications of mobile robots

Autonomous mobile robots are involved in numerous industrial, military, scientific and service missions; some of their applications are summarized in this section.

Industrial mobile robots used on a factory shop floor are called automated or autonomous guided vehicles (AGVs) [95]. AGVs are widely used for material handling, product transfer from one place to another, inspection and quality control, etc. All modern manufacturing systems use AGVs for achieving unprecedented level of flexibility and increasing overall efficiency of production processes. The complexity of guidance systems for AGVs mainly depends on the sensory system responsible for robot positioning. If an AGV's localization technology is based on the detection of artificial or natural landmarks, then advanced control approaches are needed to provide precise and reliable navigation of the robot. The further developments in mobile robotics for flexible production process give rise to another kind of robots called mobile manipulators, built from a robotic articulated arm mounted on a movable base (Fig 1.1). These robots combine universality of robotic arms with transportability of mobile platform.



Source: https://commons.wikimedia.org/wiki/File:KUKA_omniRob.jpg by KUKA Laboratories GmbH is licensed under CC BY-SA 3.0 (https://creativecommons.org/licenses/by-sa/3.0)

Figure 1.1: A mobile manipulator used in industrial applications

Addressing the consequences of natural and anthropogenic disasters requires efficient cooperation of robots and humans. Locations of disasters are often dangerous for human intervention or cannot be reached because of extreme temperatures, radioactive levels, strong wind forces and other reasons that do not allow actions of human rescuers. Modern robot rescuers (see e.g. Fig. 1.2 (a)) are light, flexible, and durable. To navigate a rescue robot through the cluttered environment [38], advanced path-following algorithms are needed.

Service robots are designed to assist humans with dirty, dull or repetitive



Source: https://flic.kr/p/a3WKdH by NIST in the public domain

(a) A rescue robot



Source:
https://commons.wikimedia.org/wiki/File:Pepper_the_Robot.jpg
by Softbank Robotics Europe is licensed under CC BY-SA 4.0
(https://creativecommons.org/licenses/by-sa/4.0)

(b) The assistive robot Pepper

Figure 1.2: Rescue and assistive robots

work. Home or domestic robots handle household tasks such as floor cleaning, mowing the lawn and pool maintenance. Mobile robot assistants are developed to serve and actively support disabled people in their daily life to help them live independently(Fig 1.2 (b)).

Mobile robotics has many applications in agriculture (Fig 1.3) and forestry, e.g. smart application of fertilizers and water, auto-guidance on field crop machinery,



Source: https://commons.wikimedia.org/wiki/File:Amazone_BoniRob_Feldroboter-Entwicklungsprojekt.jpg by Amazone GmbH & Co. KG is licensed under CC BY-SA 3.0 (https://creativecommons.org/licenses/by-sa/3.0)

Figure 1.3: An agricultural robot

fruits and vegetables harvesting, tree cutting and log processing. Along with ground robots, unmanned aerial vehicles (UAVs) have recently begun being applied to precision agriculture, e.g. creation of visual maps and spatially attributed data points that are transferable to precision guided machinery.

From an application perspective, one may distinguish between UAVs that operate as remote controlled or semiautonomous systems (typically referred to as drones) and robots that present advanced levels of autonomy. Drones are essentially teleoperated aircraft or systems capable of tracking predefined trajectories while they further integrate on-board sensors to provide situational awareness. Most often such situational awareness is visual (using optics) but can also include meteorological and environmental tasks like hurricane monitoring and chemical plume detection. Although drones will continue to be valuable assets and positively impact both civilian and military missions, nowadays, a constant trend is to develop aerial robots of advanced intelligence [88]. Machine cognition, perception and vehicle control algorithms work in concert to perform applications that go beyond just situational awareness. Reaching new levels of autonomy, these robots are designed to handle unforeseen events, interact with their environment, and adapt to a broad range of scenarios. In essence, drones were in their vast majority passive, providing eyes on the scene whereas modern aerial robots tend to become active, allowing their users to engage the scene, act autonomously and possibly interact with surroundings. UAVs (both drones and aerial robots) have been deployed in remote sensing, disaster response, surveillance, search and rescue, image acquisition, communications, transportation and payload delivery. UAVs are already used for pipeline spotting, power line monitoring, volcanic sampling, mapping, meteorology,

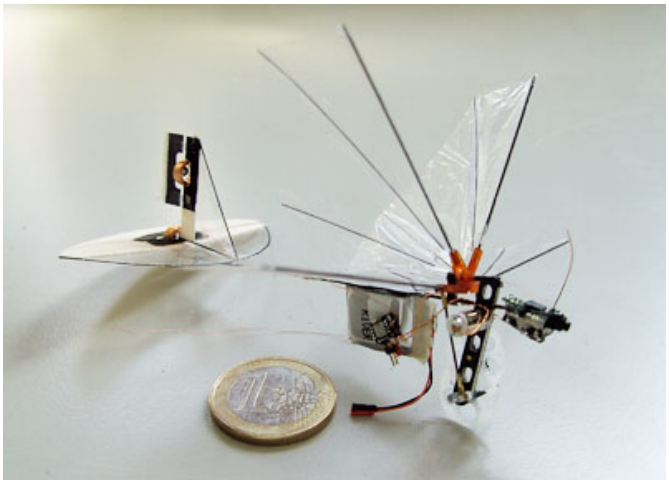
geology, agriculture and unexploded mine detection. Advanced aerial robots will be able to conduct pipeline risk assessment and repair, power line maintenance, real-time mapping, crop care and mine defusing [88].

Underwater robots or unmanned underwater vehicles (UUVs) help to understand marine and other environmental issues, protect the ocean resources of the earth from pollution, and efficiently utilize them for human welfare. Most UUVs commercially available in the market are tethered and remotely operated, so they are referred to as remotely operated vehicles (ROVs). The operators on the mother vessel control the ROV by sending control signals and power to the vehicle and receiving video, status, and other sensory data via an umbilical cable [88]. ROVs have been used for various applications including military operations, environmental and scientific missions, ocean mining, and gas and oil industry [104]. ROVs are required for more inspection, maintenance, drill support, and decommissioning activities than ever before. Autonomous manipulation on a moving base, such as terrestrial mobile robots, humanoids, and underwater robotic vehicles is a very challenging task in the area of robotics in general, especially in unstructured environments, such as underwater. It is defined as the capability of a robot system that performs intervention tasks requiring physical contact with unstructured environments without continuous human supervision. Unlike industrial manipulators which have fixed bases on the floor, autonomous manipulation requires a system capable of assessing a situation, including self-calibration based on sensory information, and executing or revising a course of manipulating action without continuous human intervention. Therefore, developing a system capable of fully autonomous manipulation would be a great achievement and make a substantial impact on a variety of application areas with significant economical, societal, and scientific importance [5].

In addition to the conventional robotic design, there is an actively developing field of biomimetic and bio-inspired technologies. Biomimetics is the use of the biological principles in engineering practice, including robot structure and mechanics, locomotion, perception, and autonomy [88]. Besides providing novel technical solutions, the biomimetic robotics also allows researchers to develop a better understanding of processes in biological systems via reproducing the effects such systems exhibit. In the context of mobile robotics, the research is primarily focused on new locomotion schemes.

Biomimetic aerial robots [21] imitate the flapping-wing flights of birds, bats and some insects (Fig. 1.4). Regarding this kind of locomotion, the main interest of researchers is focused on understanding the aerodynamics in turbulent flows [32]. Progress in this direction opens up the perspective to design miniature flying robots.

Snake-like robots (Fig. 1.5) perfectly fit into concept of the flexible mobile manipulators: the snake-like locomotion provides the great versatility and freedom of movement to access hard-to-reach areas [17, 49]. When a robot has reached its destination, it can fix its body, for instance, wrapping around a rigid cylindrical



Source: https://commons.wikimedia.org/wiki/File:DelFly_Micro_2008_V1.jpg by Cdewagter is licensed under CC BY-SA 3.0 (https://creativecommons.org/licenses/by-sa/3.0)

Figure 1.4: The DelFly flapping wing robot

object; after that, the free part of the body can be used as an articulated robot with numerous degrees of freedom.



Source: https://flic.kr/p/axECr1 by Jiuguang Wang is licensed under CC BY-SA 3.0 (http://creativecommons.org/licenses/by-sa/3.0/us/legalcode)

Figure 1.5: A snake robot

The fish-like swimming robots (Fig. 1.6) make rhythmic waveforms with their bodies to produce propulsion forces. The undulatory movement of fish provides two main advantages – maneuverability in confined areas and high propulsive efficiency. The main difference between the existing propeller and undulatory movement is

turning radius and speed. Fish can turn with a radius 1/10 of their body length, while propeller-driven ships require a much larger radius. Accordingly, the turning speed of fish is much faster than ships. Beyond maneuverability, the driving efficiency in biological swimmers also show more improvement over manmade systems [88, 91].



Figure 1.6: The robotic fish

Currently, the biomimetic robotics primarily uses model-free design approaches. Repeating the behavior of the biological species often enables a robot's designer to solve the locomotion problem, whereas the exact dynamical model of the robot is elusive. However, simple kinematic models (e.g. unicycle) can be implemented to design simple and reliable path following algorithms.

1.2 Main contributions and outline of the thesis

The thesis consists of two parts: path following control algorithms (Chapters 2-5) and roll damping control (Chapter 6).

Chapter 2 introduces some preliminary concepts and results related to path following control problems. In particular, the basic unicycle model and the concept of guiding vector field (GVF) for a general smooth path are introduced. Although the GVF may not be well-defined in some "degenerate" points; however, under natural assumptions (that hold for many practically important trajectories) almost all integral curves of the GVF avoid these exceptional points and converge to the desired trajectory. A robot that ideally follows an integral curve will thus reach the desired path.

In Chapter 3, we propose a GVF-based algorithm for planar path following and prove its basic properties. In particular, we show that the robot reaches the desired trajectory unless it "gets trapped" in a degenerate point where the vector field is undefined. In practice, such a situation is usually exceptional; however, proving its impossibility is only possible when the initial condition belongs to some region [61]. We give explicit estimates of this region. The results are reported in [44] (and, in its preliminary form, in [100]).

Chapter 4 extends the results of Chapter 3 to the problem of moving path

following (path following with respect to a moving reference frame). The results are reported in [42] and [101].

Chapter 5 presents practical experiments illustrating the results of previous chapters. In particular, we implement path following control on a wheeled robot [44, 100], a fixed wing UAV [20] and robotic fish.

Chapter 6 deals with a problem of oscillation damping, specific for marine robotics. For a ship equipped with an autopilot, we design a roll stabilization controller, which guaranties optimal damping of roll oscillations. Optimality is understood in the sense of average "power" of oscillation (the energy averaged over time). The method is based on the theory of optimal universal controllers [80, 81]. The results are reported in [43, 47].

Concluding remarks and some topics that may become interesting for a future research are presented in Chapter 7.

1.3 List of publications

Journal Papers

1. J. Wang, A. Y. Krasnov, Y. A. Kapitanyuk, S. A. Chepinskiy, Y. Chen, and H. Liu, "Path following control algorithms implemented in a mobile robot with omni wheels," Gyroscopy Navigation, vol. 7, no. 4, 2016, pp. 353-359.

2. Y. A. Kapitanyuk, A. V. Proskurnikov, and M. Cao, "A Guiding Vector-Field Algorithm for Path-Following Control of Nonholonomic Mobile Robots," IEEE Trans. Control Syst. Technol., vol. 26, no. 4, 2018, pp. 1372-1385.

3. Y. A. Kapitanyuk, A. V. Proskurnikov, and M. Cao, "Optimal Universal Controllers for Roll Stabilization," Ocean Engineering, Volume 197, 1 February 2020, 106911.

4. Y. A. Kapitanyuk, H. G. D. Marina, A. V. Proskurnikov, and M. Cao, "Guiding vector field algorithm for of a fixed wing UAV," In preparation.

Peer reviewed conference proceedings

1. Y. A. Kapitanyuk, S. A. Chepinskiy, and A. A. Kapitonov, "Geometric path following control of a rigid body based on the stabilization of sets," in Proceedings of IFAC World Congress 2014, IFAC Proceedings Volumes (IFAC-PapersOnline), 2014, vol. 19, no. 3, pp. 7342-7347.

2. J. Wang, I. A. Kapitaniuk, S. A. Chepinskiy, D. Liu, and A. J. Krasnov, "Geometric path following control in a moving frame," in Proceedings of the 1st IFAC

Conference on Modelling, Identification and Control of Nonlinear Systems (MICNON 2015), IFAC-PapersOnLine, vol. 48, no. 11, pp. 150-155, 2015.

3. Y. A. Kapitanyuk, A. V. Proskurnikov, and M. Cao, "Optimal controllers for rudder roll damping with an autopilot in the loop," in the 10th IFAC Conference on Control Applications in Marine Systems (CAMS 2016), IFAC-PapersOnLine, vol. 49, no. 23, pp. 562-567, 2016.

4. H. G. H. G. de Marina, Y. A. Y. A. Kapitanyuk, M. Bronz, G. Hattenberger, and M. Cao, "Guidance algorithm for smooth trajectory tracking of a fixed wing UAV flying in wind flows," in 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 5740-5745, 2017.

5. Y. A. Kapitanyuk, H. G. D. Marina, A. V. Proskurnikov, and M. Cao, "Guiding vector field algorithm for a moving path following problem," in Proceedings of IFAC World Congress 2017, IFAC-PapersOnLine, vol. 50, no. 1, pp. 6983-6988, 2017.

6. W. Yao, Y. A. Kapitanyuk and M. Cao, "Robotic Path Following in 3D Using a Guiding Vector Field," 2018 IEEE Conference on Decision and Control (CDC), Miami Beach, FL, 2018, pp. 4475-4480.

7. L. Chen, M. Cao, H. G. de Marina, Y. Guo and Y. Kapitanyuk, "Triangular formation maneuver using designed mismatched angles," 2019 18th European Control Conference (ECC), Naples, Italy, 2019, pp. 1544-1549.

Conference abstracts

1. I. Kapitaniuk, "Geometric path following control in a moving frame", 34th Benelux Meeting on Systems and Control, 2015, page 26

2. Y.A. Kapitanyuk, A.V. Proskurnikov, M. Cao, "Optimal damping of harmonic signals with unknown spectra", 35th Benelux Meeting on Systems and Control, 2016,, page 63

3. W. Yao, Y. Kapitanyuk, M. Cao, "3D path-following using a guiding vector field", 37th Benelux Meeting on Systems and Control, 2018, page 130

4. W. Yao, Y. A. Kapitanyuk, M. Cao, "High-Dimensional Vector Field for Path Following", 2019, page 115

5. H. G. D. Marina, Y. A. Kapitanyuk, A. Proskunikov, M. Cao, "A guiding vector field algorithm for path following control of drones," 4th Conference of the International Society for Atmospheric Research using Remotely-piloted Aircraft (ISARRA).

Chapter 2
Path following problem: preliminaries

2.1 Unicycle: the basic kinematic model

A widely used technique in path following control is to decompose the controller into an "inner" and an "outer" feedback loop respectively [9, 25, 94] as illustrated in Fig. (2.1). The "inner" *dynamic* controller is responsible for maintaining the vector of generalized velocities (in the planar case, the longitudinal speed and turn rate) by controlling the vector of forces and moments. The design of the dynamic controller is based on the robot's mathematical model and may include rejection of external disturbances, e.g. adaptive drift compensators [27]. When the dynamic controller is sufficiently fast and precise, one may consider the speed and turn rate to be the new control inputs, describing thus the robot with a simpler *kinematic* model. The path following algorithm is typically implemented in the "outer" *kinematic* controller, steering the simplified kinematic model to the prescribed path.

The kinematic model of a mobile robot can be *holonomic* or *nonholonomic*. A holonomic robot is not restricted by its angular orientation and able to move in any direction, as exemplified by helicopters, wheeled robots with omni-directional wheels [88] and fully actuated marine vessels at low speeds [14]. For nonholonomic robots only some directions of motion are possible. The simplest model of this type is the *unicycle*, which can move only in the longitudinal direction while the lateral motion is impossible. Examples of robots that can be reduced under certain

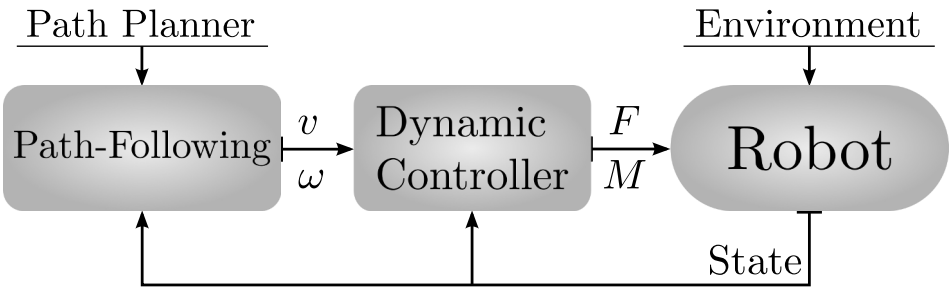


Figure 2.1: The structure of motion control systems

conditions to the unicycle-type kinematics, include differentially driven wheeled mobile robots [66, 85], fixed wing aircraft [9, 53, 67], marine vessels at cruise speeds [25] and car-like vehicles, whose rear wheels are not steerable [3, 66, 85]. The most interesting are unicycle models where the speed is restricted to be sufficiently high, making it impossible to reduce the unicycle to a holonomic model via the feedback linearization [71]. The lift force of a fixed-wing UAV, the rudder's yaw moment of a marine craft and the turn rate of a car-like robot depend on the longitudinal speed; at a low speed their maneuverability is very limited.

Although many guidance problems (e.g. arising in aerial and underwater robotics) deal with three-dimensional motion, it is still valuable to investigate 2D path following problem since there is a common practice to decompose the original tasks into simple subtasks of lower dimension [9, 25, 88]: a vertical control problem, also known as altitude or longitudinal control problems, and a horizontal, sometimes referred as lateral or planar, control. Moreover, in many applications, the motion control task can be naturally split into two separate stages: changing of attitude, for example, take-off and landing for aerial robots, and maneuvering maintaining the prescribed constant attitude, for instance, 2D path following. While there vertical and horizontal dynamics are dynamically coupled, for most applications this coupling is sufficiently small that its unwanted effects can be mitigated by control algorithms designed for disturbance rejection. For this reason, we confine ourselves to path-following algorithms on the Euclidean plane.

We consider the path following problem for the unicycle-type model where the longitudinal velocity $u_r > 0$ is a predefined constant

$$\dot{r} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = u_r m(\alpha) \in \mathbb{R}^2, \quad m(\alpha) = \begin{bmatrix} \cos\alpha \\ \sin\alpha \end{bmatrix}, \quad \dot{\alpha} = \omega. \tag{2.1}$$

Here r is the position of the robot's center of gravity C in the inertial Cartesian frame of reference $0XY$, α is the robot's orientation in this frame and $m(\alpha)$ is the unit orientation vector (see Fig. 2.2). The only control input to the system (2.1) is the angular velocity ω, and hence the system is underactuated.

2.2 Path following: problem setup

The desired curvilinear path $\mathcal{P}$ is the zero set of a function $\varphi \in C^2(\mathbb{R}^2 \to \mathbb{R})$, i.e. it is described by the implicit equation

$$\mathcal{P} \triangleq \{(x, y) : \varphi(x, y) = 0\} \subset \mathbb{R}^2. \tag{2.2}$$

The same curve $\mathcal{P}$ may be represented in the implicit form (2.2) in many ways. The principal restriction imposed by our approach is *regularity* [28] in some vicinity of

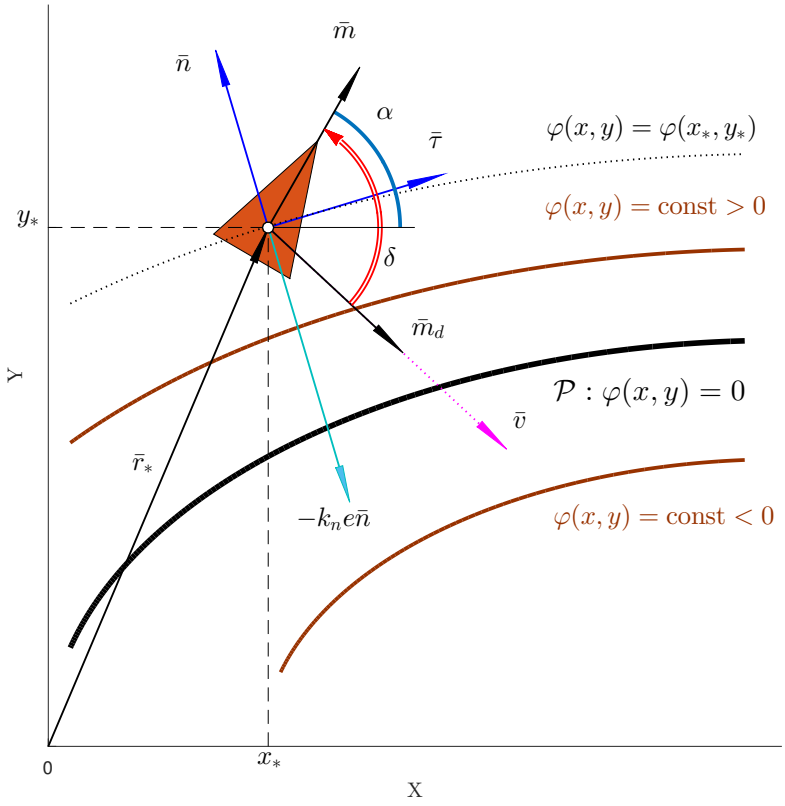


Figure 2.2: The robot orientation and level sets of the function $\varphi(x,y)$

$\mathcal{P}$ one has

$$n(x,y) \triangleq \nabla\varphi(x,y) = \left[\frac{\partial\varphi(x,y)}{\partial x}; \frac{\partial\varphi(x,y)}{\partial y}\right]^\top \neq 0 \tag{2.3}$$

As illustrated in Fig. 2.2, the plane $\mathbb{R}^2$ is covered by the disjoint *level sets* of the function φ, that is, the sets where $\varphi(x,y) = c = const$. If (2.3) holds, then the vector $n(x,y)$ is the *normal* vector to the corresponding level set at the point (x,y).

The path $\mathcal{P}$ is one of the level sets, corresponding to $c = 0$; the value $\varphi(x(t),y(t))$ can be considered as a (signed) "distance" from the robot to the path (differing, as usual, from the Euclidean distance), or the *tracking error* [28]. More generally, choosing an arbitrary strictly increasing function $\psi \in C^1(\mathbb{R} \to \mathbb{R})$ with $\psi(0) = 0$

(and thus $\psi(s)s > 0$ for any $s \neq 0$), one may define the tracking error as follows

$$e(x,y) \triangleq \psi[\varphi(x,y)] \in \mathbb{R}. \tag{2.4}$$

By definition, $e = 0$ if and only if $(x,y) \in \mathcal{P}$.

Problem 1. *To design a path following algorithm, i.e. a causal feedback law* $(x(\cdot), y(\cdot), \alpha(\cdot)) \mapsto \omega(\cdot)$*, which eliminates the tracking error* $|e(t)| \xrightarrow[t\to\infty]{} 0$*, bringing thus the robot to the predefined path* $\mathcal{P}$*.*

Notice that the definition of tracking error (2.4) involves the mapping $\psi(\cdot)$, which is a free parameter of the algorithm. Although one can get rid of this parameter replacing φ by the composition $\psi \circ \varphi$, it is convenient to distinguish between the path-defining function $\varphi(x,y)$ and the tracking error, depending on the choice of $\psi(s)$. The path representation is usually chosen as simple as possible: for instance, dealing with a straight line, it is natural to choose linear $\varphi(x,y)$, while the circular path is naturally described by $\varphi(x,y) = (x-x_0)^2 + (y-y_0)^2$. At the same time, some mathematical properties of the algorithm, in particular, the region where convergence of the algorithm is guaranteed, depends on the way the tracking error is calculated. As will be discussed, it may be convenient to choose $\psi(\cdot)$ bounded with $|\psi'(s)| \xrightarrow[|s|\to\infty]{} 0$, for instance, $\psi(s) = \arctan s^p$ or $\psi(s) = |s|^p \operatorname{sign} s/(1+|s|^p)$ with $p \geq 1$. These functions, as well as the simplest function $\psi(s) = s$ satisfy the condition, which is henceforth supposed to hold

$$\sup_{u\in\mathbb{R}} \psi'(\psi^{-1}(u)) < \infty. \tag{2.5}$$

2.3 Technical assumptions on the desired path

Henceforth the following three technical assumptions are adopted, excluding some "pathological" situations. Our first assumption enables one to use of the tracking error $e(x,y)$ as a "signed distance" to the path $\mathcal{P}$ and implies that the *asymptotic* vanishing of the error $e(r(t)) \xrightarrow[t\to\infty]{} 0$ entails the convergence to the path in the usual Euclidean metric, i.e. $\operatorname{dist}(r(t), \mathcal{P}) \to 0$. The distance from a point r_0 to the path $\mathcal{P}$ is $\operatorname{dist}(r_0, \mathcal{P}) \triangleq \inf\{|r_0 - r| : r \in \mathcal{P}\}$. More generally, the distance between sets A, B is defined as $\operatorname{dist}(A,B) \triangleq \inf\{|r_1 - r_2| : r_1 \in A, r_2 \in B$.

Assumption 2.1. For an arbitrary constant $\varkappa > 0$ one has

$$\inf\{|e(r)| : \operatorname{dist}(r, \mathcal{P}) \geq \varkappa\} > 0. \tag{2.6}$$

Our second assumption provides the regularity condition (2.3) in a sufficiently small vicinity of $\mathcal{P}$.

Assumption 2.2. The set of **critical points**, where n vanishes,

$$\mathcal{C}_0 \triangleq \{(x,y) \in \mathbb{R}^2 : \nabla\varphi(x,y) = 0\},$$

is separated from $\mathcal{P}$ by a positive distance $\operatorname{dist}(\mathcal{C}_0, \mathcal{P}) > 0$.

In the most typical cases where either $\mathcal{P}$ is a closed curve or $\mathcal{C}_0$ is compact, Assumption 2.2 boils down to the condition $\mathcal{C}_0 \cap \mathcal{P} = \varnothing$.

Our final assumption is similar in spirit to Assumption 2.1 and guarantees that the asymptotic vanishing of the normal vector $n(r(t)) \xrightarrow[t\to\infty]{} 0$ is possible only along a trajectory, converging to $\mathcal{C}_0$, i.e. $\operatorname{dist}(r(t), \mathcal{C}_0) \to 0$.

Assumption 2.3. For an arbitrary constant $\varkappa > 0$ one has

$$\inf\{|n(r)| : \operatorname{dist}(r, \mathcal{C}_0) \geq \varkappa\} > 0. \tag{2.7}$$

Assumption 2.3 implies the following useful technical lemma.

Lemma 2.4. *Consider a Lipschitz vector-function $r : [0;\infty) \to \mathbb{R}^2$ such that $r(t)$ does not converge to $\mathcal{C}_0$ as $t \to \infty$, that is, $\limsup\limits_{t\to\infty} d(t) > 0$ where $d(t) = \operatorname{dist}(r(t), \mathcal{C}_0)$. Then $\int_0^\infty |n(r(t))|^p dt = \infty$ for any $p > 0$.*

Proof. It can be easily noticed that $d(\cdot)$ is a Lipschitz function, since as can be easily shown, $|d(t_1) - d(t_2)| \leq |r(t_1) - r(t_2)| \leq M|t_1 - t_2| \,\forall t_1, t_2 \geq 0$, where $M > 0$ is the Lipschitz constant for $r(\cdot)$. Since $\limsup\limits_{t\to\infty} d(t) > 0$, a number $\varepsilon > 0$ and a sequence $t_k \to \infty$ exist such that $d(t_k) \geq 2\varepsilon$ and, therefore, $d(t) \geq \varepsilon$ for $t \in \Delta_k = [t_k; t_k + M^{-1}\varepsilon]$. Passing to subsequences, one may assume without loss of generality that $t_k + M^{-1}\varepsilon < t_{k+1}$ and thus the intervals Δ_k are disjoint. Assumption 2.3 implies that for sufficiently small $c > 0$ one has $|n(r(t))| \geq c$ whenever $t \in \Delta_k$. Hence $\int_{\Delta_k} |n(r(t))|^p dt \geq \varepsilon M^{-1} c^p$ and therefore $\int_0^\infty |n(r(t))|^p \, dt = \infty$.

□

2.4 Guiding vector field design

In this chapter, we construct the Guiding Vector Field (GVF) to be used in the path following control algorithm. We show that the integral curves of this field lead either to the desired path $\mathcal{P}$ or to the critical set $\mathcal{C}_0$. Furthermore, we give efficient criteria, ensuring that the integral curves of the second type are either absent or cover a set of zero measure on $\mathbb{R}^2$.

Besides the normal vector $n(x,y) = \nabla\varphi(x,y)$, at each point we consider the

tangent vector to the level set

$$\tau(x,y) = En(x,y), \quad E = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \tag{2.8}$$

If the point is regular (2.3), the basis (τ, n) is right-handed oriented[1] (Fig. 2.2). Our goal is to find such a vector field $v(x,y)$, where the absolute tracking error $|e|$ is decreasing along each of its integral curves (unless $e = 0$), and the curves starting on $\mathcal{P}$ do not leave it. We define this vector field by

$$v(x,y) \triangleq \tau(x,y) - k_n e(x,y) n(x,y). \tag{2.9}$$

The integral curves of the vector field (2.9) correspond to the trajectories of the autonomous differential equation

$$\frac{d}{dt}\xi(t) = v(\xi(t)) \in \mathbb{R}^2, \quad t \geq 0. \tag{2.10}$$

2.4.1 Properties of the integral curves

We notice first that the vector field v is C^1-smooth, which implies the *local* existence and uniqueness of solutions of the system (2.10). Obviously, any point $(x_0, y_0) \in \mathcal{C}_0$ corresponds to the *equilibrium* of (2.10), which is a trivial single-point integral curve. The uniqueness property implies that non-constant integral curves are free of the critical points; the corresponding solutions, however, may converge to $\mathcal{C}_0$ *asymptotically*. In general, a solution may also escape to infinity in finite time.

The following lemma establishes the principal *dichotomy* property of the integral curves, stating that any curve leads either to the desired path $\mathcal{P}$ or to the critical set $\mathcal{C}_0$.

Lemma 2.5. *Let $\xi(t)$, $t \in [0; t_*)$ where $t_* \leqslant \infty$ be a maximally prolonged solution to* (2.10)*. Then two situations are possible*

1. *either* $\operatorname{dist}(\xi(t), \mathcal{P}) \xrightarrow[t \to t_*]{} 0$*, that is, the solution converges to the desired path,*
2. *or* $t_* = \infty$ *and* $\operatorname{dist}(\xi(t), \mathcal{C}_0) \xrightarrow[t \to \infty]{} 0$*.*

Proof. The proof is based on the Lyapunov function

$$V(x,y) = \frac{1}{2} e(x,y)^2 \geq 0. \tag{2.11}$$

[1] In the subsequent constructions, one may replace τ by $-\tau$ and E by $-E$, which leads the change of the path following direction.

A straightforward computation shows that

$$\nabla e(\xi) = \psi'(\varphi(\xi))\,\nabla\varphi(\xi) = \psi'(\psi^{-1}(e(\xi)))|n(\xi)|$$

and hence V is non-increasing along the trajectory $\xi(t)$ since its derivative is

$$\begin{aligned}\dot V(\xi) &= e(\xi)\psi'(\psi^{-1}(e(\xi)))n^\top(\xi)v(\xi) \overset{(2.9)}{=} \\ &\overset{(2.9)}{=} -k_n e(\xi)^2\psi'(\psi^{-1}(e(\xi)))|n(\xi)|^2 \le 0.\end{aligned} \tag{2.12}$$

In particular, there exists the limit $e_* = \lim\limits_{t\to t_*}|e(\xi(t))| \ge 0$. If $e_* = 0$ then, due to Assumption 2.1, statement 1) holds: the solution converges to $\mathcal{P}$. Suppose now that $e_* > 0$. We are going to prove that statement 2) is valid. Notice first that $\psi'(\psi^{-1}(e(\xi(t)))$ is uniformly bounded and positive; the same holds for $V(\xi(t))$. The equality (2.12) thus implies that $\int_0^{t_*}|n(\xi(t))|^2dt < \infty$. Since $|v|^2 = (1+k_n^2e^2)|n|^2$, (2.10) implies that $\int_0^{t_*}|\dot\xi(t))|^2dt < \infty$. If one had $t_* < \infty$, the Cauchy-Schwartz inequality would imply that

$$|\xi(T)-\xi(0)|^2 = \left(\int_0^T|\dot\xi(t)|dt\right)^2 \le t_*\int_0^{t_*}|\dot\xi(t)|^2dt \;\;\forall T < t_*,$$

which contradicts the assumption that $\xi(t)$ is a maximally prolonged solution, escaping to infinity as $t\to t_*$. Therefore, $t_* = \infty$; statement 2 now follows from Lemma 2.4. □

A natural question arises on how "large" the set of trajectories is, converging to the critical set $\mathcal{C}_0$. In many practical examples, this set is *finite*. This holds, in particular if $\varphi(x,y) \ne 0$ when $|x|+|y|$ is sufficiently large and its *Hessian matrix*

$$H(x,y) = H(x,y)^\top = \begin{bmatrix}\frac{\partial^2}{\partial x^2}\varphi(x,y) & \frac{\partial^2}{\partial x\partial y}\varphi(x,y)\\ \frac{\partial^2}{\partial x\partial y}\varphi(x,y) & \frac{\partial^2}{\partial y^2}\varphi(x,y)\end{bmatrix} \tag{2.13}$$

is sign-definite at any point $(x_0,y_0)\in\mathcal{C}_0$, i.e. either $H(x_0,y_0) > 0$ or $H(x_0,y_0) < 0$. In this situation $\mathcal{C}_0$ is bounded (and thus compact) and all its points are isolated, which implies that $\mathcal{C}_0$ is finite. If the set $\mathcal{C}_0$ is finite, the convergence $\operatorname{dist}(\xi(t),\mathcal{C}_0) \xrightarrow[t\to\infty]{} 0$, obviously, means that the solution converges to a critical point: $\xi_* \triangleq \lim\limits_{t\to\infty}\xi(t) \in \mathcal{C}_0$.

Definition 2.6. Let $\xi_*\in\mathcal{C}_0$ be an equilibrium point of (2.10). The *stable manifold* of ξ_*, denoted by $W(\xi_*)$, is the set of all points ξ_0 such that the solution of (2.10), starting at $\xi(0)=\xi_0$, exists for all $t\ge 0$ and $\xi_* \triangleq \lim\limits_{t\to\infty}\xi(t)$.

We will use the following corollary of the central manifold theorem (see e.g. Theorem 4.1 and Proposition 4.1 in [79]).

Lemma 2.7. *If both eigenvalues of the Jacobian matrix $J(\xi_*) = \frac{\partial}{\partial \xi} v(\xi_*)$ are strictly unstable $Re\,\lambda_{1,2} J(\xi_*) > 0$, then $W(\xi_*) = \{\xi_*\}$. If $J(\xi_*)$ has at least one strictly unstable eigenvalue, then $W(\xi)$ is a set of zero measure.*

Lemma 2.7 in turn has the following important corollary.

Corollary 2.8. *Let the set $\mathcal{C}_0$ be finite and for any of its points $\xi_* \in \mathcal{C}_0$ the matrix $e(\xi_*)H(\xi_*)$ has a* negative[2] *eigenvalue. Then the maximally prolonged solution of* (2.10) *(possibly, existing on finite interval only) converges to $\mathcal{P}$ for almost all initial conditions $\xi(0)$. If $e(\xi_*)H(\xi_*) < 0$ for any $\xi_* \in \mathcal{C}_0$, this convergence takes place whenever $\xi(0) \notin \mathcal{C}_0$.*

Proof. A straightforward computation shows that if $\nabla\varphi(\xi_*) = 0$ then $J(\xi_*) = (E - k_n e(\xi_*))H(\xi_*)$ and

$$\begin{aligned} \operatorname{Tr} J(\xi_*) &= -k_n e(\xi_*) \operatorname{Tr} H(\xi_*), \\ \det J(\xi_*) &= (1 + k_n^2 e(\xi_*)^2) \det H(\xi_*). \end{aligned}$$

It can be easily shown that a symmetric 2×2 matrix M is non-negatively definite $M \geq 0$ if and only $\det M = \lambda_1(M)\lambda_2(M) \geq 0$ and $\operatorname{Tr} M = \lambda_1(M) + \lambda_2(M) \geq 0$. Here $\lambda_i(M)$ stand for the eigenvalues of M. Since $eH(e)$ is *not* non-negatively definite, either $\det J < 0$ or $\operatorname{Tr} J > 0, \det J \geq 0$. In both situations, the matrix $J(\xi_*)$ has at least one strictly unstable eigenvalue. Furthermore, if $eH(e) < 0$ then $\operatorname{Tr} J > 0, \det J > 0$, i.e. both eigenvalues of J are strictly unstable. The statement now follows from Lemma 2.7. □

2.4.2 The GVF and the "ideal" motion of the robot

The main idea of the path following controller, designed in the next section, is to steer the robot to the integral curve of the field (2.9). In other words, when the robot is passing a point $r = (x, y) \in \mathbb{R}^2$, its desired orientation is

$$m_d(x, y) = \frac{1}{|v(x, y)|} v(x, y). \tag{2.14}$$

The field of unit vectors $m_d(x, y)$, henceforth referred to as the *guiding vector field* (GVF), is defined at any regular point (x, y), where $n \neq 0$ and thus $|v| = \sqrt{1 + k_n^2 e^2}|n| \neq 0$. Fig. 4.1 illustrates the relation between the vectors r, m, τ, n, v, m_d.

[2] Recall that H is a symmetric matrix, so its eigenvalues are real.

Consider the desired motion of the robot, "ideally" oriented along the integral curves of the GVF at any point. Its position vector $r(t)$ obeys the differential equation

$$\dot{r}(t) = u_r m_d(r(t)), \quad t \geq 0. \tag{2.15}$$

The following lemma is a dual of Lemma 2.5.

Lemma 2.9. *Let $r(t)$, $t \in [0; t_*)$ where $t_* \leqslant \infty$ be a maximally prolonged solution to* (2.15)*. Then two situations are possible*

1. *either* $\operatorname{dist}(r(t), \mathcal{C}_0) \xrightarrow[t \to t_*]{} 0$*,*
2. *or* $t_* = \infty$ *and* $\operatorname{dist}(r(t), \mathcal{P}) \xrightarrow[t \to \infty]{} 0$ *that is, the solution converges to the desired path.*

Proof. If $t_* < \infty$ then the limit $r_* = r(t_* - 0) = r(0) + \int_0^{t_*} m_d(r(t))dt$ exists and, since the solution is maximally prolonged, one obviously has $|n(r_*)| = 0$, i.e. $r_* \in \mathcal{C}_0$ and statement 1 holds. The case of $t_* = \infty$ is considered similar to the proof of Lemma 2.5. Introducing the Lyapunov function (2.11), its derivative is shown to be

$$\begin{aligned} \dot{V}(r) &= u_r e(r) \psi'(\psi^{-1}(e(r))) n^\top(r) m_d(r) \overset{(2.9)}{=} \\ &\overset{(2.9)}{=} \underbrace{-\frac{u_r k_n e(r)^2}{\sqrt{1 + k_n^2 e(r)^2}} \psi'(\psi^{-1}(e(r)))}_{\Psi(e(r))} |n(r)| \leq 0. \end{aligned} \tag{2.16}$$

There exists the limit $e_* = \lim_{t \to \infty} |e(r(t))|$. If $e_* = 0$, statement 2 holds thanks to Assumption 2.1. Otherwise, $e_* > 0$ and $\Psi(e(r(t)))$ is uniformly positive. In view of (2.16), one has $\int_0^\infty |n(r(t))| dt < \infty$. Lemma 2.4 entails now statement 1. □

Remark 2.10. The equation (2.16) implies that $\dot{V} = -k_n V |n| \theta(e)$, where $\theta(e) \to \theta_0 > 0$ as $e \to 0$. Assumptions 2.2 and 2.3 imply that $|n(r)| \geq \varkappa$ as $e \approx 0$. Therefore, the desired path $\mathcal{P}$ is *locally* exponentially attractive: if $|e(0)| \leq \varepsilon$, where ε is sufficiently small, then $|e(t)| \leq e^{-k_n u_r \beta t} |e(0)|$, where $\beta > 0$ is a constant, depending on $\psi(\cdot)$ and the normal vector $n(x, y)$ in the vicinity of $\mathcal{P}$. The coefficient k_n corresponds to for the attraction to $\mathcal{P}$. In the limit case $k_n = 0$, the path $\mathcal{P}$ is not asymptotically stable; the higher $k_n > 0$ one chooses, the stronger is the attraction to the path $\mathcal{P}$ in its small vicinity.

Obviously, the integral curves of (2.15) are in one-to-one correspondence with non-equilibrium integral curves of (2.10). Corollary 2.8 can now be reformulated as follows.

Corollary 2.11. *Let the set $\mathcal{C}_0$ be finite and for any of its points $\xi_* \in \mathcal{C}_0$, the matrix $e(\xi_*)H(\xi_*)$ has a negative eigenvalue. Then for almost all initial conditions $r(0)$ the*

solutions of (2.15) *can be prolonged up to* ∞ *and converge to* $\mathcal{P}$. *If* $e(\xi_*)H(\xi_*) < 0$ *for any* $\xi_* \in \mathcal{C}_0$, *this holds for any* $r(0) \notin \mathcal{C}_0$.

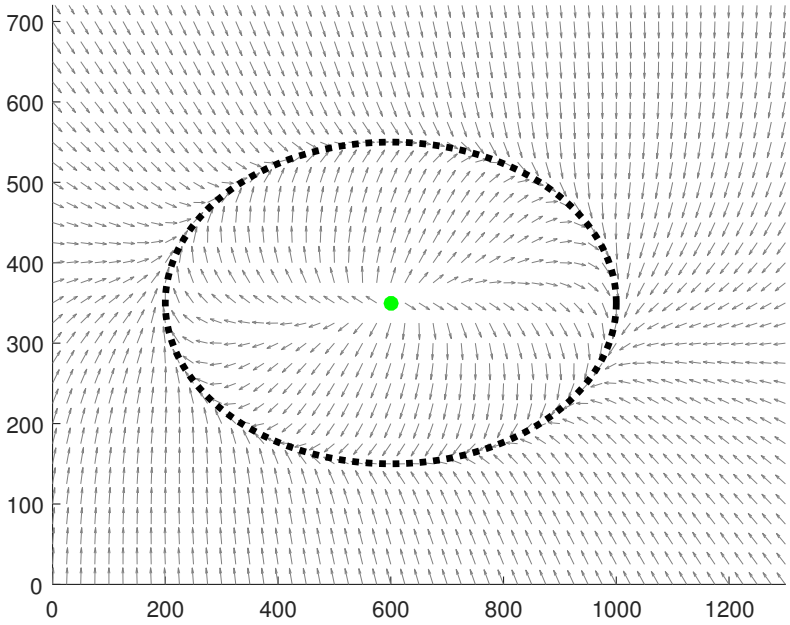


Figure 2.3: The GVF for an elliptic path

Note that the conditions of Corollary 2.11 always hold for *strictly convex* function $\varphi(x, y)$ (that is, $H(x, y) > 0$ at any point) since at the critical point (if it exists) the function φ attains its *global minimum*. Therefore, at this point (x_*, y_*) one has $\varphi(x_*, y_*) < 0$ and thus $e < 0$, which implies that $eH < 0$.

Fig. 2.3 demonstrates the GVF for the elliptic path, defined by the function

$$\varphi(x, y) = \frac{(x - x_0)^2}{a^2} + \frac{(y - y_0)^2}{b^2} - 1,$$

The corresponding GVF has the unique critical point at the center of ellipse. This critical point is “repulsive” and no trajectory of (2.15) converges to it. Fig. 2.4 illustrates the GVF for the Cassini oval, defined by the non-convex function

$$\varphi(x, y) = (x - x_0)^4 + (y - y_0)^4 - 2a^2((x - x_0)^2 - (y - y_0)^2) + a^4 - b^4.$$

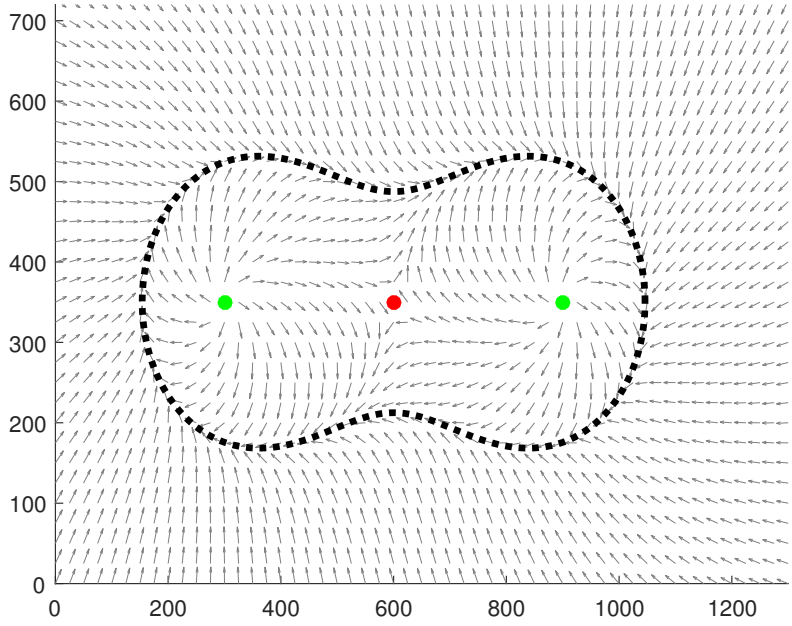


Figure 2.4: The GVF for a Cassini oval.

The corresponding set $\mathcal{C}_0$ consists of two "locus points" $(x_0 \pm a, y_0)$ and the "center" (x_0, y_0). At the locus points one has $H > 0$, whereas at the center H has one positive and one negative eigenvalue, so the robot potentially can be "trapped" at the center, but the set of corresponding initial conditions has zero measure.

Chapter 3
GVF-based path following

3.1 Overview of path following algorithms

One approach to steering to the desired path is to fix its time-parametrization; the path is thus treated as a function of time or, equivalently, the trajectory of some reference point. Path steering is then reduced to the *reference point tracking* problem that has been extensively studied in control theory and solved for a very broad class of nonlinear systems [39, 59]. An obvious advantage of the trajectory tracking approach is its applicability to a broad range of paths that can be e.g. non-smooth or self-intersecting. However, this method has unavoidable fundamental limitations, especially when dealing with general nonlinear systems. As shown in [86], unstable zero dynamics generally make it impossible to track a reference trajectory with arbitrary predefined accuracy; more precisely, the integral tracking error is uniformly positive independent of the controller design. Furthermore, in practice the robot's motion along the trajectory can be quite "irregular" due to its oscillations around the reference point. It is impossible to guarantee either path following at a precisely constant speed, or even the motion with a perfectly fixed direction. Attempting to keep close to the reference point, the robot may "overtake" it due to unpredictable disturbances and then turn back.

As has been clearly illustrated in the influential paper [2], these performance limitations of trajectory tracking can be removed by carefully designed *path following* algorithms. Unlike the tracking approach, path following control treats the path as a geometric curve rather than a function of time, dealing with its implicit equations or some time-free parametrization. The algorithm thus becomes "flexible to use a timing law as an additional control variable" [2]; this additional degree of freedom allows to maintain a constant forward speed or any other desired speed profile, which is extremely important e.g. for aerial vehicles, where the lifting force depends on the robot's speed. The *dynamic* controller, maintaining the longitudinal speed, is usually separated from the "geometric" controller, steering the robot to the desired path.

A widely used approach to path-following, originally proposed for car-like wheeled robots [64, 84], assumes the existence of the *projection* point, that is, the closest point on a path, and the robot's capability to measure the distance to

it (sometimes referred to as the "cross-track error"). The robot's mathematical model is represented in the Serret-Frenet frame, consisting of the tangent and normal vectors to the trajectory at the projection point. This representation allows to design efficient path following controllers for autonomous wheeled vehicles [1, 64, 84] that eliminate the cross-track error and maintain the desired vehicle's speed along the path. Further development of this approach leads to algorithms for the control of complex unmanned vehicles such as cars with multiple trailers [85] and agricultural tractors [61]. Projection-based *sliding mode* algorithms [1, 61] are capable to cope with uncertainties, caused by the non-trivial geometry of the path, lateral drift of the vehicle and actuator saturations. The necessity to measure the distance to the track imposes a number of limitations on the path-following algorithm. Even if the nearest point is unique, the robot should either be equipped with special sensors [1] or solve real-time optimization problems to find the cross-track error.

In general, the projection point cannot be uniquely determined when e.g. the robot passes a self-intersection point of the path or its position is far from the desired trajectory. A possible way to avoid these difficulties has been suggested in [89] and is referred to as the "virtual target" approach. The Serret-Frenet frame, assigned to the projection point in the algorithm from [64, 84], can be considered as the body frame of a *virtual target vehicle* to be tracked by the real robot. A modification of this approach, offered in [89], allows the virtual target to have its own dynamics, taken as one of the controller's design parameters. The design from [89] is based on the path following controller from [64], using the model representation in the Serret-Frenet frame and taking the geometry of the path into account. However, the controller from [89] implicitly involves target tracking since the frame has its own dynamics. Avoiding the projection problem, the virtual target approach thus inherits disadvantages of the usual target tracking. In presence of uncertainties the robot may slow down and turn back in order to trace the target's position [92].

A guidance strategy of a human helmsman inspired another path following algorithm, referred to as the *line-of-sight* (LOS) method [14, 26, 92] which is primarily used for air and marine crafts. Maintaining the desired speed of the robot, the LOS algorithm steers its heading along the LOS vector which starts at the robot's center of gravity and ends at the *target* point. This target is located ahead of the robot either on the path [92] or on the line, tangent to the path at the projection point [26]. Unlike the virtual target approach, the target is always chosen at a fixed prescribed distance from the robot, referred to as the *lookahead distance*. The maneuvering characteristics substantially depend on the lookahead distance: the shorter the distance is chosen, the more "aggressively" it steers. A thorough mathematical examination of the LOS method has been carried out in the recent paper [26], establishing the uniform semi-global exponential stability

(USGES) property.

Differential-geometric methods for invariant set stabilization [28, 39, 59] have given rise to a broad class of "set-based" [23] path following algorithms. Treating the path as a geometric set, the algorithm is designed to make it invariant and attractive (globally or locally). Typically the path is considered as a set where some (nonlinear) output of the system vanishes, and thus the problem of its stabilization boils down to the output regulation problem. To solve it, various linearization techniques have been proposed [3, 28, 29, 35, 45, 46, 51, 101]. For stabilization of a closed *strictly convex* curve, an elegant passivity-based method has been established in [23].

The path following strategy considered in this chapter is based on the idea of the reference *Guiding Vector Field*. Vector field algorithms are widely used in collision-free navigation and extremum seeking problems [38, 62, 63, 73]; their efficiency in path following problems has been recently demonstrated in [53, 67, 92, 96]. A vector field is designed such that its integral curves approach the path asymptotically. Steering the robot along the integral curves, the control algorithm drives it to the desired path. Unlike many path following algorithms that guarantee convergence only in a *sufficiently small* vicinity of the desired path, the vector field algorithm guarantees convergence of any trajectory, which does not encounter the "critical" points where the vector field is degenerate. In particular, in any invariant domain without critical points the convergence to the path can be proved. For holonomic robots described by a single or double integrator model, a general vector-field algorithm for navigation along a general smooth curve in an n-dimensional space has been discussed in [30]. However, for more realistic nonholonomic vehicles the vector-field algorithms have been studied mainly for straight lines and circular paths [8, 53, 67], where they demonstrate better, in several aspects, performance compared to other approaches [92]. Unlike [53, 67, 92], we propose and analyze rigorously a vector-field algorithm for guidance of a general nonholonomic robot along a general smooth planar path, given in its implicit form.

3.2 The Path Following Controller Design

In this section, we design the controller, solving Problem 1 via steering the robot to the integral curves of GVF (2.14). This controller uses the GVF $m_d(x, y)$ at the current robot's position and the function $\omega_d(x, y, \alpha)$, which is introduced below and measures the GVF's "curvature" along the robot's trajectory.

3.2.1 A technical lemma

Lemma 3.1. *Let $m_d(t) = m_d(x(t), y(t))$ stand for the GVF along a trajectory of the robot. Then its derivative $\dot{m}_d(t)$ is*

$$\dot{m}_d(t) = -\omega_d(x(t), y(t), \alpha(t)) E m_d(t), \tag{3.1}$$

where $\omega_d : \mathbb{R}^3 \to \mathbb{R}$ is continuous and uniquely determined by the functions $\varphi(\cdot)$, $\psi(\cdot)$ and the constant k_n from (2.9).

Proof. Differentiating the equality $|m_d(t)|^2 = 1$, one has that $\dot{m}_d(t)^\top m_d(t) = 0$, that is, $\dot{m}_d(t) \perp m_d(t)$. Therefore, the vector $\dot{m}_d(t)$ is proportional to the unit vector $Em_d(t)$, so that $\dot{m}_d(t) = -\omega_d(t) E m_d(t)$ and the scalar multiplier $\omega_d(t)$ can be found from $\omega_d(t) = -\dot{m}_d(t)^\top E m_d(t)$. It remains to prove that $\omega_d(t)$ in fact depends only on the trajectory $(x(t), y(t), \alpha(t))$, and this dependence is continuous. Introducing the vector field (2.9) $v(t) = v(x(t), y(t))$ and the tracking error $e(t) = e(x(t), y(t))$ along the robot's trajectory, a straightforward computation shows that

$$\begin{aligned} \dot{m}_d &= \frac{d}{dt}\frac{v}{\|v\|} = \left(\frac{I_2}{\|v\|} - \frac{vv^\top}{\|v\|^3}\right)\dot{v}, \\ \dot{v} &= u_r[E - k_n e I_2] H(x,y) m(\alpha) - k_n \dot{e}\, n(x,y), \\ \dot{e} &= u_r \psi'(\varphi(x,y))\, n(x,y)^\top m(\alpha). \end{aligned} \tag{3.2}$$

Here I_2 is the 2×2 identity matrix and H is the Hessian (2.13). Since $\varphi \in C^2$, $\dot{m}_d$ and m_d continuously depend on the triple $(x(t), y(t), \alpha(t))$, the same holds for

$$\omega_d = -\dot{m}_d^\top E m_d.$$

□

To clarify the meaning of the function ω_d, suppose for the moment that the robot's speed is $u_r = 1$. If the robot is moving strictly along the integral curves of the GVF, then ω_d is the signed curvature of the robot's trajectory at its current position. In general, ω_d can be treated as the "desired" curvature of the robot's trajectory, which may differ from its real curvature.

3.2.2 The steering algorithm

As was discussed in the foregoing, the idea of the path following algorithm is to steer the robot's orientation along the guiding vector field $m_d = m_d(x, y)$. We introduce the directed angle $\delta = \delta(x, y, \alpha) \in (-\pi; \pi]$ between m_d and m (see Fig. 2.2). The function δ is thus defined at any point (x, y, α) and C^1-smooth at the points where $m(\alpha) \neq -m_d(x, y)$.

The orientation vector's derivative along the trajectory is

$$\dot{m}(t) = \frac{d}{dt} m(\alpha(t)) = \omega(t) \begin{bmatrix} -\sin\alpha(t) \\ \cos\alpha(t) \end{bmatrix} = -\omega(t) E m(t). \tag{3.3}$$

On the other hand, m may be decomposed (Fig. 2.2) as

$$m = (\cos\delta) m_d - (\sin\delta) E m_d = [(\cos\delta) I_2 - (\sin\delta) E] m_d, \tag{3.4}$$

At any point where $\delta(t) < \pi$ and thus $\dot{\delta}(t)$ exists, one obtains

$$\begin{aligned} -\omega E m \overset{(3.3)}{=} \dot{m} = -\dot{\delta} \underbrace{[(\sin\delta) I_2 + (\cos\delta) E] m_d}_{=Em} + \\ + [(\cos\delta) I_2 - (\sin\delta) E] \dot{m}_d \overset{(3.1)}{=} -\dot{\delta} E m - \\ - \omega_d \underbrace{[(\cos\delta) I_2 - (\sin\delta) E] E m_d}_{=Em} = -(\dot{\delta} + \omega_d) E m, \end{aligned}$$

entailing the following principal relation between δ, ω and ω_d

$$\dot{\delta} = \omega - \omega_d. \tag{3.5}$$

Furthermore, at *any* time the following equality is valid

$$\frac{d}{dt} \sin\delta = -\frac{d}{dt} m^\top E m_d = (\omega - \omega_d) \cos\delta. \tag{3.6}$$

When $\delta(t_0) = \pi$ and $\omega(t) - \omega_d(t) \le 0$ for $t \approx t_0$, equation (3.6) entails that $\sin\delta \ge 0$ and hence $\delta = \frac{\pi}{2} + \arcsin(\sin\delta)$ for $t \in [t_0; t_0 + \varepsilon)$, where $\varepsilon > 0$ is sufficiently small. In this situation, the function $\delta(x(t), y(t), \alpha(t))$ has the right derivative[1] $\dot{\delta} = D^+\delta$, satisfying (3.5) as $t \in [t_0; t_0 + \varepsilon)$.

We now describe our path-following algorithm:

Algorithm 1: GVF Path Following

$$\omega(t) = \omega_d(x(t), y(t), \alpha(t)) - k_\delta \delta(x(t), y(t), \alpha(t)). \tag{3.7}$$

$$\omega_d = -\dot{m}_d^\top E m_d$$

$\dot{m}_d$ is found from (3.2)

Here $k_\delta > 0$ is a constant, determining the convergence rate.

[1] By definition, the right derivative of a function $f(t)$ at $t = t_0$ (written $f'(t_0 + 0)$ or $D^+ f(t_0)$) is defined by $D^+ f(t_0) = \lim\limits_{t \to t_0 + 0} \frac{f(t) - f(t_0)}{t - t_0}$.

When $\delta(x(t), y(t), \alpha(t)) < \pi$, the equality (3.5) holds and thus

$$\boxed{\dot{\delta} = \omega - \omega_d = -k_\delta \delta.} \tag{3.8}$$

Furthermore, even for $\delta(x(0), y(0), \alpha(0)) = \pi$ one has $\omega - \omega_d = -k_\delta \delta < 0$ as $t \approx t_0$ and hence (3.8) retains its validity at $t = 0$, treating $\dot{\delta}$ as the right derivative $D^+ f$. Thus, considering $\dot{\delta}$ as a new control input, the algorithm (3.5) is equivalent to a very simple proportional controller (3.8), providing, in particular, that $\delta(x(t), y(t), \alpha(t)) < \pi \, \forall t > 0$.

3.2.3 Local existence and convergence of the solutions

In this subsection we examine the properties of the solutions of the closed-loop system (2.1), (3.7), rewritten as follows

$$\begin{aligned} \dot{x}(t) &= u_r \cos \alpha(t), \\ \dot{y}(t) &= u_r \sin \alpha(t), \\ \dot{\alpha}(t) &= \omega_d(x(t), y(t), \alpha(t)) - k_\delta \delta(x(t), y(t), \alpha(t)). \end{aligned} \tag{3.9}$$

The right-hand side of (3.9) is continuous at any point (x_0, y_0, α_0), where $n(x_0, y_0) \neq 0$ and $\delta_0 = \delta(x_0, y_0, \alpha_0) < \pi$. However, the discontinuity at the points where $\delta_0 = \pi$ makes the usual existence theorem [50] inapplicable. To avoid this problem, we consider the equivalent "augmented" system

$$\begin{aligned} \dot{x}(t) &= u_r \cos \alpha(t), \\ \dot{y}(t) &= u_r \sin \alpha(t), \\ \dot{\alpha}(t) &= \omega_d(x(t), y(t), \alpha(t)) - k_\delta \delta_*(t), \\ \dot{\delta}_*(t) &= -k_\delta \delta_*(t). \end{aligned} \tag{3.10}$$

As was discussed in the previous subsection, any solution of (3.9) satisfies (3.10) with $\delta_*(t) = \delta(x(t), y(t), \alpha(t))$, and vice versa: choosing a solution $(x(t), y(t), \alpha(t), \delta_*(t))$, where $\delta_*(0) = \delta(x(0), y(0), \alpha(0)) \in (-\pi; \pi]$, one has $\delta_*(t) = \delta(x(t), y(t), z(t))$ for any $t \geq 0$ due to (3.8). Unlike (3.9), the right-hand side of (3.10) is a C^1-smooth function of (x, y, α, δ_*) at any point where $n(x, y) \neq 0$. The standard existence and uniqueness theorem [50] implies the following lemma.

Lemma 3.2. *For any point $\zeta_0 = (x_0, y_0, \alpha_0)$, such that $n(x_0, y_0) \neq 0$, there exists the unique solution $\zeta(t) = (x(t), y(t), \alpha(t))$ with $\zeta(0) = \zeta_0$. Extending this solution to the maximal existence interval $[0; t_*)$, one either has $t_* = \infty$ or $t_* < \infty$ and $(x(t), y(t)) \xrightarrow[t \to t_*]{} (x_*, y_*) \in \mathcal{C}_0$.*

Proof. Reducing the Cauchy problem for the closed-loop system (3.9) to the Cauchy problem for (3.10), one shows that the solution exists locally and is unique [50]. Let its maximally prolongable solution $(x(t), y(t), \alpha(t))$ be defined on $\Delta_* \stackrel{\Delta}{=} [0; t_*)$ with $t_* < \infty$. Since $|\dot{r}(t)| = u_r$, the limit exists

$$r_* = \lim_{t \to t_*} r(t) = r(0) + \int_0^{t_*} \dot{r}(t) dt.$$

We are going to show that $r_* \in \mathcal{C}_0$, i.e. $n(r_*) = 0$. Suppose, on the contrary, that $|n(r_*)| > 0$; therefore, $|v(x(t), y(t))|$ is *uniformly* positive on Δ_*. Using (3.2), this implies that $\dot{m}_d(t)$, and hence $\omega_d(x(t), y(t), \alpha(t))$ and $\omega(t)$ are uniformly bounded on Δ_*. Thus there exists the finite limit

$$\alpha_* = \lim_{t \to t_*} \alpha(t) = \alpha(0) + \int_0^{t_*} \omega(t) dt,$$

enabling one to define the solution at $t = t_*$ and then to prolong it to $[t_*, t_* + \varepsilon)$, i.e. the solution is not maximally prolonged. This contradiction implies that $n(r_*) = 0$. □

Note that if $\delta(0) = 0$, that is, the robot was perfectly oriented along the GVF at the starting moment, (3.5) implies that $\delta(t) \equiv 0$ so that the robot follows the integral curve of the GVF. As was shown in Section 2.4, in this "ideal" situation the robot either approaches the desired path $\mathcal{P}$ or is driven to one of the critical points. The latter situation is practically impossible if the conditions of Corollary 2.11 are valid since the set of integral curves, leading to the set $\mathcal{C}_0$, has zero measure.

One may consider (3.9) as a system with *slow-fast* dynamics. Informally, the controller (3.8) provides the exponential convergence of the robot to an integral curve of the GVF; after this "fast" transient process, the robot "slowly" follows this integral curve and approaches the desired trajectory, unless it is "trapped" in a critical point. Ignoring the "fast dynamics", one may suppose that the statement of Lemma 2.9 remains valid for a general solution of the system (3.9). This argument, however, is not mathematically rigorous. Recalling the proofs of Lemmas 2.5 and 2.9, one may notice that the central argument was the *non-increasing* property of the Lyapunov function (2.11). Although the deviation of the robot from the integral curve exponentially decreases due to (3.8), the non-increasing property, in general, fails: when the robot is positioned very close to the desired path, the tracking error may increase. However, this effect does not destroy the dichotomy property (any solution converges either to $\mathcal{P}$ or to $\mathcal{C}_0$) under the following assumption, which usually holds in practice (e.g. if $\varphi(\cdot)$ is a polynomial).

Assumption 3.3. There exist $\theta \in (0; k_\delta)$ and $C > 0$ such that

$$|n(r)| \le Ce^{\theta|r|} \quad \text{as } |r| \to \infty. \tag{3.11}$$

The latter condition can be relaxed in the case of a closed path; however, we adopt it to consider both bounded and unbounded paths in a unified way. Note that if $\varphi(x, y)$ is a polynomial function, then $k_\delta > 0$ can be arbitrarily small.

Henceforth all Assumptions 2.1, 2.2, 2.3 3.3 are supposed to be valid.

Theorem 3.4. *For any maximally prolonged solution $(x(t), y(t), \alpha(t))$ of (3.9), defined for $t \in [0; t_*)$, one of the following statements holds:*

1. *either $t_* = \infty$ and $\operatorname{dist}(r(t), \mathcal{P}) \xrightarrow[t\to\infty]{} 0$, or*
2. *$\operatorname{dist}(r(t), \mathcal{C}_0) \to 0$ as $t \to t_*$.*

In other words, for any initial condition the algorithm drives the robot to either the desired path $\mathcal{P}$ or $\mathcal{C}_0$.

Proof. In the case where $t_* = \infty$ statement 2 holds due to Lemma 3.2. Suppose that $t_* = \infty$. Differentiating the function (2.11) along the trajectories, it can be shown that

$$\begin{aligned}
\dot V &= u_r e\psi'(\psi^{-1}(e))n^\top m \overset{(3.4)}{=} \\
&\overset{(3.4)}{=} u_r e\psi'(\psi^{-1}(e))n^\top [m_d\cos\delta - Em_d\sin\delta] \overset{(2.14)}{=} \\
&\overset{(2.14)}{=} \frac{u_r e\psi'(\psi^{-1}(e))}{|v|} n^\top [v\cos\delta - Ev\sin\delta] \overset{(2.9)}{=} \\
&\overset{(2.9)}{=} \frac{u_r e\psi'(\psi^{-1}(e))}{\sqrt{1+k_n^2e^2}}|n|(-k_n e\cos\delta + \sin\delta) = \\
&= \Phi(e)|n|(-k_n e\cos\delta + \sin\delta).
\end{aligned} \tag{3.12}$$

Here $\Phi(e)$ denotes the bounded, in view of (2.5), function

$$\Phi(e) \triangleq \frac{u_r e\psi'(\psi^{-1}(e))}{\sqrt{1+k_n^2e^2}}. \tag{3.13}$$

Since $|\sin\delta| \le |\delta|$, Assumption 3.3 entails that $\int_0^\infty |\Phi(e(t))n(t)\sin\delta(t)|\,dt < \infty$. Notice now that $(-\Phi(e)e) \le 0$ and $\cos\delta(t) > 0$ as t becomes sufficiently large. Thus the integral $I = \int_0^\infty (-e\Phi(e)|n|)\cos\delta\,dt$ exists, being either finite or equal to $-\infty$. This implies, thanks to (3.12), the existence of $\int_0^\infty \dot V\,dt = \lim\limits_{t\to+\infty} V(t) - V(0)$. Since $V \ge 0$, one has $I > -\infty$ and therefore there exists the limit $e_* = \lim_{t\to\infty}|e(x(t), y(t))|$. If $e_* = 0$, then statement 1 holds due to Assumption 2.1. Otherwise, $e\Phi(e)$ is uniformly positive and thus, recalling that $\cos\delta(t) \to 1$ as

$t \to \infty$, one obtains that $\int_0^\infty |n(x(t), y(t))| dt < \infty$, which implies statement 2 thanks to Lemma 2.4. □

Corollary 3.5. *If $\mathcal{C}_0 = \varnothing$, then for any initial condition the solution of* (3.9) *is infinitely prolongable and the algorithm solves the path following problem*

$$\operatorname{dist}(r(t), \mathcal{P}) \xrightarrow[t\to\infty]{} 0.$$

Corollary 3.5 is applicable to linear mappings $\varphi(x, y) = ax + by + c$ (with $|a| + |b| \neq 0$) and many other functions, e.g. $\varphi(x, y) = y + f(x)$. These functions, however, usually correspond to *unbounded* desired curves, whereas for closed paths the GVF m_d is usually not globally defined.

The experiments show that under the assumptions of Corollary 2.11 the robot always "evades" the finite set of critical points and converges to the desired trajectory. This looks very natural since after very fast transient dynamics the robot "almost precisely" follows some integral curve, which leads to $\mathcal{P}$ "almost surely". We formulate the following hypothesis.

Hypothesis. *Under the assumptions of Corollary 2.11, for almost all initial conditions $(x(0), y(0), \alpha(0))$ the robot's trajectory $(x(t), y(t))$ converges to the desired path $\mathcal{P}$.*

Whereas the proof of this hypothesis remains a challenging problem, it is possible to guarantee the global existence of the solutions and their convergence to the desired path in some broad *invariant set*, free of the critical points. The corresponding result, which does not rely on the assumptions of Corollary 2.11, is established in the next subsection.

3.2.4 An invariant set, free of critical points

In this subsection we give a sufficient condition, guaranteeing that a solution of (3.9) does not converge to $\mathcal{C}_0$. This criterion requires the initial condition $(x(0), y(0), \alpha(0))$ to belong to some *invariant* set, free of critical points. Similar restrictions arise in most of the path following algorithms; for example, in the projection-based algorithms the convergence can be rigorously proved only in some region of attraction where the projection to the desired curve is well defined [61].

Assumptions 2.1 and 2.2 imply the uniform positivity of the error on $\mathcal{C}_0$, that is, the following inequality holds

$$e_c = \inf\{|e(x, y)| : (x, y) \in \mathcal{C}_0\} > 0. \tag{3.14}$$

Consider the following set

$$\mathcal{M} = \{(x, y, \alpha) : n \neq 0, |\delta| < \arctan(k_n e_c), |e| < e_c\}. \tag{3.15}$$

Recall that k_n is the constant parameter from (2.9) and $\delta = \delta(x, y, \alpha)$ is the angle between the robot's heading and the vector field direction (see Fig. 4.1). By definition, $\mathcal{M} \cap \mathcal{C}_0 = \varnothing$. The following lemma states that in fact $\mathcal{M}$ is an *invariant* set, i.e. any solution starting in $\mathcal{M}$ remains there.

Theorem 3.6. *Any solution of* (3.9)*, starting at* $(x(0), y(0), \alpha(0)) \in \mathcal{M}$*, does not leave* $\mathcal{M}$*, is infinitely prolongable and satisfies the inequality*

$$|e(t)| \leq \max\{|e(0)|, k_n^{-1} \tan \delta(0)\} < e_c. \tag{3.16}$$

For such a solution, one has $\operatorname{dist}(r(t), \mathcal{P}) \xrightarrow[t\to\infty]{} 0$*, i.e. the algorithm* (3.7) *solves the path following problem in* $\mathcal{M}$*.*

Proof. Consider a solution $(x(t), y(t), \alpha(t))$, starting at $(x(0), y(0), \alpha(0)) \in \mathcal{M}$. Due to (3.8), one has $|\delta(t)| \leq |\delta(0)| < \frac{\pi}{2}\ \forall t \geq 0$, and hence $\cos\delta(t) > 0$. By noticing that $\Phi(e)e \geq 0$ and thus $|\Phi(e)| = \Phi(e)\operatorname{sign} e$, one has

$$\begin{aligned} -\dot{V} &\overset{(3.12)}{=} -u_r|\Phi(e)|\operatorname{sign} e\,|n|(e\cos\delta + \sin\delta) = \\ &= u_r|\Phi(e)||n|(|e| + \operatorname{sign} e \tan\delta)\cos\delta \geq \\ &\geq u_r|\Phi(e)|\,|n|(|e| - |\tan\delta|)\cos\delta. \end{aligned}$$

In particular, $\dot{V} \leq 0$ whenever $|e| \geq |\tan\delta|$.

Notice that if $|e(t)| \geq |\tan\delta(t)|$ for any t (where the solution exists), then, obviously $|e(t)| \leq |e(0)|$ and thus (3.16) holds. Suppose now that $|e(t_0)| < |\tan\delta(t_0)|$ at some $t_0 \geq 0$. We are going to show that $|e(t)| \leq |\tan\delta(t_0)| \leq |\tan\delta(0)|$ for any $t \geq t_0$. Indeed, had we $|e(t_1)| > |\tan\delta(t_0)|$ at some point t_1, there would exist $t_* < t_1$ such that $|e(t_*)| = |\tan\delta(t_0)|$ and $|e(t)| > |\tan\delta(t_0)|$ as $t \in (t_*; t_1]$. Using (3.8), it can be easily shown that $|\tan\delta(t_0)| \geq |\tan\delta(t)|$ for $t \geq t_0$, and hence $|e(t)| > |\tan\delta(t)|$ is non-increasing when $t \in (t_*; t_1]$, which contradicts the assumption that $|e(t_1)| > |\tan\delta(t_0)| = |e(t_*)|$. We have proved that in both cases 1) and 2) the inequality (3.16) holds at any point where the solution exists; thus the solution stays in $\mathcal{M}$. By definition (3.14) of e_c, the vector $r(t)$ cannot converge to $\mathcal{C}_0$ in finite or infinite time, i.e. for the considered solution the statement in Theorem 3.4 holds. □

Remark 3.7. The condition $(x(0), y(0), \alpha(0)) \in \mathcal{M}$ restricts the robot to be "properly" headed in the sense that

$$|\delta(x(0), y(0), \alpha(0))| < \arctan(k_n e_c) < \pi/2. \tag{3.17}$$

Since $e_c > 0$, (3.17) is valid for sufficiently large k_n whenever $|\delta| < \pi/2$. In other words, Theorem 3.6 guarantees convergence to the path from any starting

position with $|e(0)| < e_c$ and $|\delta(0)| < \pi/2$, choosing large k_n. Furthermore, if the desired path $\mathcal{P}$ is *closed* and the direction of circulation along it is unimportant, the condition $|\delta(0)| < \pi/2$ can be provided by reverting the vector field $m_d \mapsto -m_d$ (which corresponds to the replacement of $\varphi \mapsto -\varphi$ and $|\delta| \mapsto \pi - |\delta|$) unless $\delta(0) = \pm\pi/2$ (in practice it is never possible).

Remark 3.8. Even if (3.17) is violated at the starting time $t = 0$, it obviously holds when $t > t_0 = k_\delta^{-1}[\ln|\delta(0)| - \ln\arctan(k_n e_c)]$ thanks to (3.8). Thus, if one is able to prove that the solution is prolongable up to t_0 and $|e(x(t_0), y(t_0))| < e_c$, Theorem 3.6 provides the convergence of the robot's position to the desired path.

Remark 3.8 suggests the way to relax the restriction on the initial robot's orientation (3.17). Since the robot moves at the constant speed $u_r > 0$, it covers the distance $u_r t_0$ until its orientation satisfies (3.17). The path following is guaranteed if $u_r t_0$ is less than the *viability distance* of the initial position, i.e. the distance from it to the set where $|e| \geq e_c$.

Definition 3.9. Given a point $r_0 = (x_0, y_0)$ with $|e(r_0)| < e_c$, the number $d_0 = \inf\{\mathrm{dist}(r_0, r) : e(r) \geq e_c\} > 0$ is said to be its *viability distance*.

Theorem 3.6 and Remark 3.8 yield in the following.

Corollary 3.10. *Let the initial position of the robot $r(0) = (x(0), y(0))$ with $e(r(0)) < e_c$ have the viability distance $d_0 > 0$. If this viability distance satisfies the condition*

$$d_0 > \frac{u_r}{k_\delta} \ln \frac{|\delta(x(0), y(0), \alpha(0))|}{\arctan(k_n e_c)}, \tag{3.18}$$

then the solution of (3.9) *gets into the set $\mathcal{M}$ in finite time, and is prolongable up to ∞ satisfying $\mathrm{dist}(r(t), \mathcal{P}) \xrightarrow[t\to\infty]{} 0$.*

Note that (3.18) holds for any $\alpha(0)$ under the assumption

$$d_0 > \frac{u_r}{k_\delta} \ln \frac{\pi}{\arctan(k_n e_c)}. \tag{3.19}$$

The condition (3.19) gives an estimate for the region, starting in which the robot necessarily converges to the desired path $\mathcal{P}$. Taking the original orientation relative to the field into account, this estimate can be tightened by using (3.18). Typically d_0 is *uniformly* positive in the vicinity of $\mathcal{P}$ (this holds, for instance, when $\mathcal{P}$ is a closed curve). Thus (3.19) guarantees the algorithm to converge in the vicinity of $\mathcal{P}$, choosing k_δ/u_r sufficiently large. As was mentioned, practical experiments with natural trajectories, satisfying the assumptions of Corollary 2.11, show that the robot is always attracted to the desired path, although the mathematical proof of this remains a non-trivial task.

Remark 3.11. Although explicit calculation of the viability distance is complicated, conditions (3.18) and (3.19) in fact require only its *lower* bound. Such a bound can be explicitly obtained, for instance, if the error is Lipschitz

$$|\nabla e(r)| = \psi'(\varphi(r))|n(r)| \le c = const \quad \forall x, y. \tag{3.20}$$

If the condition (3.20) holds, then $|e(r) - e(r_0)| \le c|r - r_0|$ and hence the viability distance of r_0 is estimated as follows

$$d_0 \ge (e_c - e(x_0, y_0))/c.$$

Condition (3.20) can often be provided under an appropriate choice of $\psi(\cdot)$. For instance, when $\varphi(x, y) = (x - x_0)^2 + (y - y_0)^2 - R^2$ (circular path), one can choose $\psi(s) = \arctan s$ so that $\psi'(s) = 1/(1 + s^2)$. More generally, if $|\varphi(r)| \ge C_1|r|^\beta$ and $|\nabla\varphi(r)| \le C_2|r|^\gamma$ as $|r| \to \infty$, then ∇e is globally bounded; we choose $\psi(s) = \arctan(s^p)$, where $p > 0$ is sufficiently large so that $p - 1 + \gamma \le 2\beta p$.

3.3 Discussion and Extensions

In this section, we compare the proposed method with the existing path following control algorithms. Potential extensions of the proposed method are also discussed.

3.3.1 Comparison with other GVF algorithms

The vector-field path following algorithms have been mainly considered for straight lines and circular paths [67, 92]. The GVF designed for these paths is the same as constructed in Section III; the only difference is the path following controller: unlike [67], we do not consider sliding-mode algorithms and confine ourselves to simple linear controllers. A general approach to the Lyapunov-based GVF design has been suggested in [53]. The desired path is supposed to be the set of the global minima of some predefined Lyapunov function; e.g. for the path (4.3), our usual Lyapunov function (2.11) can be chosen. Departing from this Lyapunov function, a broad class of GVF has been suggested, including (2.14) as a special case. The paper [53] also offers a methodology of proving the convergence of integral curves to the desired path (considered in our Section 2.4) and design of path following controllers. For the case of unicycle-like robot the control algorithm, suggested in [53, Eq. (35)], is equivalent to our controller (3.8). Describing a general framework, the paper [53] focuses however on some specific problems of flight control, namely, following a *loiter circle* (planar circle at the fixed altitude) or some other planar path, which can be obtained from a loiter circle by a smooth "warping" transformation, preserving the vector field's properties. Many assumptions adopted

in [53] are restrictive, e.g. the Lyapunov function should be radially unbounded, excluding straight lines and other unbounded paths from the consideration. The analysis of the closed-loop system in [53] is not fully rigorous as it completely ignores the problem of the solution existence up to ∞, which becomes non-trivial when $\mathcal{C}_0 \neq \varnothing$.

Unlike the previous papers [53, 67, 92], we consider a general smooth path (4.3), which can be closed or unbounded; for closed trajectories, we do not restrict the Lyapunov function to be radially unbounded. We provide a mathematically rigorous analysis of our algorithm and specify an explicit set of initial conditions, starting at which the robot evades the critical points and converges to the desired trajectory.

3.3.2 The GVF method vs. alternative path following algorithms

It should be noticed that many approaches to path following, discussed in Introduction, are not applicable to general smooth paths of non-constant curvature and nonholonomic robots; for example, various virtual target approaches [89] typically require to control the robot's velocity. We compare the GVF method with two commonly used algorithms [92]: the projection-based line-of-sight (LOS) method [14, 26] and the "nonlinear guidance law" (NGL) [75, 92].

The LOS approach assumes the existence of the unique *projection* of the robot location r onto the path $\mathcal{P}$, that is, the point P closest to r in the Euclidean metric. The distance to the path $d = d(r, P)$ serves as the tracking (or "cross-track") error. The desired robot's orientation α_{LOS} is the absolute bearing to the unique point P_Δ (Fig. 3.1), such that the vector $\overrightarrow{PP_\Delta}$ has length $\Delta > 0$ and is tangent to $\mathcal{P}$ at the projection point. The direction of this vector is the desired direction of path following (in our example, the robot circulates along the path clockwise). The controller (3.7) is replaced by

$$\omega_{LOS} = c(P)u_r - k_{LOS}\left(\alpha - \alpha_{LOS}\right),$$

where $k_{LOS} > 0$ is a constant gain and $c(P)$ is the curvature of the curve $\mathcal{P}$ at the projection point P. The parameters of the algorithm are the *lookahead distance* Δ and the gain k_{LOS}.

The second path following algorithm to be compared with the GVF method is the "nonlinear guidance law" (NGL) [75, 92]. At the current robot position r, draw a circle of radius R. The circle intersects the path $\mathcal{P}$ at two points q, q'. It is supposed that the robot moves sufficiently close to the desired path and the direction of the path following is fixed, so it is always possible to choose the intersection point, lying *ahead* of the robot on the path $\mathcal{P}$ (in Fig. 3.1, this is point q). The absolute bearing α_R to this point is the desired direction of motion. The

angular velocity controller is designed as

$$\omega_R = -k_R\,(\alpha - \alpha_R)\,.$$

The constants $R > 0, k_R > 0$ are the algorithm's parameters.

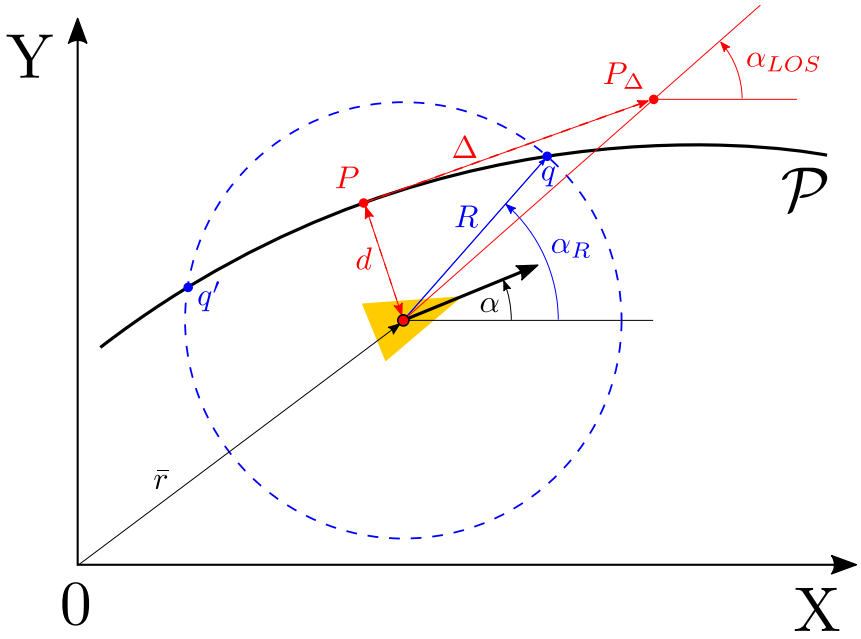


Figure 3.1: Geometric objects, employed by the LOS (red) and the NGL (blue) path following algorithms

For the sake of comparison, we will use the same elliptic path, determined by the function (5.1) and the same parameters of the GVF algorithm as in Subsect. 5.1-A. The initial position and heading of the robot are

$$x = 200; y = 450; \alpha = 0.0278.$$

It is clear that any path following controller is very sensitive to the parameter choice, and varying the parameters one can always make the convergence rate faster or slower. To compare the behavior of different algorithms, one thus needs to choose parameters, providing approximately the same convergence rate; in our situation, the dynamics of the Euclidean distance $d(t) = \mathrm{dist}(r(t), \mathcal{P})$ is similar to the GVF method, choosing $k_{LOS} = 2$ and $\Delta = 70$ in the LOS method and $k_R = 2$ and $R = 40$ in the NGL algorithm (Fig. 3.2). As one can see, the GVF method gives much better transient behavior than the other algorithms, which have visible

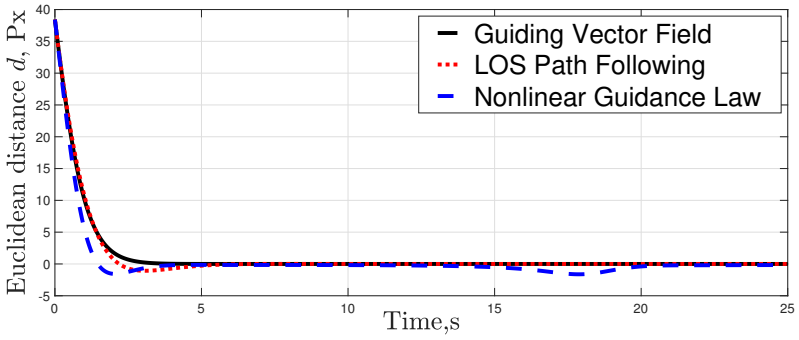


Figure 3.2: The dynamic of the Euclidean distance d

overshoots. To eliminate the overshoot of the LOS method, one has to increase the lookahead distance, but in this case the convergence to the desired path becomes slower. The NGL method has the largest overshoot; using this method, it is also impossible to eliminate completely the tracking error since the controller has no information about the changing curvature of the path. To decrease the overshoot, one has to increase the radius R; this however amplifies the oscillations in the tracking error d. Thus the GVF method demonstrates better performance than the considered alternative methods.

3.4 Conclusions

In this section, we have proposed a new algorithm for path following control of nonholonomic robots exploiting the idea of a guiding vector field. Unlike the existing results, the desired path can be an arbitrary smooth curve in its implicit form, i.e. the zero set of a smooth function. We have examined mathematical properties of the algorithm and given global conditions for following asymptotically the desired path.

Chapter 4
The moving path following

4.1 Introduction

The traditional problem setup for path following assumes that the path is fixed in space. However, there are applications for which it is useful to follow a path that is attached to an external reference frame that moves independently with respect to a global inertial coordinate frame [70]. In principal, the ability to modify easily the spatial position and the orientation of the entire path may significantly increase flexibility and usability of mobile robots. However, for a number of practical tasks the mobility of a spatial configuration is the inherent feature. The typical example is the tracking of a ground target by a fixed-wing UAV. In performing such missions, since the fixed-wing UAV generally moves faster than the ground target and also this type of vehicles cannot hover in place, the aircraft requires to follow a special moving pattern. For this, the circular *standoff tracking* concept is introduced [69]. These circular flight formations are recommended for various target tracking applications, in which for each UAV the maximum altitude flight ensures the maximum visibility and the minimum radius turn ensures the minimum distance to the target at the maximum altitude. Many applications, however, require to use more sophisticated curves than combinations of straight lines and circles. Such tasks include atmospheric monitoring by UAVs [83] and the ocean sampling by underwater gliders [72]. In particular, for the problem of the cloud monitoring [83] the desired path can be obtained by enclosing the shape of the cloud by a smooth curve. However, to perform this task successfully, one has to take into account a drift of the cloud due to the influence of the wind. Moreover, because of the presence of wind it becomes necessary to consider impact of disturbance in the robot's model.

The method of path following, presented in this chapter, is an extension of the guidance algorithm for nonholonomic robots given in Chapter 3 to the case where a *general C^2-smooth curvilinear path* is attached to an external reference frame that moves uniformly with respect to a global inertial coordinate frame. The robot's motion is influenced by a constant disturbance. The presented algorithm is based on the idea of following a vector field that converges smoothly to the desired path, where the convergence is global under some assumptions. Instead of considering

the Euclidean distance, the notion of error is given by the implicit equation of the desired trajectory, making the tracking task much easier to be implemented.

The chapter is organized as follows. The path following problem is formulated in Section 4.2, and Section 4.3 offers the vector field algorithm for path following. This algorithm is validated by experiments described in Section 5.1.

4.2 Problem statement

Similar to Chapter 3 we focus on the planar motion. Therefore, we consider the unicycle-type model where the longitudinal speed is a predefined constant $s > 0$:

$$\begin{cases} \dot{p}(t) &= sm(t) + w, \\ \dot{\alpha}(t) &= \omega(t). \end{cases} \tag{4.1}$$

Here $p(t) = [x(t)\ y(t)]^\top \in \mathbb{R}^2$ stands for the position of the robot's center of gravity in the inertial Cartesian frame $\mathcal{O}_N$, $\alpha(t)$ is the orientation of the robot in this frame, $m(t) = [\cos(\alpha(t))\ \sin(\alpha(t))]^\top$ is the unit orientation vector, $w \in \mathbb{R}^2$ is a *constant* vector representing the disturbance in the inertial frame $\mathcal{O}_N$. We assume the angular velocity ω to be the only control input to the system.

We also assume the existence of a suitable state estimation scheme that provides

i. the position $p(t)$ and its derivative $\dot{p}(t)$ with respect to $\mathcal{O}_N$, from e.g. a global positioning system (GPS) signal,

ii. the orientation of the robot $\alpha(t)$ with respect to $\mathcal{O}_N$, which can be obtained from a well calibrated compass in areas far away from the Earth's poles and

iii. the instant speed s in the body-fixed frame.

To focus on the ideas of the path following algorithm, we confine ourselves to the simplest case of an external frame motion with a constant translational velocity. For the sake of clarity and simplicity we do not consider the rotational motion of $\mathcal{O}_T$. The kinematics of the external frame $\mathcal{O}_T$ with respect to $\mathcal{O}_N$ is represented by the following equations:

$$\begin{cases} \dot{p}_T(t) &= v_T, \\ \dot{\alpha}_T(t) &= 0, \end{cases} \tag{4.2}$$

where $p_T(t) = [x_T(t)\ y_T(t)]^\top \in \mathbb{R}^2$ stands for the position of the origin of the external frame $\mathcal{O}_T$ with respect to the frame $\mathcal{O}_N$, $v_T \in \mathbb{R}^2$ is the constant vector of the velocity, α_T is the orientation of the external frame $\mathcal{O}_T$ with respect to the frame $\mathcal{O}_N$ and $\dot{\alpha}_T \equiv 0$ is the angular speed of the external frame $\mathcal{O}_T$ with respect to the frame $\mathcal{O}_N$. We assume that v_T is known.

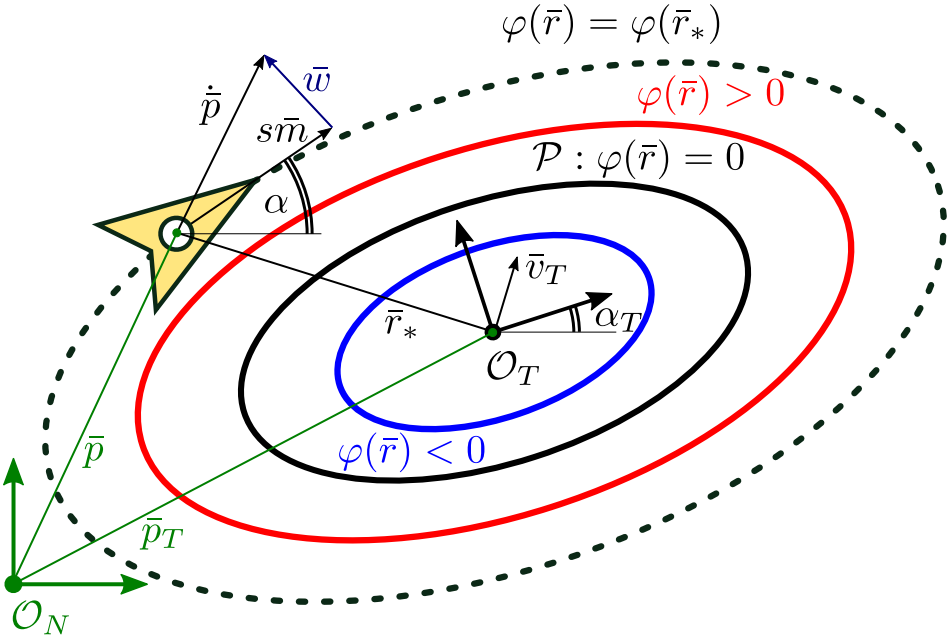


Figure 4.1: The robot orientation and level sets of the function $\varphi(r)$

The robot's position with respect to the frame $\mathcal{O}_T$ can be defined in the following way

$$r(t) = R^\top(\alpha_T)(p(t) - p_T(t)),$$
$$R(\alpha_T) = \begin{bmatrix} \cos(\alpha_T) & -\sin(\alpha_T) \\ \sin(\alpha_T) & \cos(\alpha_T) \end{bmatrix},$$

where $r(t) = [r_x(t)\ r_y(t)]^\top \in \mathbb{R}^2$ is the relative position of the robot with respect to the origin of the frame $\mathcal{O}_T$, $R(\alpha_T)$ is the rotational matrix characterizing the orientation of $\mathcal{O}_T$ with respect to $\mathcal{O}_N$.

As it was discussed in 2.3 a curvilinear desired path $\mathcal{P} \subset \mathbb{R}^2$ is described by the implicit equation

$$\mathcal{P} = \{r : \varphi(r) = 0\}, \tag{4.3}$$

where the function $\varphi : \mathbb{R}^2 \to \mathbb{R}$ is C^2-smooth. The principal restriction imposed by our approach is *regularity*: in some vicinity of $\mathcal{P}$ one has

$$\|\nabla\varphi(r)\| = \left\| \left[\frac{\partial\varphi(r)}{\partial r_x}; \frac{\partial\varphi(r)}{\partial r_y} \right]^\top \right\| \neq 0, \tag{4.4}$$

where $\|\cdot\|$ is the Euclidean norm of the vector.

As illustrated in Fig. 4.1, the plane $\mathbb{R}^2$ is covered by the disjoint *level sets* of the function φ, that is, the sets where $\varphi(r) = c = const$. The path is one of the level sets, corresponding to $c = 0$; the value $\varphi(r)$ can be considered as a (signed) "distance" from the robot to the path (differing, as usual, from the Euclidean distance). The main difference in the problem statement of the moving path following control is that the desired path and correspondent tracking error are defined in the "relative" coordinates with respect to the origin of the frame $\mathcal{O}_T$. Similarly to the results of Chapter 2 and Chapter 3, consider a function $e = \psi[\varphi(r)]$ as a tracking error.

Problem 2. *To design a moving path following algorithm, i.e. a causal feedback law* $(r_x(\cdot), r_y(\cdot), \alpha(\cdot)) \mapsto \omega(\cdot)$*, which eliminates the tracking error* $|e(t)| \xrightarrow[t\to\infty]{} 0$*, bringing thus the robot to the predefined path* $\mathcal{P}$*.*

Below we will use the following auxiliary relations. The derivation of a rotational matrix can be found in the following form

$$\dot{R}(\alpha(t)) = -\omega(t) E R(\alpha(t)), \tag{4.5}$$

$$E = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

Here E is a skew-symmetric matrix with properties

$$EE = -I_2, \quad R^\top E R = E,$$

where I_2 is the 2×2 identity matrix. Similarly, the derivation of the unit orientation vector can be found in the following form

$$\dot{m}(t) = -\omega(t) E m(t). \tag{4.6}$$

We can determine the first and second derivatives of the relative position vector:

$$\begin{cases} \dot{r}(t) &= R^\top(\alpha_T)\,(\dot{p}(t) - v_T) = R^\top(\alpha_T)(sm(t) + w), \\ \ddot{r}(t) &= -\omega(t) s E R^\top(\alpha_T) m(t). \end{cases} \tag{4.7}$$

In addition, for the sake of simplicity in the analysis, we consider the following realistic assumption.

Assumption 4.1. The constant speed s is greater than the sum of the Euclidean norms of the disturbance vector w and the velocity vector v_T, i.e. $s > \|w\| + \|v_T\|$.

In fact, it is quite straightforward to realize that this assumption is necessary if one wants to reach a desired path from almost every initial position.

4.3 The steering algorithm

The path following control presented in this section is based on two steps. The first one is about constructing a guidance vector field such that once the robot is tracking it, the vehicle will converge to $\mathcal{P}$. The second step deals with the task of steering the robot in order to converge to such a guiding vector field.

4.3.1 Design of the vector field

The design procedure of the guiding vector field is identical to the approach presented in Section 2.4. We define $n(r) := \nabla\varphi(r)$ as the normal vector to the curve corresponding to the level set $\varphi(r)$ at the point r and the tangent vector τ at the same point r is given by the rotation

$$\tau(r) = En(r).$$

Using the quadratic potential function

$$V(r) = \frac{1}{2}e^2(r) \tag{4.8}$$

one can derive the vector field v identical to (2.9)

$$v(r) = \tau(r) - k_n e(r)n(r), \quad k_n = const > 0. \tag{4.9}$$

The desired orientation is obtained normalizing (4.9)

$$m_d(r) = \frac{v(r)}{\|v(r)\|}, \quad \|v(r)\| \neq 0. \tag{4.10}$$

Fig. (4.2) illustrates the relations between the vectors $r(t)$, $m(t)$, $\tau(r)$, $n(r)$, $v(r)$ and $m_d(r)$.

The controller offered in the next subsection employs the field vector $m_d(r)$ at any point as well as its derivative along the trajectory $\dot{m}_d$. Differentiating (4.9), one arrives at

$$\dot{v} = (E - k_n eI_2)\, H\dot{r} - k_n \dot{e}n(r), \tag{4.11}$$

$$\dot{e} = \psi'(\varphi(r))\, n^\top(r)\dot{r}(t), \tag{4.12}$$

$$\dot{m}_d = \frac{d}{dt}\frac{v}{\|v\|} = -\frac{1}{\|v\|}Em_d m_d^\top E\dot{v}. \tag{4.13}$$

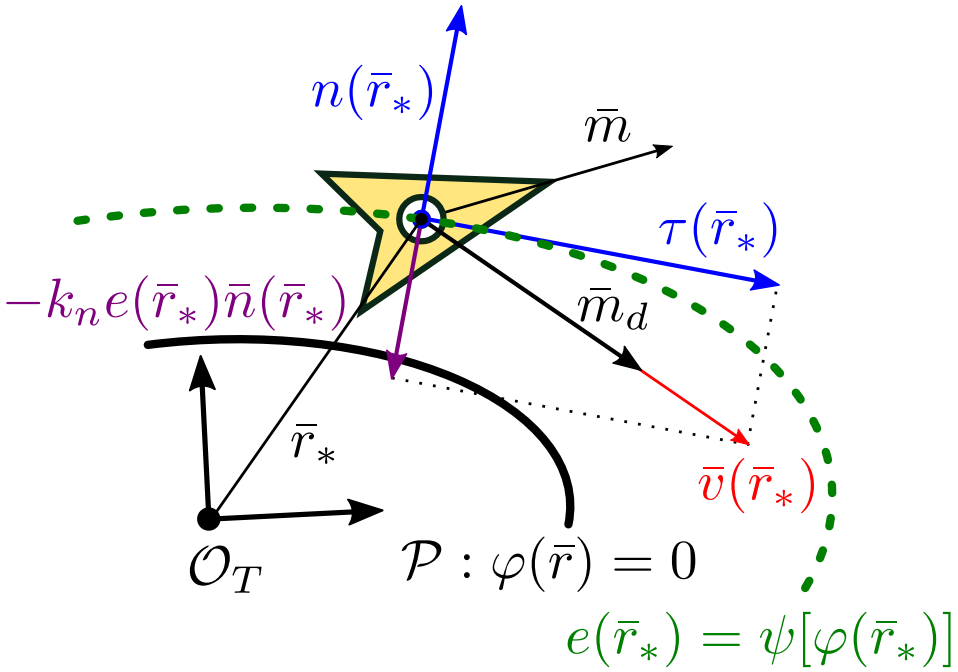


Figure 4.2: The construction of the *guiding vector field*

Since $\|\bar{m}_d\|^2 = 1$, one has $\bar{m}_d^\top \dot{\bar{m}}_d = 0$, and hence

$$\boxed{\dot{\bar{m}}_d = -\omega_d E \bar{m}_d \Longleftrightarrow \omega_d = -\dot{\bar{m}}_d^\top E \bar{m}_d,} \tag{4.14}$$

where ω_d is a scalar function that can be explicitly found from (4.11)-(4.13).

4.3.2 Path following controller design

Now we are going to present the algorithm solving Problem 2 by making the robot converge to the guidance vector field defined in (4.10).

The relative velocity $\dot{r}$ can be trivially decomposed as

$$\dot{r}(t) = \|\dot{r}(t)\| m_r(t), \tag{4.15}$$

where $\|\dot{r}(t)\|$ is an instant relative speed and $m_r(t)$ is the unit orientation vector. To find the relation between the relative velocity and the control input u, one has to find the derivation of the vector $m_r(t)$ using (4.7)

$$\dot{m}_r(t) = -\frac{1}{\|\dot{r}(t)\|} E m_r(t) m_r^\top(t) E \ddot{r}(t) = -\omega(t) \frac{s}{\|\dot{r}(t)\|} m_r^\top(t) R^\top(\alpha_T) m(t) E m_r(t).$$

Now, it is possible to introduce the new control variable u^* in the following way

$$u^*(t) = \omega(t) \frac{s}{\|\dot{r}(t)\|} m_r^\top(t) R^\top(\alpha_T) m(t). \tag{4.16}$$

Thus, one can rewrite the equations above in the compact form

$$\boxed{\dot{m}_r(t) = -u^*(t) E m_r(t).} \tag{4.17}$$

Comparing with (4.6) it is clear that the new control input u^* characterizes the angular speed of the vector $m_r(t)$. Although Assumption 4.1 is quite conservative, it guarantees that the changing of coordinates (4.16) is well-defined.

As we have discussed in Section 3.2.2, the idea of the path following algorithm is to steer the robot along the guiding vector field m_d. We introduce the directed angle $\delta \in (-\pi; \pi]$ between m_d and m_r. The function δ is being C^1-smooth on the set of points where $m_r \neq -m_d$. By noticing that $\sin\delta = -m_r^\top E m_d$ and $\cos\delta = m_r^\top m_d$, at any time $t \geq 0$ when $m_r \neq -m_d$ (that is, $\delta < \pi$) one has

$$-u^*(t) E \bar{m}_r(t) = \dot{\bar{m}}_r(t) = -\left(\dot{\delta}(t) + \omega_d(t)\right) E \bar{m}_r(t) \Longleftrightarrow \dot{\delta}(t) = u^*(t) - \omega_d(t).$$

Calculating ω_d at any point, the control input is

$$\boxed{u^*(t) = \omega_d(t) - k_\delta \delta(t).} \tag{4.18}$$

Here $k_\delta > 0$ is a constant, determining the convergence rate. Collecting together formulas (4.14), (4.16) and (4.18) we obtain the final form of the control law:

Algorithm 2: Moving Path Following

$$\omega(t) = \frac{\|\dot{r}(t)\|}{s} \frac{\left(-\dot{m}_d^\top(t) E m_d(t) - k_\delta \delta(t)\right)}{m_r^\top(t) R^\top(\alpha_T) m(t)}. \tag{4.19}$$

Here $\dot{r}(t)$ is found from (4.7); $\dot{m}_d(t)$ is found from (4.13); $m_d(t)$ is found from (4.10); and $m_r(t)$ is found from (4.15).

4.3.3 Properties of the solutions

In comparison with the original solution of the path following problem for the driftless nonholonomic model in Chapter 3, the additional components such as the constant disturbance vector w and the kinematic of the moving frame $\mathcal{O}_T$

have been added. However, using the change of coordinates (4.7) and (4.16) and Assumption 4.1, one transforms the moving path following problem to the standard path-following problem considered in the Chapter 3. Hence, all results in Chapter 3 are valid for the problem in question.

4.4 Numerical simulations

In this section we are going to present two cases. The first result demonstrates the performance of the guidance algorithm in case of the fixed external frame. The second experiment demonstrates the numerical simulation of the moving path following problem.

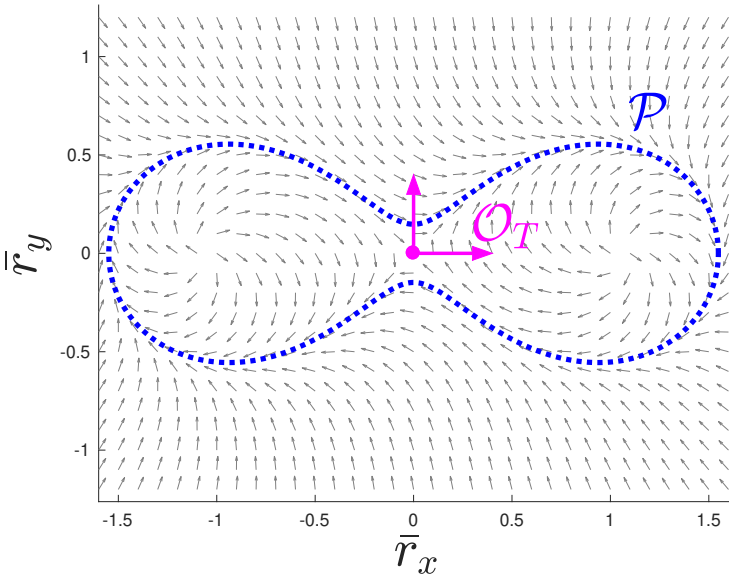


Figure 4.3: The GVF for the Cassini oval (4.20)

4.4.1 The Cassini oval in the moving frame

In this case, we consider the standard unicycle model (4.1) with the constant speed $s = 1$ and the disturbance vector $w = [0.5; 0]^\top$. We define the function $\varphi(r)$

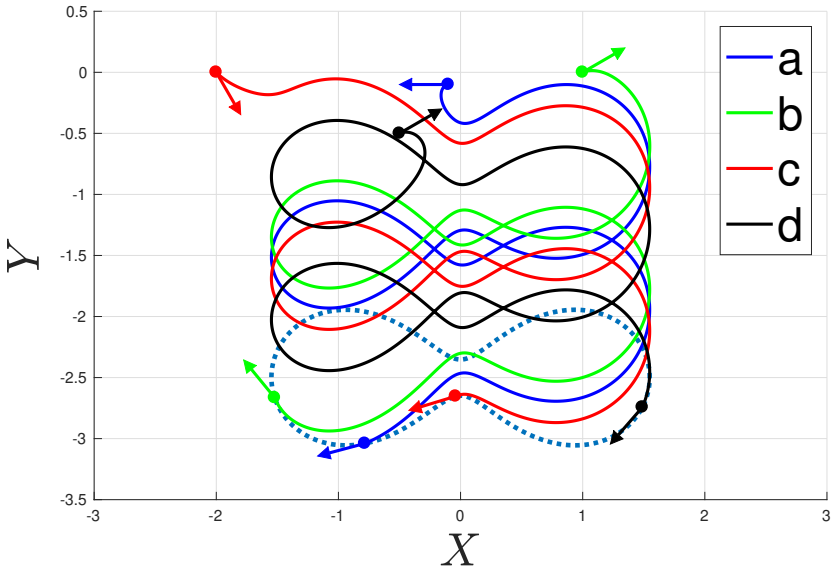


Figure 4.4: The resulting robot's trajectories corresponding to different initial conditions (4.21)

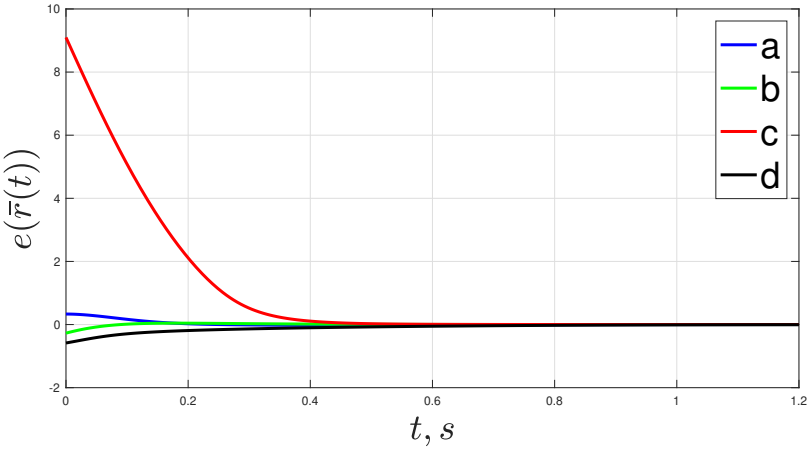


Figure 4.5: The dynamics of the tracking error $e(r(t))$

generating our desired path $\{\mathcal{P} : \varphi(r) = 0\}$ as follows

$$\varphi(r) = (r_x^2 + r_y^2)^2 - 2c^2(r_x^2 - r_y^2) - a^4 + c^4, \tag{4.20}$$

where $a = 1.1$, $c = 1.09$ are parameters of the curve. The initial position of the frame $\mathcal{O}_T$ is $p_T(0) = [0; -0.5]^\top$ m and corresponding orientation $\alpha_T = 0$ rad; the translational velocity has been chosen as $v_T = [0; -0.2]^\top$.

We choose $k_n = 1$, $k_\delta = 10$ and define the tracking error in the following form $e(r) = \psi(\varphi(r)) = \varphi(r)$. The GVF (2.14) for this case is depicted in Fig. 4.3.

Fig. 4.4 illustrates four trajectories (labeled a, b, c and d), corresponding to the initial conditions

$$\begin{aligned} a &: (x = -0.1, y = -0.1, \alpha = -\pi), \\ b &: (x = 1.0, y = 0.0, \alpha = \pi/6), \\ c &: (x = -2.0, y = 0.0, \alpha = -\pi/3), \\ d &: (x = -0.5, y = -0.5, \alpha = \pi/6). \end{aligned} \tag{4.21}$$

The corresponding tracking errors are displayed in Fig. 4.5.

4.5 Conclusions

In this chapter we offer an extension of the guiding vector field algorithm for a moving path-following control of nonholonomic robots in presence of disturbances. We considered the problem when a desired path is given with respect to the external coordinate frame that is uniformly moving. The guidance strategy is based on following a reference vector field generated from the implicit representation of the desired path. The results have been experimentally validated, using numerical simulation and the E-puck robot (see Chapter 5).

Chapter 5
Experimental validation

5.1 Experiments with wheeled robot

In this section the experiments with real robots are reported. In one of these experiments, the desired path is an ellipse, and the other experiments deals with the more sophisticated Cassini oval (see Section 2.4).

We test the results using the E-puck mobile robotic platform [65]. The experimental setup consists of the differential wheeled E-puck robot, a server computer, an overhead camera and a communication module. The robot is identified by a data matrix marker on its top (see Fig. 5.1). The position of the central point and the orientation of the marker are recognized by a vision algorithm running at a control computer connected to the overhead camera. The available workspace is a planar area of 2.6x2 meters covered by the image of 1280x720 Px (henceforth the acronym Px is used for pixels). For convenience we operate with pixels as coordinates. The PC runs real-time calculation of the control actions based on the pose information of the robot and computes the control inputs for the robot. The results of computation as the desired angular and linear velocities of the robot are

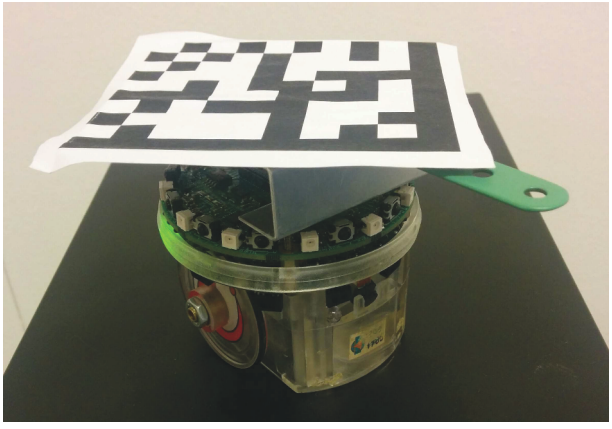


Figure 5.1: The E-puck robot with marker on top, used in experiments

translated into the commands for its left and right wheels. The commands from the control computer to the robot are sent via Bluetooth at the fixed frequency of 20 Hz.

We verify the guidance algorithm for two closed trajectories, followed by the robot in the *clockwise* direction. In both cases the forward velocity was set to $u_r = 50$ Px/s, and we set the parameters of the algorithm $k_n = 3$ and $k_\delta = 2$. We choose $\psi(s) = s$, so that $e(x, y) = \varphi(x, y)$. Below we describe the trajectories and show the numerical data from the experiments.

5.1.1 Circulation along the ellipse

For our first experiment, we choose the elliptic trajectory, defined by the function

$$\varphi(x, y) = k_s \left(\frac{(x - x_0)^2}{p^2} + \frac{(y - y_0)^2}{q^2} - R^2 \right). \tag{5.1}$$

Obviously, the level curves of φ are ellipses. The path $\mathcal{P}$ corresponds to the ellipse with semiaxes pR and qR, centered at the point (x_0, y_0).

For the experiment we choose $x_0 = 600$Px, $y_0 = 350$Px, $R = 400$Px, and the semiaxes scale factors $p = 1$, $q = 0.5$ (see Fig. 5.2). To provide $\varphi(x, y) \in [-5; 5]$ in the working area, we choose the scaling factor $k_s = 10^{-5}$. The robots's trajectories (labeled a, b, c and d) under four different initial conditions are shown in Fig. 5.2. The respective starting conditions are

$$\begin{aligned} a &: (x = 472, y = 311, \alpha = 0.0768), \\ b &: (x = \ 30, y = 555, \alpha = 0.0278), \\ c &: (x = 408, y = 369, \alpha = 2.1515), \\ d &: (x = \ 78, y = 133, \alpha = 4.0419) \end{aligned} \tag{5.2}$$

(the coordinates x, y are in pixels, and α is in radians). Fig. 5.3 illustrates the dynamics of the tracking error $\varphi(x(t), y(t))$.

In this example the set of critical points is $\mathcal{C}_0 = \{(x_0, y_0)\}$, which corresponds to $e(x_0, y_0) = -k_s R^2$ and hence $e_c = R^2$. The set $\{(x, y) : |e(x, y)| < e_c\}$ consists of all points, where

$$0 < \frac{(x - x_0)^2}{p^2} + \frac{(y - y_0)^2}{q^2} < 2R^2. \tag{5.3}$$

Theorem 3.6 guarantees that if the robot starts at any of these points and (3.17) holds, then it converges to the desired path. Using the geometry of the specific set (5.3), Remark 3.8 allows to prove convergence for many other initial conditions: it is clear, for instance, that if the robot starts from some point, lying outside the

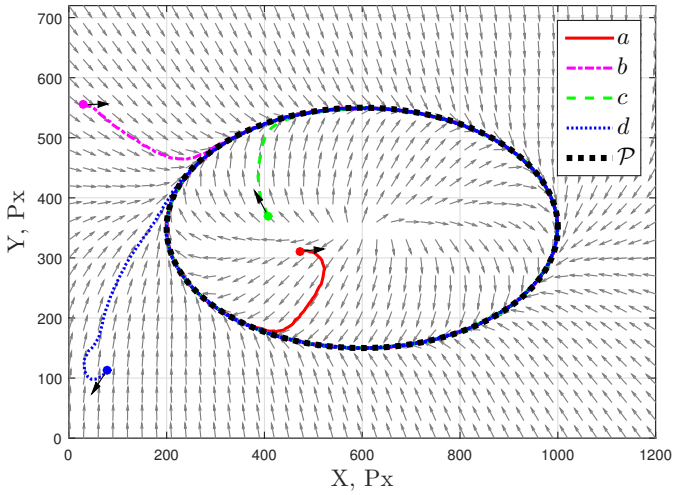


Figure 5.2: Elliptical path: the vector field and robot's trajectories.

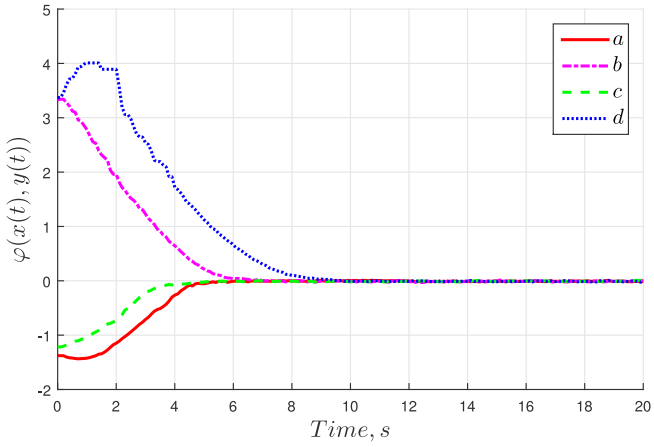


Figure 5.3: Elliptical path: the dynamics of the tracking error $e = \varphi(x, y)$.

ellipse and sufficiently far from it, then it enters into the domain (5.3) with a "proper" orientation, satisfying (3.17), and thus the problem of path following is solved. The practical experiments, however, show that the robot converges to the path from *any* point, except for the ellipse' center, independent of the initial robot's orientation, whereas the mathematically rigorous proof remains an open problem.

5.1.2 Circulation along the Cassini oval

For our second experiment the path chosen is referred to as the *Cassini oval* (Fig. 5.4) and determined by the function

$$\varphi(x,y) = k_s\left[\left(\Delta x^2+\Delta y^2\right)^2 - 2q^2\left(\Delta x^2-\Delta y^2\right) - p^4 + q^4\right],$$
$$\Delta x = (x-x_0),\ \Delta y = (y-y_0).$$

We choose here $k_s = 10^{-10}$, $p = 330$Px and $q = 300$Px. The oval is centered at $(x_0, y_0) = (600, 350)$Px.

Fig. 5.4 illustrates four trajectories (labeled a, b, c and d), corresponding to the initial conditions

$$\begin{aligned} a &: (x = 233, y = 184, \alpha = 2.9287),\\ b &: (x = 106, y = 202, \alpha = 4.2487),\\ c &: (x = 355, y = 343, \alpha = 5.4071),\\ d &: (x = 503, y = 619, \alpha = 0.1022). \end{aligned} \tag{5.4}$$

(the coordinates x, y are in pixels, and α is in radians). The corresponding tracking errors are displayed in Fig. 5.5.

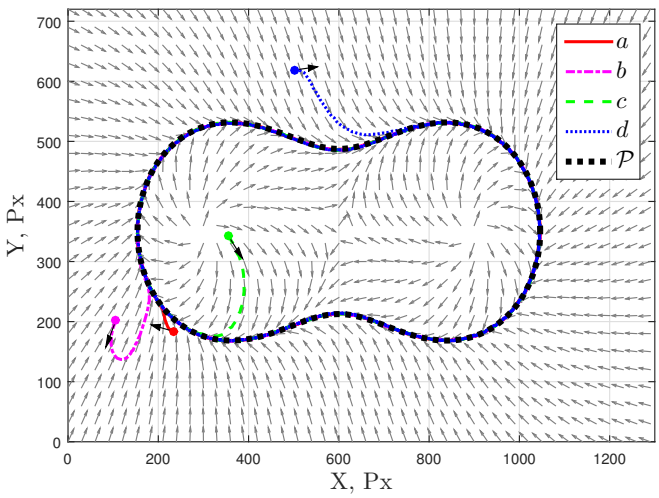


Figure 5.4: Cassini oval: the vector field and robot's trajectories.

As discussed in Section 2.4, the set of critical points is $\mathcal{C}_0 = \{(x_0 \pm q, y_0), (x_0, y_0)\}$. If $q^2 < p^2 < 2q^2$ (such inequalities are valid in our example) then it can be checked

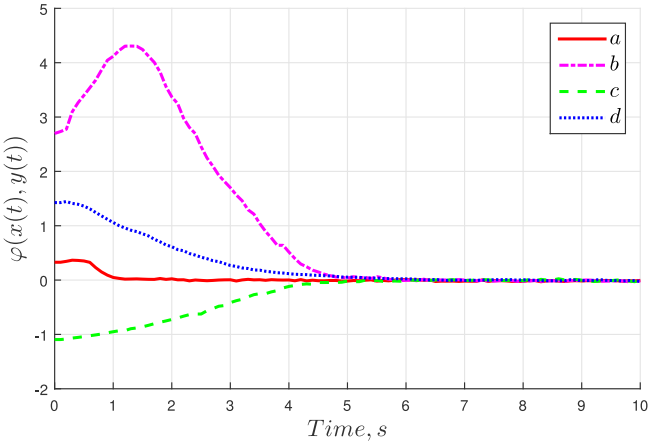


Figure 5.5: Cassini oval: the dynamics of the tracking error $e = \varphi(x, y)$.

that $e_c = |e(x_0, y_0)| = p^4 - q^4$, and the set $\{(x, y) : |e(x, y)| < e_c\}$ is a domain, shown in Fig. 5.6. Theorem 3.6 guarantees that starting at any of these points with

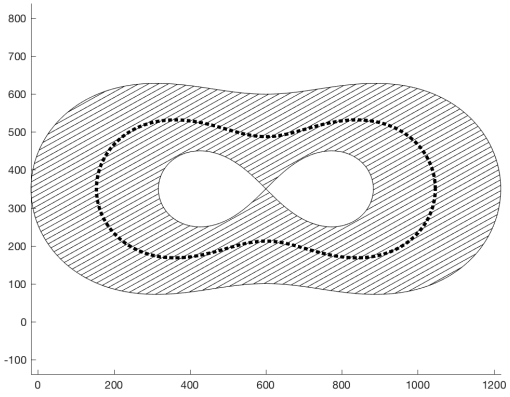
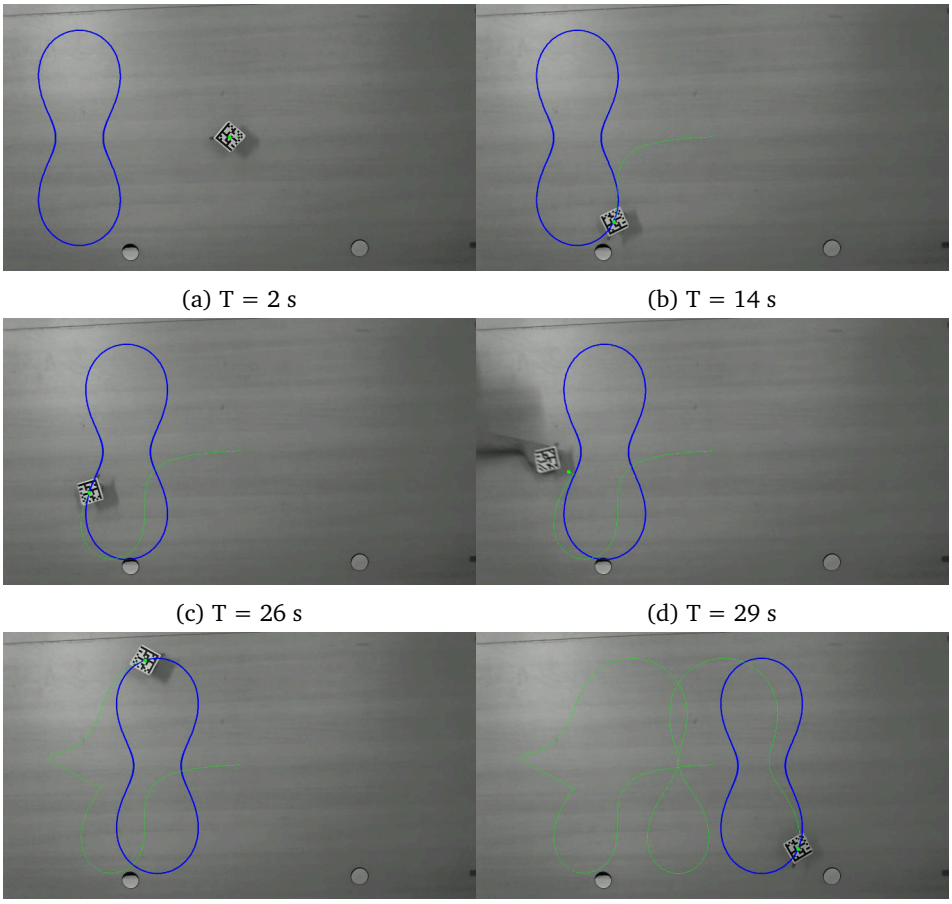


Figure 5.6: Cassini oval: the set $\{(x, y) : |e(x, y)| < e_c\}$.

the orientation satisfying (3.17), the robot converges to the desired path. Similar to the case of the elliptic path, Remark 3.8 allows to prove convergence for many other initial conditions; experiments show that in fact the robot converges to the desired path from any non-critical point.

5.1.3 Moving path following



(a) T = 2 s (b) T = 14 s

(c) T = 26 s (d) T = 29 s

(e) T = 42 s (f) T = 120 s

Figure 5.7: Moving Path Following: robot's trajectory

For the third experiment, the desired path again was the *Cassini oval* determined by the function

$$\varphi(r_x, r_y) = k_s \left[\left(r_x^2 + r_y^2\right)^2 - 2q^2 \left(r_x^2 - r_y^2\right) - p^4 + q^4 \right].$$

We choose here $k_s = 10^{-10}$, $p = 210$Px and $q = 200$Px. The initial position of the Cassini oval is centered at $(x_0, y_0) = (200, 360)$Px and initial orientation is $\pi/2$. The translation velocity of the center of the target frame is $(v_T^x, v_T^y) = (5, 0)$Px/s. The speed of the mobile robot was chosen as $s = 30$Px/s.

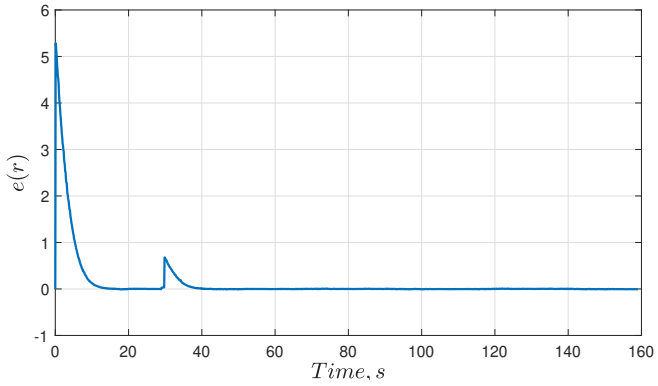


Figure 5.8: Moving Path Following: the dynamics of the tracking error $e = \varphi(r(t))$.

The initial position of the robot (the coordinates x, y are in pixels, and α is in radians) is $(x = 640, y = 364, \alpha = -2.8357)$.

Fig. 5.7 illustrates the resulting robot's trajectory. We have to notice, that around T = 30 s of the experiment the position of the robot was disturbed by manual intervention to verify the convergence rate of the proposed control approach. The corresponding tracking errors are displayed in Fig. 5.8.

5.2 Experiments with a fixed wing UAV

We have used the fixed wing UAV (Fig. 5.9) equipped with the Paparazzi autopilot [74] as the experimental platform. It is about 0.850 kg of mass and 1.2 m of wingspan. The nominal airspeed is $s = 11.5$ m/s. The chosen board for the autopilot includes the usual sensors of three axis gyros, accelerometers, magnetometers and a GPS. Therefore we can measure p, $\dot{p}$ and α. The microcontroller on board is an STMicroelectronics STM32F4. Although there is a logging system on board, the vehicle counts with a serial radio link in order to monitor its status from the ground. The *guiding vector field* algorithm has been programmed as a (guidance) module in Paparazzi and can be combined or integrated easily with other modules in the system. In particular, we have set the periodic frequency of the *guidance vector field module* to 60 Hz. The external frame $\mathcal{O}_T$ was fixed, i.e. $v_T = [0; 0]^\top$, however, we have to use the algorithm for moving path following due to presence of external disturbance (wind) $w \approx 4$m/s. Details about the autopilot configuration are reported in [20].



Figure 5.9: The fixed wing UAV used for the approbation

5.2.1 Sinusoidal path

As a benchmark we chose a sinusoidal path $\mathcal{P}$, namely

$$\varphi(r) = r_x - A\sin(\omega r_y),\ r = R^\top(\alpha_T)(p - p_T), \tag{5.5}$$

where $r = \begin{bmatrix} r_x; & r_y \end{bmatrix}^\top$, $A = 50$ m, $\omega = 0.0251$ rad/m, the position of the fixed external frame $\mathcal{O}_T$ has been chosen as $p_T = \begin{bmatrix} 37.53; & 73 \end{bmatrix}^\top$ m and $\alpha_T = -\pi/2$ rad is the orientation of the frame $\mathcal{O}_T$ with respect to $\mathcal{O}_N$. We choose $k_n = 0.006$, $k_\delta = 0.8$ and define $e(r) = \psi(\varphi(r)) = \varphi(r)$.

Fig. 5.10 illustrates both the resulting UAV's trajectory together and the GFV (2.14) for the given path (5.5). The corresponding tracking error is displayed in Fig. 5.11.

5.2.2 Elliptical path

The second flight experiment have taken place at the aero model club of Eole at Muret, close to the city of Toulouse in France. We were flying on the 18th of August of 2016 between the 14:00 and 18:00 hours local time. The wind according to the weather service of MeteoFrance was about 5m/s blowing from the East with gusts of about 10m/s. The external frame $\mathcal{O}_T$ was fixed, i.e. $v_T = [0; 0]^\top$. Therefore the environmental condition satisfies Assumption 4.1.

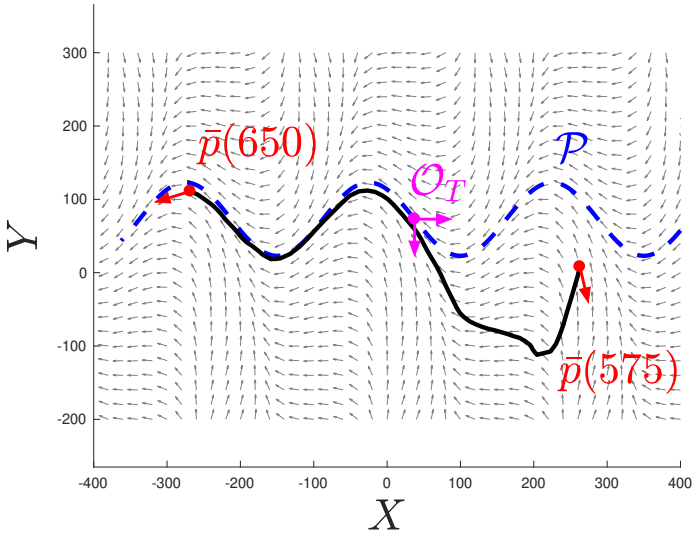


Figure 5.10: The resulting UAV's trajectory and the *guiding vector field*

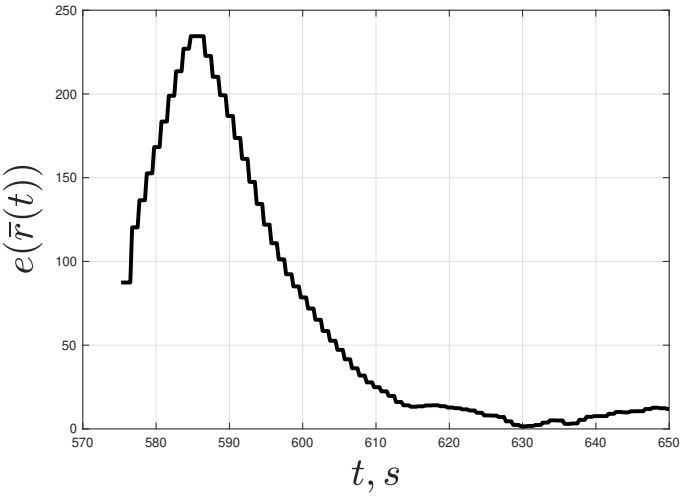


Figure 5.11: The dynamics of the tracking error $e(r(t))$

As a benchmark we consider two ellipses as desired $\mathcal{P}$, namely

$$\varphi(r) = r^\top \begin{bmatrix} 1/a^2 & 0 \\ 0 & 1/b^2 \end{bmatrix} r - 1, \; r = R^\top(\alpha_T)(p - h), \tag{5.6}$$

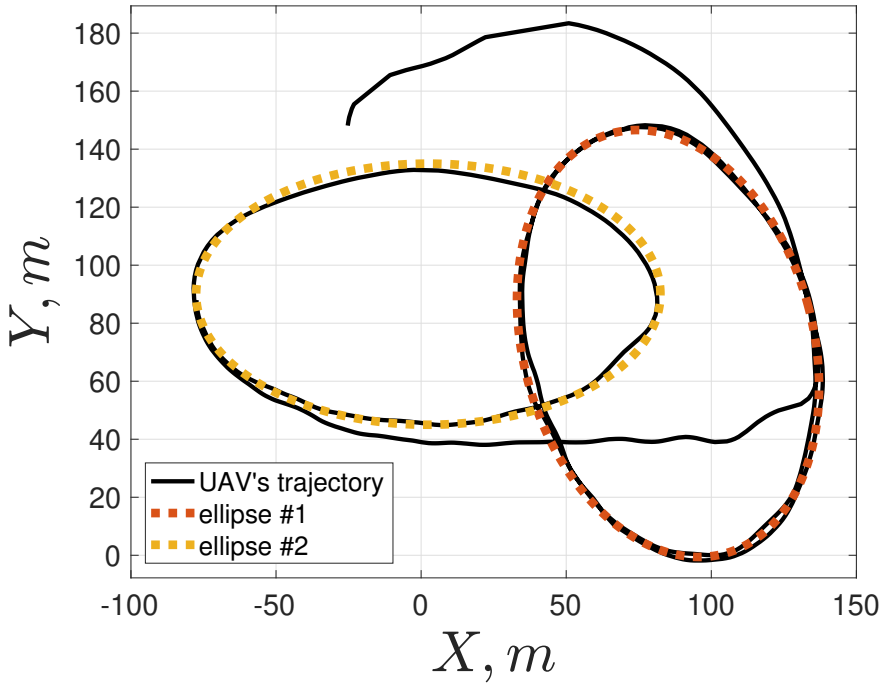


Figure 5.12: The resulting UAV's trajectory

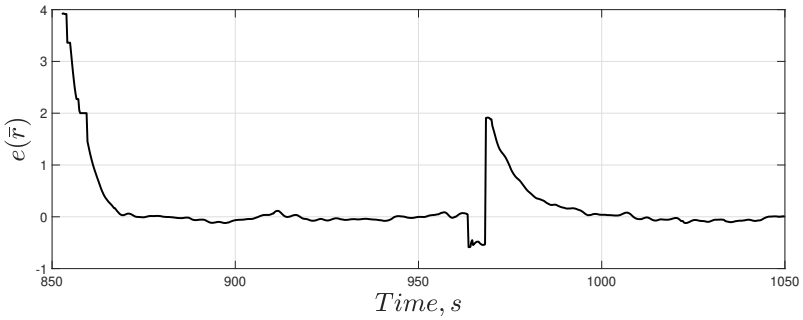


Figure 5.13: The dynamics of the tracking error $e(r(t))$

where $h = \begin{bmatrix} h_x & h_y \end{bmatrix}^T$ is the center of the ellipse with respect to $\mathcal{O}_N$, α_T is the rotation angle of the ellipse with respect to the horizontal axis of $\mathcal{O}_N$ and a and b are the length of the ellipse's axis. Note that only for $p = h$ we have that $\nabla\varphi(r) = 0$.

We choose $k_n = 0.4$, $k_\delta = 1$ and define $e(r) = \psi(\varphi(r)) = \varphi(r)$. The parameters

of the curves have been chosen as follows:

$$\begin{cases} a = 50, & b = 75, & \alpha = 15\ deg, & h = [85; 73], & t1 = [852; 962]; \\ a = 80, & b = 45, & \alpha = 0\ deg, & h = [2.5; 90], & t2 = [968; 1050]. \end{cases}$$

Fig. 5.12 illustrates the resulting UAV's trajectory together with both reference ellipses. The corresponding tracking errors are displayed in Fig. 5.13.

5.3 Experiments with robotic fish

The robotic fish developed by the Intelligent Control Laboratory, Peking University was used (Fig. 5.14). The robot consists of a streamlined head, a flexible body and a caudal fin.



Figure 5.14: The robotic fish

The head is made of rigid plastic, which accommodates an onboard control unit, a radio communication module and a pack of rechargeable batteries. The battery pack is placed at the bottom to lower the center of mass and consequently stabilize the vertical posture (Fig. 5.15). A pair of small pectoral fins is used to diminish the roll movement when the robotic fish swims. Since this pair of fins is rigidly attached to the sides of the head, the controllable robotic fish's locomotion is restricted by horizontal 2D plane. The rubberized flexible body contains three rotational joints that are linked together by aluminum frame. Each joint is driven by a cheap servomotor, which controls its relative joint angle with respect to its neighbors. The caudal fin is attached to the last third joint. Its shape is designed to produce enough propulsion forces to maintain the controllable swimming.

The complete locomotion control architecture is shown in Figure 5.16. The onboard control system gets two commands: target speed setpoint and the rate of turn of the robotic fish. These commands are converted to the parameters of the reference waveform via the corresponding lookup table. These parameters are passed to the lower level that consists of the coupled network of three oscillators, called central pattern generator (CPG), attached to the joints. Simultaneous actions of the joints driven by oscillators change the shape of the flexible fish tale and, thus, produce the required propulsion force.

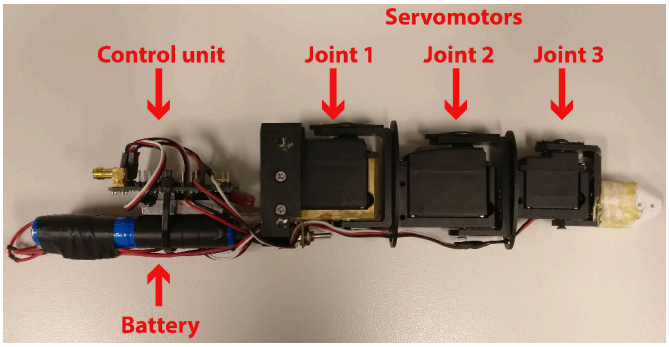


Figure 5.15: The internal structure of the robotic fish

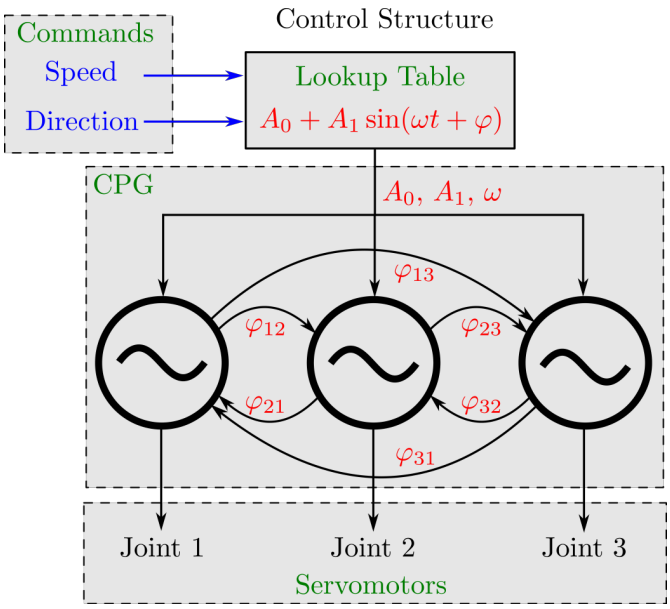


Figure 5.16: The control structure of the robotic fish

The experimental setup consists of the robotic fish, a server computer, an overhead camera and a communication module. The position of the control point and the orientation of the robot are recognized by a image processing algorithm running at a control computer connected to the overhead camera. The PC runs real-time calculation of the control actions based on the pose information of the robot and he desired angular and linear velocities and send them to the robotic fish using radio transmitter.

We verify the performance of the path following control for time-varying GVF.

The algorithm synchronizes the motion of two robotic fish by adjusting radii of the corresponding reference GVF. The basic idea is simple: If two robots circulate maintaining the equal speed around the same central point with different radius, then the one with bigger radius travels more distance and, respectively, has lower phase velocity. Formally it can be written as follows

$$\varphi_1(x_1, y_1) = (x_1 - x_1^0)^2 + (y_1 - y_1^0)^2 - (R + k_{sync}\sin(\phi_1 - \phi_2))^2, \quad (5.7)$$

$$\varphi_2(x_2, y_2) = (x_2 - x_2^0)^2 + (y_2 - y_2^0)^2 - (R + k_{sync}\sin(\phi_2 - \phi_1))^2, \quad (5.8)$$

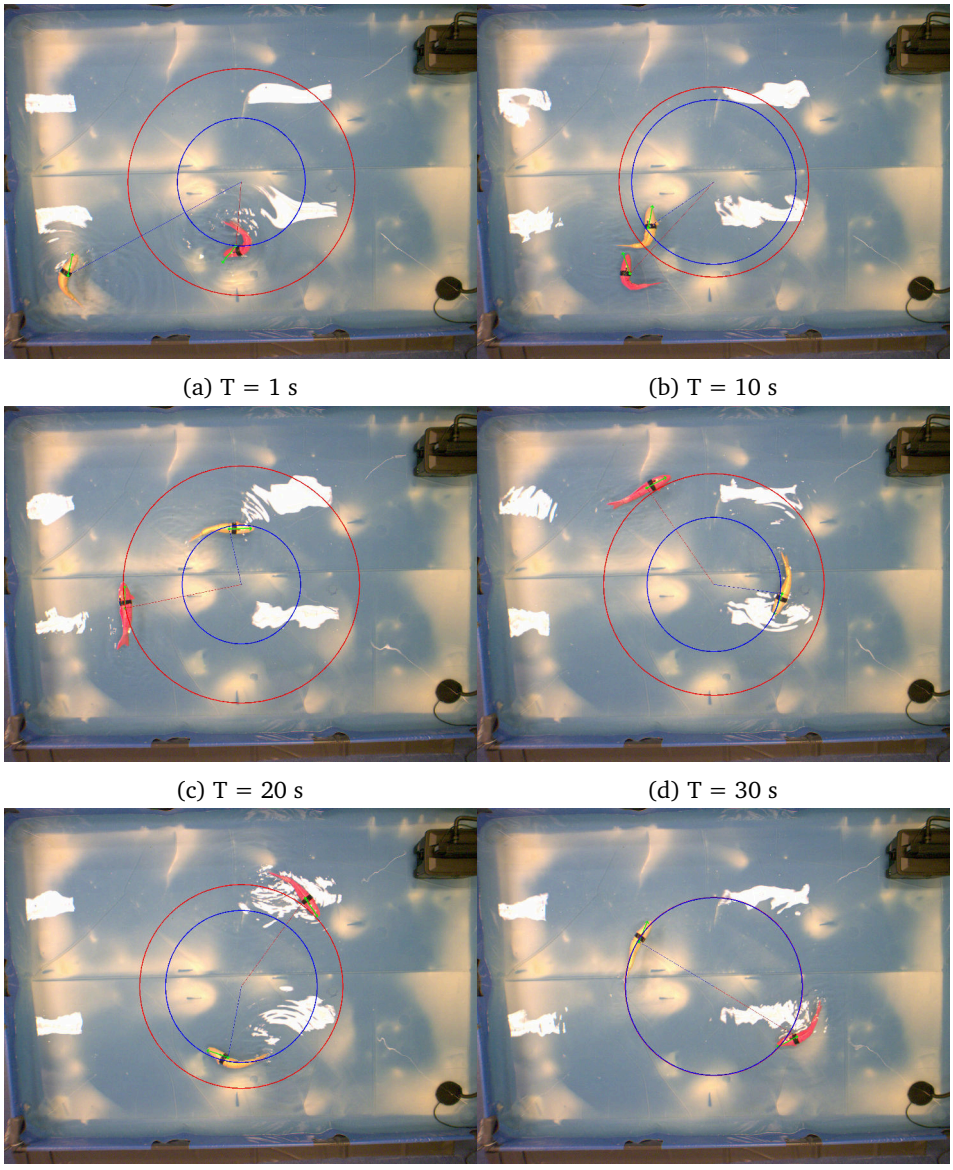
$$\phi_1 = atan2(y_1 - y_1^0, x_1 - x_1^0), \quad (5.9)$$

$$\phi_2 = atan2(y_2 - y_2^0, x_2 - x_2^0), \quad (5.10)$$

where x_1, y_1 and x_2, y_2 stand for coordinates of the robots; $R \in \mathbb{R} > 0$ is the reference radius of the circle; $k_{sync} \in \mathbb{R}$ is the coefficients of coupling between robots; $\phi_1 \in (-\pi, \pi]$ and $\phi_2 \in (-\pi, \pi]$ are corresponding phase angles of the robots; x_1^0, y_1^0 and x_2^0, y_2^0 are centers of the circular orbits for the first and the second robot respectively. Formally, the phases are functions of time, however, they vary slowly compared to the fish' motions, so we neglect this slow dynamics and apply the path following algorithms to the trajectories defined by (5.7) and (5.8) respectively.

The first part of the experiment demonstrate the "counterphase" synchronization or, in other words, the phase difference between robots has to be equal to π radian. The robots follows the circular orbits with the same origin at point $(x_1^0, y_1^0) = (x_2^0, y_2^0) = (800, 600)$Px; the reference radius of the circle $R = 300$Px; $k_{sync} = -100$. Fig. 5.17 shows the snapshots of the experiment.

The second part of the experiment demonstrate the "synphase" synchronization, i.e. the phase angles of robots has to be identical. In order to avoid the collision between robots, the circular orbits have different coordinates of the centers $(x_1^0, y_1^0) = (1000, 600)$Px and $(x_2^0, y_2^0) = (600, 600)$Px; the reference radius of the circle $R = 300$Px; $k_{sync} = 100$. Fig. 5.18 shows the snapshots of the experiment.

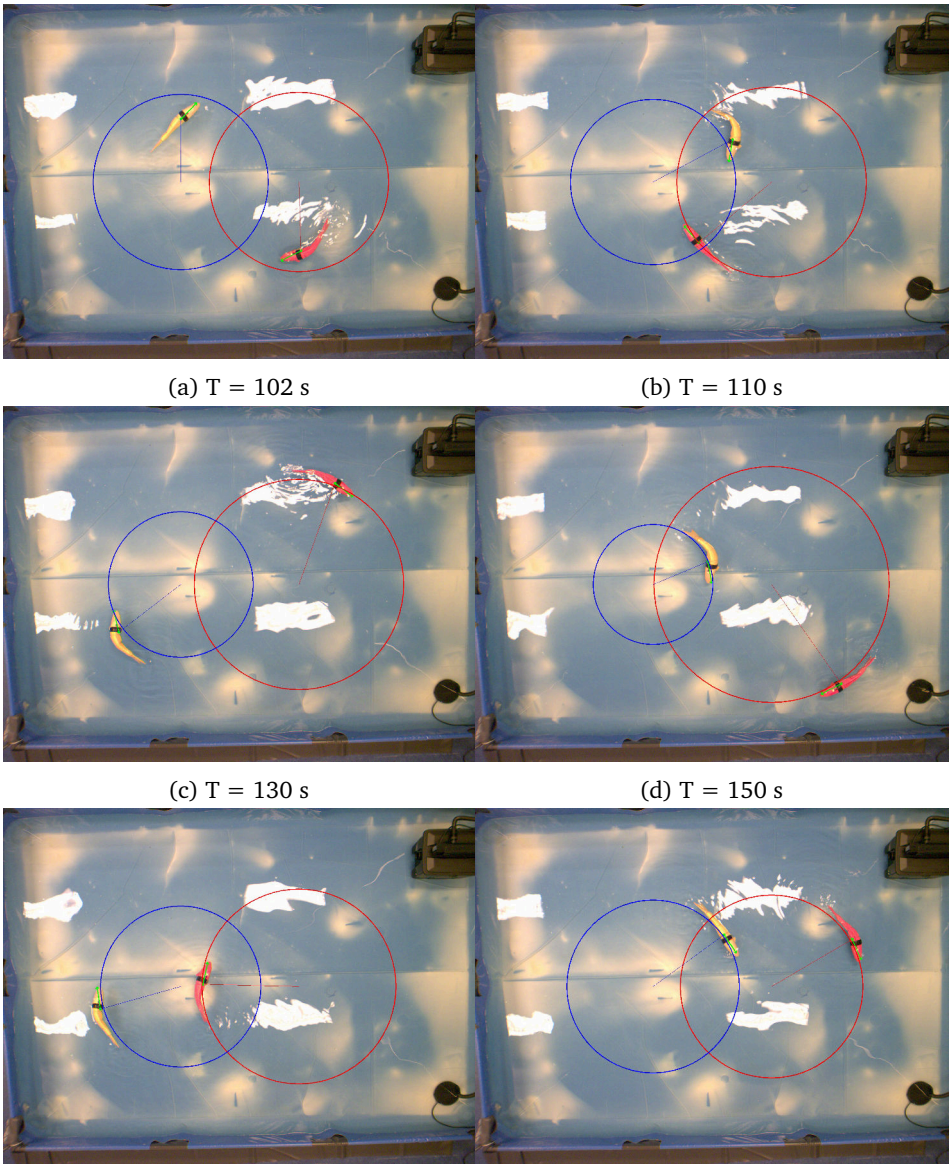


(a) T = 1 s (b) T = 10 s

(c) T = 20 s (d) T = 30 s

(e) T = 40 s (f) T = 53 s

Figure 5.17: Robotic fish: robots' trajectories "counterphase" synchronization



(a) T = 102 s (b) T = 110 s

(c) T = 130 s (d) T = 150 s

(e) T = 170 s (f) T = 190 s

Figure 5.18: Robotic fish: robots' trajectories "synphase" synchronization

Chapter 6
Roll Damping for Marine Vessels

6.1 Introduction

Roll damping is a classical problem in ship motion control [25, 77, 78]. Passive roll damping can be provided by special equipment such as bilge keels, water-tanks and moving weights [60, 77, 78]; however, these devices cannot be easily adapted to the unsteady environment and the changing wave's spectrum. This limitation can be overcome by active (controlled) roll damping, which can be provided by gyroscopic stabilizers, stabilizing fins and/or actuators (rudders and thrusters) used for the vessel's steering. This is illustrated by the *rudder roll stabilization* (RRS), proposed originally for a vessels equipped with a single rudder [15, 18, 57]. Since fins, rudders and thrusters affect both yaw and roll motion of the vessel, the roll damping controller should be integrated with the heading controller (autopilot). These control systems can share some actuators and pursue concurrent goals of roll damping control and course steering.

A vessel's coupled yaw-roll motion can be modeled by a dynamical system, whose inputs are the rudder's and fins' angles and whose outputs stand for the ship's heading and roll. After linearizing this model, classical methods of linear control, e.g. loop shaping and Quantitative Feedback Theory [12, 15, 18, 33, 36] can be applied to stabilize yaw and roll motion. To cope with nonlinearities, methods of feedback linearization and sliding mode control can be used [52, 56]. For vessels equipped with fin stabilizers, classical methods usually decouple the roll motion from the yaw motion [34, 93]; however, as it was observed in [16] ignoring internal cross-couplings often reduces the overall performance.

The roll dynamics of a vessel appear to be non-minimum phase, leading thus to the *fundamental limitation* [15, 31]: a controller stabilizing the vessel's heading cannot fully attenuate the wave-induced roll oscillations. A natural question arises, namely which level of the roll oscillation damping can be provided without deteriorating the yaw control. Mathematically, the latter goal is usually formulated as optimality of a special performance index, which penalizes the time-averaged steering error, roll angle and the control effort. Besides the control input, such a functional implicitly depends on the uncertain wave disturbance that affects the ship's motion. Unlike the aforementioned stabilization techniques, optimization-

based algorithms assume that some model of the disturbance is known. Most typically, the wave-induced motion is approximated by either a "colored noise" or a random polyharmonic signal [25, 78].

The wave model of the first type approximates the wave disturbance by the output of some low-pass shaping filter, fed by a white noise. This approach, prevailing in the literature, reduces roll damping control design to standard methods of optimal controller synthesis, such as the linear-quadratic Gaussian (LQG) control [22, 97], $\mathcal{H}_\infty$ control [11, 19, 87, 90] and model-predictive control (MPC) [77]. As usual in stochastic and minimax control, optimal controllers do not deliver optimal solutions for any specific realization of the stochastic disturbance, providing optimality either "on average" (in the sense of expectation) or in the "worst-case" scenario. Another downside of the mentioned methods is the necessity to estimate the spectral density of the wave motion.

An alternative "discrete" model of the wave motion, often used in marine engineering [58, 68, 77], approximates the wave motion by the sum of sinusoids with known frequencies, where the constant amplitudes are obtained via sampling of the spectral density and random phase shifts are uniformly distributed in $[0, 2\pi]$ in order to get different realizations. For this model of the wave disturbance and linearized vessel's yaw-roll dynamics, the optimal roll damping may be considered as a linear-quadratic optimization problem, where the control system is affected by a partially uncertain polyharmonic signal. A relevant extension of the classical LQR control to cope with such problems has been developed in [54, 55, 80, 81, 82, 102]. It appears that (under natural assumptions) an *optimal universal controller* (OUC) exists, which is independent of the uncertain signal's parameters, delivers the optimal process for *arbitrary* values of these parameters. Furthermore, the OUC can be found in the class of linear stabilizing controllers; a convenient parametrization of such OUCs has been found [102].

In this chapter, we apply Yakubovich's theory of OUC to the problem of optimal roll damping. We illustrate the efficiency of OUCs in the optimal roll damping problem and compare it with classical controllers by using numerical simulations that utilize the "benchmark" vessel's model from [77]. The OUC theory provides a method for combined fin-rudder stabilization control design, avoiding the undesired counteraction between different actuators and improving the resulting efficiency of the control system. Unlike the usual LQR [78], the OUC does need to measure the full state vector and provides optimality for any polyharmonic signal from the specified class; to find OUC, one does not need to solve the Riccati equation. Unlike LQG and H_∞ approaches, the OUC design does not require one to know the spectral density of the wave motion (or, equivalently, the structure of the shaping filter). The OUC depends only on the fixed wave's frequencies and ensures optimality of the cost functional *for any* realization of the random disturbance.

The chapter is organized as follows. In Section 6.2, mathematical models of

the vessel's motion and wave disturbances are considered. In Section 6.3 the theory of OUC in general problems of linear-quadratic optimization with uncertain disturbances is introduced. In Section 6.4, we apply this theory to design an optimal roll damping controller, whose performance is studied numerically in Section 6.6.

6.2 Mathematical models

We first introduce mathematical models of the ship's yaw-roll motion and the wave disturbances.

6.2.1 The vessel's motion

The movements of a marine vessel (as a rigid body) have six degrees of freedom. The standard 6-DoF mathematical model can be found in [25, 78]. However, it is more convenient to use a simplified reduced-order model [25, 77, 97], which is derived under two simplifying assumptions: 1)the effects of the pitch and heave motion of the vessel on its surge, sway, roll and yaw dynamics are negligibile; 2) the vessel's speed is changing slowly relative to the remaining coordinates. Under these assumptions, the yaw and the roll controllers can be designed for a simplified linearized model.

In the original papers on rudder roll damping [18, 57], the simplest configuration of the vessel with one rudder has been considered, whose angle is the single control input of the system. In general, the vessel can be equipped with multiple actuators (rudders, azimuth and tunnel thrusters, waterjets etc.); however, for the sake of autopilot and roll damping control design they are usually replaced by an equivalent "virtual rudder", whose "angle" stands for the scaled rotating yaw moment, distributed among the actuators by a separate *control allocation* system [41]. In addition to this, we allow the vessel to have two synchronized stabilizing fins, whose angle serves as the second control input.

Denoting the rudder, the fin, the roll and the yaw (or heading) angles by, respectively, $\delta_{rud}(t)$, $\delta_{fin}(t)$, $\varphi(t)$ and $\psi(t)$ (Fig. 6.1), the reduced-order vessel's model has the structure illustrated in Fig. 6.2. The system is affected by the environmental disturbance, represented by its roll and yaw components[1] $d_\varphi(t)$, $d_\psi(t)$. The transfer functions from δ_{rud} and δ_{fin} to φ and ψ, denoted by $W_{\varphi r}(s)$,

[1] We consider the "motion superposition" model [77], where the disturbance is modeled as an uncertain displacement from the original trajectory of the vessel. An alternative approach, referred to as the "force superposition" [77], treats the disturbance as an additional force, acting on the ship's hull.

$W_{\varphi f}(s)$, and $W_{\psi r}(s)$, $W_{\psi r}(s)$ respectively, can be written as follows [77, Sect. 8.2]

$$\begin{aligned} W_{\varphi r}(s) &= \frac{K_{\varphi r}(q_1 - s)(q_2 + s)}{(p_1 + s)(p_2 + s)(s^2 + 2\zeta_\varphi \omega_\varphi s + \omega_\varphi^2)}, \\ W_{\psi r}(s) &= \frac{K_{\psi r}(q_3 + s)(s^2 + 2\zeta_q \omega_q s + \omega_q^2)}{s(p_1 + s)(p_2 + s)(s^2 + 2\zeta_\varphi \omega_\varphi s + \omega_\varphi^2)}, \\ W_{\varphi f}(s) &= \frac{K_{\varphi f}(q_4 - s)(q_5 + s)}{(p_1 + s)(p_2 + s)(s^2 + 2\zeta_\varphi \omega_\varphi s + \omega_\varphi^2)}, \\ W_{\psi f}(s) &= \frac{K_{\psi f}(q_6 - s)(s^2 + 2\zeta_t \omega_t s + \omega_t^2)}{s(p_1 + s)(p_2 + s)(s^2 + 2\zeta_\varphi \omega_\varphi s + \omega_\varphi^2)}, \end{aligned} \tag{6.1}$$

where $q_i > 0$, $p_j > 0$, $\omega_\varphi, \omega_q, \omega_t > 0$ and $\zeta_\varphi, \zeta_q, \zeta_t \in (0; 1)$ are constants.

Along with the transfer function, one can introduce the state-space model of the system

$$\begin{aligned} \dot{x}_v(t) &= A_v x_v(t) + B_v \delta(t) \\ y_v(t) &= C_v x_v(t) + G_v d(t). \end{aligned} \tag{6.2}$$

Here the vessel's reduced state vector $x_v(t) = (\varphi, p, \psi, r, v)^\top$ consist of the roll angle φ, the roll rate $p = \dot{\varphi}$, the yaw angle ψ, the yaw rate $r = \dot{\psi}$ and the sway velocity v. The disturbance $d(t) = (d_\varphi, d_\psi)^\top$ stands for the wave-induced motion of the vessel. The vector $y_v(t) = (\varphi, \psi)^\top \in \mathbb{R}^2$ stands for the system's output, whose components φ and ψ are measured, respectively, by a vertical reference unit (VRU) sensor [7] and a gyro or GPS compass and the control input is presented by the vector $\delta(t) = (\delta_{rud}, \delta_{fin})^\top$ We omit the exact formulas for A_v, B_v, C_v, G_v, since they are not explicitly used in the controller design.

6.2.2 The disturbance model

The environmental disturbances, influencing a marine craft's motion, are due to the waves, the wind and the current. The fast oscillations in the roll and the heading angles are mainly caused by the waves, whereas the current and the wind are changing much more slowly and their effect is usually modeled as a constant roll angle and stationary heading deviation. Henceforth, the disturbance $d(t)$ stands for the wave-induced motion only. We use a *polyharmonic* approximation of this motion [25, 78]

$$\begin{aligned} d_\varphi(t) &= \sum_{i=1}^{p} a_i^\varphi \sin\left(\omega_i t + \phi_i^\varphi\right), \\ d_\psi(t) &= \sum_{i=1}^{p} a_i^\psi \sin\left(\omega_i t + \phi_i^\psi\right). \end{aligned} \tag{6.3}$$

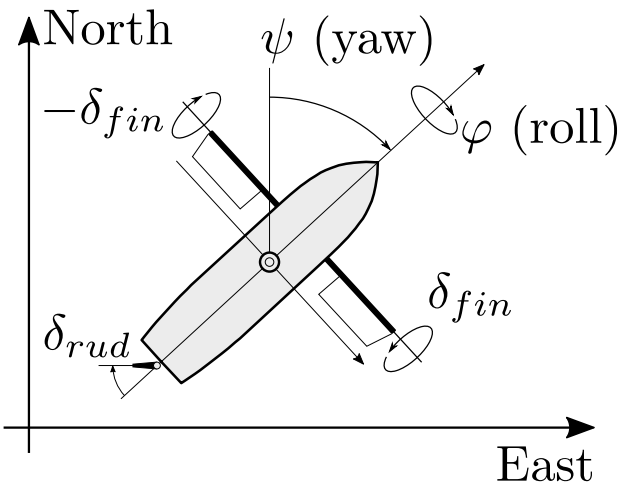


Figure 6.1: The rudder (δ_{rud}), the fin (δ_{fin}), roll (φ) and yaw (ψ) angles.

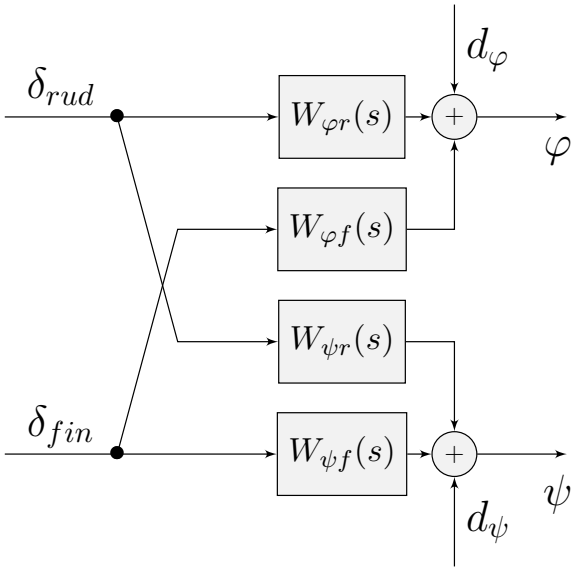


Figure 6.2: The reduced model of vessel's dynamics

Here the spectrum $\omega_1, \ldots, \omega_p \geq 0$ is known. The special case $p = 1$ corresponds to the model of *regular* waves; however, a real state of the sea is best described by a random or *irregular* wave model. This stochastic process can be approximated by the model (6.3) with p being sufficiently large. The constant amplitudes a_i^φ and

a_i^{φ} are obtained via sampling the spectral density with a small enough step $\Delta\omega$ to ensure that the fundamental period of the finite sum of sinusoidal components is longer than the desired duration of the simulation. The random phase shifts ϕ_i^{φ} and ϕ_j^{φ} used to generate different realizations of the stochastic process are uniformly distributed in $[0, 2\pi]$. Although the model (6.3) of irregular waves can describe a sea state quite accurately, the direct use of it in the control design is difficult due to the high dimension. The better strategy is to consider a few "dominating" frequencies corresponding to the peaks of the spectral density. In general, the localization and the shape of the spectral density highly depend on many parameters of motion such as the average speed of the vessel, sailing conditions and a frequency response of the vessel's hull; however, these "dominating" frequencies can be efficiently estimated in real time, see e.g. [10, 13, 24, 37] and references therein. For simplicity and clarity of presentation, we proceed to assume that the number and the values of such frequencies are *known*.

It should be noted that in the existing control literature the wave motion is usually approximated by the "colored noise", that is, the output from a low-pass shaping filter fed by the white noise signal. The simplest approximation for the shaping filter's transfer function (that is, the wave spectrum), is

$$H(s) = \frac{K_w s}{s^2 + 2\zeta_0\omega_0 s + \omega_0^2}. \tag{6.4}$$

Here the constant $K_w > 0$ determines the wave strength, ω_0 is the encounter frequency and ζ_0 is the damping ratio [78]. Unlike our approach, using only the information about the frequencies, the existing approaches, as discussed in Introduction, typically use all parameters of the transfer function $H(s)$, whose identification is a self-standing non-trivial problem.

6.3 Linear-quadratic optimization in presence of uncertain polyharmonic signals

In this section, the basic ideas of the theory of OUC are given for the reader's convenience, following the survey paper [81].

We start with introducing some notation. The set of complex $m \times n$ matrices is denoted by $\mathbb{C}^{m\times n}$. The Hermitian complex-conjugate transpose of a matrix $M \in \mathbb{C}^{m\times n}$ is denoted by $M^* \in \mathbb{C}^{n\times m}$. We use $\imath \triangleq \sqrt{-1}$ to denote the imaginary unit. The real part of a number $z \in \mathbb{C}$ is denoted by $Re\, z$.

6.3.1 A family of uncertain optimization problems

Consider a linear time-invariant MIMO system, influenced by an exogenous signal

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + Ed(t), \\ y(t) &= Cx(t) + Du(t) + Gd(t). \end{aligned} \tag{6.5}$$

Here $x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m, y(t) \in \mathbb{R}^k$ stand for, respectively, the state vector, the control and the observed output. The signal $d(t) \in \mathbb{R}^l$ is a polyharmonic process with *known* spectrum $\omega_1, \dots, \omega_N$

$$d(t) = Re \sum_{j=1}^{N} d_j e^{\imath \omega_j t}, \tag{6.6}$$

whose complex amplitudes $d_i \in \mathbb{C}^l$ (absorbing also the phase shifts) are uncertain. The components of this exogenous signal may include disturbances, measurement noises and reference signals.

In presence of the oscillatory disturbance (6.6), the solutions of (6.5) do not vanish at infinity. The goal of control is to guarantee boundedness of the solution $(x(t), u(t))$ and its optimality in the sense of the following quadratic performance index

$$J[x, u, d] = \varlimsup_{T \to \infty} \frac{1}{T} \int_0^T \mathcal{F}[x(t), u(t), d(t)]\, dt. \tag{6.7}$$

Here $\mathcal{F}$ is a quadratic form, which is assumed to be non-negative definite $\mathcal{F} \geq 0$. Considering the integrand in (6.7) as a measure of the solution's "energy", its average value J can be thought of as the solution's average "power". Formally, the control goal can be formulated as follows

$$\begin{gathered} \text{minimize } J(x(\cdot), u(\cdot), d(\cdot)) \\ \text{subject to (6.5) and } \sup_{t \geqslant 0}(|x(t)| + |u(t)|) < \infty. \end{gathered} \tag{6.8}$$

In fact, (6.8) defines an infinite family of optimization problems, corresponding to different choices of the amplitudes $d_1, \dots, d_N$. Obviously, the set of optimal processes also depends on the amplitudes and hence cannot be found explicitly. Nevertheless, it can be shown that an *optimal universal* controller (OUC) exists that provides an optimal process *for any* uncertain amplitudes d_i, solving thus the whole family of optimization problems (6.8).

Definition 6.1. A causal operator $\mathcal{U} : y(\cdot) \mapsto u(\cdot)$ is an OUC for the family of optimization problems (6.8), if for any initial condition $x(0) \in \mathbb{R}^n$ and any amplitudes

$d_1, \dots, d_N$ in (6.6) there exists a unique solution of the closed-loop system

$$\dot{x} = Ax + Bu + Ed,\ y = Cx + Du + Gd, \quad u(\cdot) = \mathcal{U}y(\cdot),$$

which is bounded and delivers an optimum to (6.8).

6.3.2 A class of linear OUC

Although the existence of OUCs may seem exceptional, such controllers exist under rather mild assumptions on the system and the cost functional.

We assume that the system (6.5) is stable, that is, $\det(sI_n - A) \neq 0$ whenever $Re\, s \geq 0$. If the system is stabilizable and detectable, one may always augment it with an observer-based stabilizing controller, so the stability assumption can be adopted without loss of generality.

Let $F = F^\top$ stand for the matrix of the quadratic form $\mathcal{F}(x, u, d)$ and $F_0 = F_0^\top$ be the matrix of the quadratic form $\mathcal{F}_0(x, u) = \mathcal{F}(x, u, 0)$, that is,

$$\begin{aligned}\mathcal{F}(x,u,d) = \begin{bmatrix} x \\ u \\ d \end{bmatrix}^\top F \begin{bmatrix} x \\ u \\ d \end{bmatrix} = \begin{bmatrix} x \\ u \end{bmatrix}^\top F_0 \begin{bmatrix} x \\ u \end{bmatrix} + \\ + 2d^\top F_{dx} x + 2d^\top F_{du} u + d^\top F_{dd} d,\end{aligned} \tag{6.9}$$

where $F_{dx}, F_{du}, F_{dd} = F_{dd}^\top$ are matrices of appropriate dimensions. We introduce the rational complex-valued matrix $\Pi(\imath\omega) = \Pi(\imath\omega)^*$ as follows

$$\tilde{u}^* \Pi(\imath\omega)\tilde{u} = \begin{bmatrix} A_{\imath\omega}^{-1} B\tilde{u} \\ \tilde{u} \end{bmatrix}^* F_0 \begin{bmatrix} A_{\imath\omega}^{-1} B\tilde{u} \\ \tilde{u} \end{bmatrix},\ A_s := sI_n - A,$$

and assume that the *frequency-domain* condition holds

$$\Pi(\imath\omega) \geq \varepsilon I_m, \quad \varepsilon = const > 0. \tag{6.10}$$

The condition (6.10) is a standard solvability condition for classical LQR problems, providing the existence of the stabilizing solution to the Riccati equation [4]. It always holds when $F_0(x, u)$ is positively definite, which is a natural assumption in practice. The condition(6.10) cannot be discarded and, moreover, its "strong" violation in the sense that $\tilde{u}^* \Pi(\imath\omega_0)\tilde{u} < 0$ for some $\omega_0 \in \mathbb{R}$ and $\tilde{u} \in \mathbb{C}^m$ implies[2] the ill-posedness of the problem (6.8): $\inf J = -\infty$ for any signal (6.6).

Under non-restrictive assumptions, the OUC exists and can be found among

[2]For a similar discrete-time optimization problem, the proof is available in [55], and the continuous-time case is considered in the same way.

linear controllers

$$N\left(\frac{d}{dt}\right)u(t) = M\left(\frac{d}{dt}\right)y(t), \tag{6.11}$$

where N and M stand for matrix polynomials; the matrix $N(s)$ is square and $\det N \not\equiv 0$. The relevant result is given by the following theorem.

Theorem 6.2. *[81] Let the system* (6.5) *be stable and the inequality* (6.10) *hold. Then the linear controller* (6.11) *is an OUC for the family of problems* (6.8) *if the following two conditions hold*

1. *the closed-loop systems is stable, that is,*

$$\det\begin{bmatrix} sI_n - A & -B \\ -M(s)C & N(s) - M(s)D \end{bmatrix} \neq 0, \tag{6.12}$$

$$\forall s : Re\, s \geqslant 0;$$

2. *the* ***closed-loop*** *transfer function W_{ud} from d to u satisfies the interpolation equations*

$$W_{ud}(\imath\omega_j) = R_j,\ \forall j = 1, 2, ..., N, \tag{6.13}$$

where the constant matrices R_j are as follows

$$R_j = -\Pi^{-1}(\imath\omega_j)\begin{bmatrix} A_{\imath\omega_j}^{-1}B \\ I_m \\ 0 \end{bmatrix}^* F \begin{bmatrix} A_{\imath\omega_j}^{-1}E \\ 0 \\ I_l \end{bmatrix}.$$

Note that, unlike the classical LQR problem, where the optimal controller is uniquely defined from the Riccati equation, the OUC in the problem (6.8) is not unique; to find it, one need not solve Riccati equations. We will use Theorem 6.2 in a special situation, where $\mathcal{F}$ depends only on the output and the control, i.e. F admits the decomposition

$$F = \begin{bmatrix} C & D & G \\ 0 & I_m & 0 \end{bmatrix}^* \hat{F} \begin{bmatrix} C & D & G \\ 0 & I_m & 0 \end{bmatrix}, \tag{6.14}$$

where $\hat{F} = \hat{F}^* \in \mathbb{C}^{m+n}$. In this situation, one has

$$\begin{aligned} \Pi(\imath\omega) &= \begin{bmatrix} W_{yu}^0(\imath\omega) \\ I_m \end{bmatrix}^* \hat{F} \begin{bmatrix} W_{yu}^0(\imath\omega) \\ I_m \end{bmatrix}, \\ R_j &= -\Pi^{-1}(\imath\omega_j)\begin{bmatrix} W_{yu}^0(\imath\omega_j) \\ I_m \end{bmatrix}^* \hat{F} \begin{bmatrix} W_{yd}^0(\imath\omega_j) \\ 0 \end{bmatrix}. \end{aligned} \tag{6.15}$$

Here $W^0_{yu}(s)$ and $W^0_{yd}(s)$ stand for the **open-loop** transfer functions from respectively u and d to y

$$W^0_{yu}(s) := CA_s^{-1}B + D,\ W^0_{yd}(s) := CA_s^{-1}E + G.$$

Recalling that A is a Hurwitz matrix, it can be shown that the closed-loop system is stabilized by the controller (6.11), whose coefficients are as follows

$$\begin{aligned} M(s) &= \Delta(s)r(s), \\ N(s) &= M(s)\left[CA_s^{-1}B + D\right] + \rho(s)I_m, \\ \Delta(s) &:= \det(A_s) = \det(sI_n - A). \end{aligned} \tag{6.16}$$

Here $r(s)$ is a matrix polynomial and $\rho(s)$ is a *scalar* Hurwitz polynomial with $\deg\rho \geq \deg M$. Such a controller is "feasible" in the sense that its transfer matrix $N^{-1}M$, as well as the closed-loop system's transfer matrices from d to x, u, are proper. For the controller (6.11),(6.16), one obtains

$$W_{ud}(s) = \frac{M(s)}{\rho(s)}W^0_{yd}(s), \tag{6.17}$$

and the interpolation constraints (6.13) boil down to

$$\Delta(\imath\omega_j)r(\imath\omega_j)W^0_{yd}(\imath\omega_j) = \rho(\imath\omega_j)R_j. \tag{6.18}$$

The constraints (6.18) can be satisfied when

$$\det\left[W^0_{yd}(\imath\omega_j)W^0_{yd}(\imath\omega_j)^*\right] \neq 0\ \forall j = 1, \ldots, N. \tag{6.19}$$

Here W^0_{yd} is the open-loop transfer matrix from d to y. The conditions (6.19) typically hold when $\dim y \geq \dim d$. Furthermore, if (6.19) holds, the coefficients of r and ρ can be chosen as continuous functions of ω_j, so that the controller is robust to small deviations in the spectrum $\omega_j^{'} \approx \omega_j$. Choosing an arbitrary Hurwitz polynomial ρ of degree $\deg\rho \geqslant 2N + \deg\delta - 1$, one needs to find the matrix polynomial r with $\deg r \leq 2N - 1$, satisfying the conditions

$$\begin{aligned} r(\imath\omega_j) &= r^0(\imath\omega_j), \\ r^0(s) &:= \frac{\rho(s)R_j}{\Delta(s)}W^0_{yd}(s)^*\left[W^0_{yd}(s)W^0_{yd}(s)^*\right]^{-1}. \end{aligned} \tag{6.20}$$

Separating the real and imaginary parts, one obtains $2N$ equations for $2N$ real coefficients of r.

It appears that any OUC (6.11) is equivalent, in some sense [81, 102], to the controller (6.16) with some polynomials r, ρ, satisfying the interpolation con-

straints (6.18).

Remark 6.3. Note that the controller (6.16) in fact does not depend on the state-space model (6.5), involving only the system's characteristic polynomial $\Delta(s)$ and the open-loop transfer function $W^0_{yu}(s) := D + C(sI - A)^{-1}B$ from u to y (Fig. 6.3). In the case where $\mathcal{F} = \mathcal{F}(y, u)$ depends only on y and u, the interpolation conditions (6.18) also involve only the values of $W^0_{yu}(\imath\omega_j)$ and $W^0_{yd}(\imath\omega_j)$ rather than the whole state model (6.5). Hence, in this special situation, the design of OUC requires only the knowledge of $\Delta(s)$, $W^0_{yu}(s)$ and $W^0_{yd}(s)$, which are independent of the minimal state-space realization.

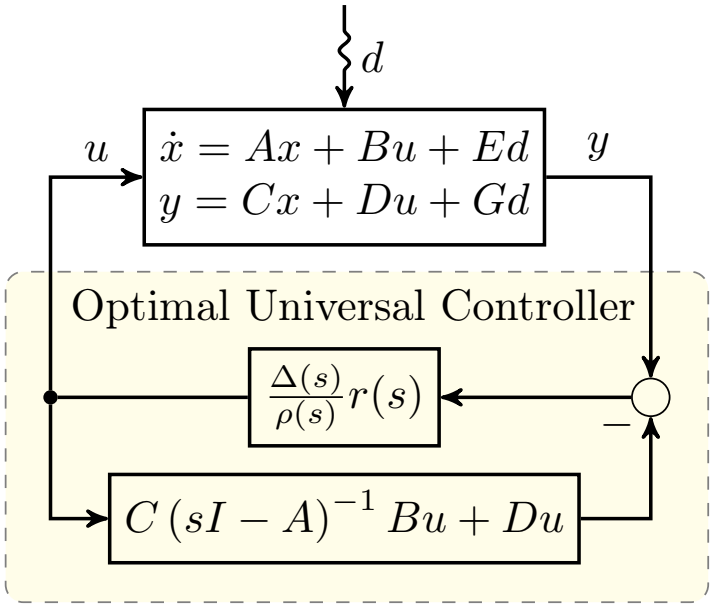


Figure 6.3: The structure of the OUC (6.16)

Remark 6.4. As discussed in [55], the important property of the OUC (6.11) is its *robustness* against small changes in the frequencies ω_j, whereas the straightforward LQR-based design leads to a controller that is formally optimal yet non-robust to deviations in spectrum. The results from [55] deal with discrete-time systems, but this robustness property is retained by the continuous-time OUC (6.11).

Remark 6.5. In general, one has a lot of freedom in choosing the coefficients of $\rho(s)$, and their "optimal" choice remains an important research topic as a subject of ongoing research. In practice, the polyharmonic model of the disturbance is usually imprecise: the signal $d(t)$ contains frequencies other than ω_j (whose amplitudes are sufficiently small). Fig. 6.3 suggests that, ideally, the OUC's transfer

function $r(s)\Delta(s)/\rho(s)$ should have a sufficiently narrow bandpass, containing the frequencies ω_j in order to damp these unmodeled spectrum. Although this requirement is not very formal, it can be used for practical tuning of the OUC's parameters.

6.4 Optimal Universal Roll Damping Controllers

In this section, we reduce the optimal roll damping problem to a special case of the problem (6.8). The cost functional will depend only on the control effort and output. In view of Remark 6.3, in this situation one does not need to know a special state-space representation of the open-loop system, requiring only its characteristic polynomial and transfer matrices W^0_{yu}, W^0_{yd}. In this sense, an optimal controller can be designed in the frequency domain.

We assume that the vessel's heading is stabilized by a known *autopilot* (Fig. 6.4). Behind this statement, there are two practical considerations. First of all, it allows splitting of the adjustment procedure for a motion control system on the vessel in two sequential stages: the independent tuning of an autopilot and the following design of the roll damping controller. The second reason is the flexibility and the modularity; the roll damping system may be supplied by a manufacturer of the equipment such as high-performance rudders or active fins independent of the development of the autopilot, which is in itself a challenging task. The autopilot design problem has been thoroughly studied in the literature [25, 68, 77, 99] and is beyond the scope of this thesis. Furthermore, we assume that the roll damping system is aware of the measured heading of the vessel and the constant heading setpoint $\bar{\psi}$. In practice, $\bar{\psi}(t)$ can be a function of time, e.g. when autopilot steers the vessel along a curvilinear path. However, these dynamics are much slower than the ship's roll motion, and hence are neglected in the roll damping system design. The deviation among them (heading error) $e_\psi(t)$, along with the roll damping error $e_\varphi(t)$ are the inputs to the roll damping system (Fig. 6.4). Mathematically,

$$e_\psi(t) := \psi(t) + d_\psi(t) - \bar{\psi},$$

$$e_\varphi(t) := \varphi(t) + d_\varphi(t).$$

The rudder angle $\delta_{rud}(t)$ is the sum of the autopilot's and the roll damping controller's commands (Fig. 6.4), denoted respectively by $\delta_{AP}(t)$ and $u_1(t)$. The fin angle $\delta_{fin}(t)$ is used as the second control input $u_2(t)$. Denoting the autopilot's

transfer function by $W_{AP}(s)$, one has

$$\delta_{rud}(t) = \delta_{AP}(t) + u_1(t) = W_{AP}\left(\frac{d}{dt}\right) e_\psi(t) + u_1(t)$$
$$\delta_{fin}(t) = u_2(t).$$

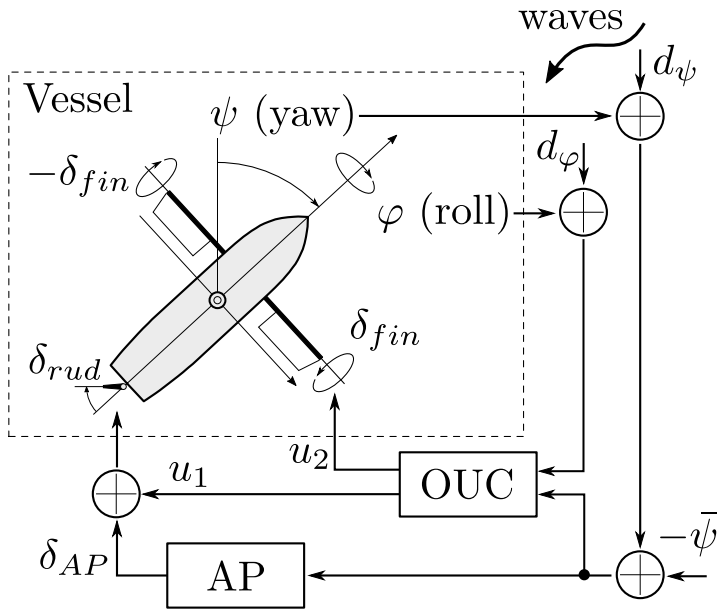


Figure 6.4: The vessel's control system: the autopilot (AP) and the optimal universal controller (OUC) as a roll damping system.

The yaw-roll dynamics of the vessel, closed by the autopilot, are represented by the input-output model

$$y(t) = W^0_{yu}\left(\frac{d}{dt}\right) u(t) + W^0_{yd}\left(\frac{d}{dt}\right) d(t),$$
$$y(t) := \begin{bmatrix} e_\varphi(t) \\ e_\psi(t) \end{bmatrix}, \; u(t) := \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}, \; d(t) := \begin{bmatrix} \bar{\psi} \\ d_\varphi(t) \\ d_\psi(t) \end{bmatrix}. \tag{6.21}$$

Here $d_\varphi(t), d_\psi(t)$ are the polyharmonic components of the wave-induced motion (6.3). Considering $\bar{\psi}$ as a harmonic signal of zero frequency, $d(t)$ is a special case of (6.6) with $l = 3$ and $N = 1 + p$, where ω_j, $k = 1, \ldots, p$ are the wave frequencies from (6.6) and $\omega_{1+p} = 0$. The transfer functions W^0_{yu}, W^0_{yd} depend on the

autopilot's transfer function W_{AP} (from e_ψ to δ_{AP}) and the functions W_{yaw}, W_{roll} from (6.1). The exact formulas for W^0_{yu}, W^0_{yd} are derived in Section 6.5 and it can be easily seen from these formulas that (6.19) always holds for any wave spectrum $\omega_1, \dots, \omega_N \in \mathbb{R}$.

The cost functional penalizes the mean square values of the following three variables

i. the roll displacement (e_ϕ),

ii. the heading deviation (e_ψ), and

iii. the control effort.

Denoting the corresponding penalty weights by $\alpha, \beta, \gamma_{1,2} > 0$, we introduce the quadratic cost functional as follows

$$\begin{aligned} J &= \varlimsup_{T\to\infty} \frac{1}{T} \int_0^T \mathcal{F}(y(t), u(t))\, dt, \\ \mathcal{F}(y,u) &:= \alpha e_\varphi^2 + \beta e_\psi^2 + \gamma_1 u_1^2 + \gamma_2 u_2^2. \end{aligned} \tag{6.22}$$

The Hermitian form $\mathcal{F}$ can be represented in the form (6.14), where $\hat{F}$ is defined by

$$\hat{F} = \begin{pmatrix} \alpha & 0 & 0 & 0 \\ 0 & \beta & 0 & 0 \\ 0 & 0 & \gamma_1 & 0 \\ 0 & 0 & 0 & \gamma_2 \end{pmatrix}.$$

The matrix function $\Pi(\imath\omega)$ and the matrices R_j are defined by (6.15); $\Pi(\imath\omega) > 0$ since $\gamma_1, \gamma_2 > 0$.

Problem 3. *To design the OUC (Definition 6.1) $(\varphi(\cdot), \psi(\cdot)) \mapsto (u_1, u_2)$ which provides optimality of the functional* (6.22) *for any wave disturbance* (6.3) *(frequencies ω_i are known, amplitudes and phases are unknown).*

This formalization of the RRS problem makes it possible to apply the theory of optimal universal controllers, discussed in the previous section. To design OUC (6.11) with the coefficients (6.16), one has to choose $\rho(s)$ to be a scalar Hurwitz polynomial with $\deg\rho \geq \deg r + \deg\Delta$, whereas r is a 2×2 matrix polynomial that satisfies (6.18). By fixing $\rho(\imath\omega_j)$ and splitting the real and imaginary parts in the interpolation condition (6.18), one obtains a pair of real-valued matrix equations for the coefficients of $r(s)$. The only exception is $j = N = p+1$: since $\omega_N = 0$, the equation (6.18) is real-valued. Hence we get $1+2p$ equations for the coefficients of the polynomial r. To satisfy them, the polynomial $r(s)$ should have $1+2p$ real-valued coefficients, i.e. it suffices to choose $\deg r = 2p$ and $\deg\rho \geq \deg\Delta + 2p$.

The just described algorithm to design an OUC for the roll damping problem can be summarized as follows.

Algorithm 3: Design of an OUC

1. choose a Hurwitz polynomial $\rho(s)$ with $\deg \rho(s) \geq 2p + \deg \Delta$;
2. compute the matrices R_j from (6.15) (here $N = 1 + p$, $\omega_1, \ldots, \omega_p$ are the wave frequencies from (6.3) and $\omega_N = \omega_{1+p} = 0$);
3. compute $W^0_{yd}(\imath\omega_j)$ (see 6.5);
4. find the real coefficients of the matrix polynomial $r(s) = r_0 + \ldots + r_{2p}s^{2p}$ from (6.20);
5. the controller (6.11) with the coefficients (6.16) provides optimality of (6.22) for any uncertain amplitudes and phases.

For the detailed derivation of the OUC controller one may represent the transfer functions (6.1) as follows

$$\begin{aligned} W_{\varphi r}(s) &= \frac{sb_{\varphi r}(s)}{a(s)},\ W_{\psi r}(s) = \frac{b_{\psi r}(s)}{a(s)}, \\ W_{\varphi f}(s) &= \frac{sb_{\varphi f}(s)}{a(s)},\ W_{\psi f}(s) = \frac{b_{\psi f}(s)}{a(s)}, \end{aligned} \tag{6.23}$$

In order to stabilize the vessel's heading, the autopilot controller is chosen to be

$$W_{ap}(s) = \frac{b_{ap}(s)}{a_{ap}(s)}. \tag{6.24}$$

A straightforward computation of $W^0_{yu}(s)$, $W^0_{yd}(s)$ (see 6.5) shows that

$$\begin{aligned} W^0_{yu}(s) &= \frac{1}{\Delta(s)} \begin{bmatrix} sa_{ap}(s)b_{\varphi r}(s) & b^0_{\varphi u_2}(s) \\ a_{ap}(s)b_{\psi r}(s) & a_{ap}(s)b_{\psi f}(s) \end{bmatrix}, \\ W^0_{yd}(s) &= \begin{bmatrix} -\frac{sb_{\varphi r}(s)b_{ap}(s)}{\Delta(s)} & \frac{1}{\Delta(s)} & \frac{sb_{\varphi r}(s)b_{ap}(s)}{\Delta(s)} \\ -\frac{a(s)a_{ap}(s)}{\Delta(s)} & 0 & \frac{a(s)a_{ap}(s)}{\Delta(s)} \end{bmatrix}, \\ \Delta(s) &= a(s)a_{ap}(s) - b_{\psi r}(s)b_{ap}(s), \\ b^0_{\varphi u_2}(s) &= a_{ap}(s)b_{\varphi f}(s) + b_{ap}(s)\frac{b_{\varphi r}(s)b_{\psi f}(s) - b_{\varphi f}(s)b_{\psi r}(s)}{a(s)}. \end{aligned}$$

Obviously, $\deg \Delta(s) = \deg a(s) + \deg a_{ap}(s)$.

The application of this procedure to a specific vessel's model is illustrated in the next sections.

6.5 Transfer matrices of the ship-autopilot system

In this section we are going to present the transformation procedure on how to obtain the models in equations (6.21) based on the general dynamics of the vessel described by the transfer function

$$\begin{aligned}\varphi(t) =& W_{\varphi r}\left(\frac{d}{dt}\right)\delta_{rud}(t) + W_{\varphi f}\left(\frac{d}{dt}\right)\delta_{fin}(t),\\ \psi(t) =& W_{\psi r}\left(\frac{d}{dt}\right)\delta_{rud}(t) + W_{\psi f}\left(\frac{d}{dt}\right)\delta_{fin}(t).\end{aligned}$$

The observed outputs of the system are

$$e_\psi(t) = \psi(t) + d_\psi(t) - \bar{\psi}, \quad e_\varphi = \varphi(t) + d_\varphi(t),$$

where $\bar{\psi}$ is the heading setpoint. We introduce the two control inputs as follows

$$\begin{aligned}u_1 &= \delta_{rud}(t) - W_{AP}\left(\frac{d}{dt}\right)e_\psi(t),\\ u_2 &= \delta_{fin}(t),\end{aligned}$$

where W_{AP} is the autopilot's transfer function, stabilizing the vessel's yaw motion. Putting the equations together, one arrives at the following

$$\begin{bmatrix}1 & -W_{\varphi r}W_{ap}\\ 0 & 1 - W_{\psi r}W_{ap}\end{bmatrix}\begin{bmatrix}e_\varphi\\ e_\psi\end{bmatrix} = \begin{bmatrix}W_{\varphi r} & W_{\varphi f}\\ W_{\psi r} & W_{\psi f}\end{bmatrix}\begin{bmatrix}u_1\\ u_2\end{bmatrix} + \begin{bmatrix}0 & 1 & 0\\ -1 & 0 & 1\end{bmatrix}\begin{bmatrix}\bar{\psi}\\ d_\varphi\\ d_\psi\end{bmatrix}.$$

Assuming that the autopilot stabilizes the yaw loop i.e. $1 - W_{\psi r}W_{ap} \neq 0$ this yields

$$\begin{bmatrix}e_\varphi\\ e_\psi\end{bmatrix} = \begin{bmatrix}W^0_{\varphi u_1} & W^0_{\varphi u_2}\\ W^0_{\psi u_1} & W^0_{\psi u_2}\end{bmatrix}\begin{bmatrix}u_1\\ u_2\end{bmatrix} + \begin{bmatrix}W^0_{\varphi\bar{\psi}} & 1 & W^0_{\varphi d_\psi}\\ W^0_{\psi\bar{\psi}} & 0 & W^0_{\psi d_\psi}\end{bmatrix}\begin{bmatrix}\bar{\psi}\\ d_\varphi\\ d_\psi\end{bmatrix},$$

where

$$\begin{aligned}W^0_{\varphi u_1} &= (1 - W_{\psi r}W_{ap})^{-1}W_{\varphi r},\\ W^0_{\psi u_1} &= (1 - W_{\psi r}W_{ap})^{-1}W_{\psi r},\\ W^0_{\varphi u_2} &= W_{\varphi f} + (1 - W_{\psi r}W_{ap})^{-1}W_{\varphi r}W_{ap}W_{\psi f},\\ W^0_{\psi u_2} &= (1 - W_{\psi r}W_{ap})^{-1}W_{\psi f},\\ -W^0_{\varphi\bar{\psi}} &= W^0_{\varphi d_\psi} = (1 - W_{\psi r}W_{ap})^{-1}W_{\varphi r}W_{ap},\\ -W^0_{\psi\bar{\psi}} &= W^0_{\psi d_\psi} = (1 - W_{\psi r}W_{ap})^{-1}.\end{aligned}$$

Recalling that

$$y = \begin{bmatrix} e_\varphi \\ e_\psi \end{bmatrix}, \quad u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \quad d = \begin{bmatrix} \bar{\psi} \\ d_\varphi \\ d_\psi \end{bmatrix},$$

the transfer matrices from u and d respectively to y are given by

$$W^0_{yu} = \begin{bmatrix} W^0_{\varphi u_1} & W^0_{\varphi u_2} \\ W^0_{\psi u_1} & W^0_{\psi u_2} \end{bmatrix}, W^0_{yd} = \begin{bmatrix} W^0_{\varphi \bar{\psi}} & 1 & W^0_{\varphi d_\psi} \\ W^0_{\psi \bar{\psi}} & 0 & W^0_{\psi d_\psi} \end{bmatrix}.$$

6.6 Numerical simulation

In this section we consider a numerical example to illustrate the proposed approach. The exact expressions for the transfer functions (6.23) of a vessel from [77, Appendix B] obtained via linearization of the nonlinear 4-DoF model at the constant speed 8 m/s. This vessels has a single rudder and two synchronous fins.

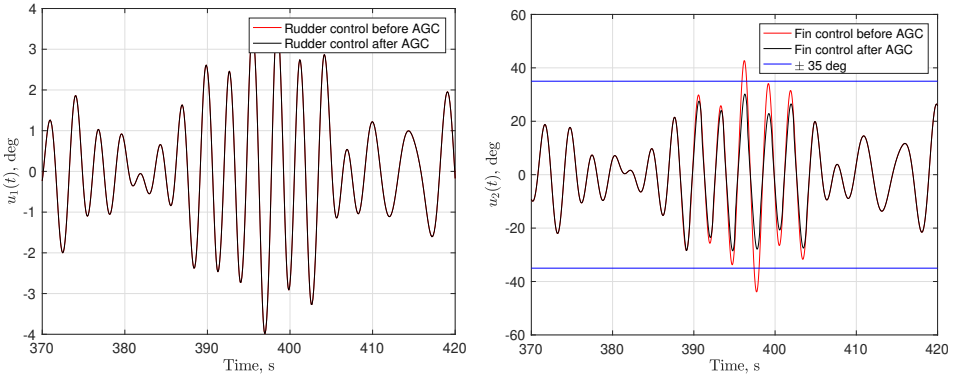
$$\begin{gathered} a(s) = s(s + 0.4375)(s + 0.04404)(s^2 + 0.2164s + 1.31), \\ b_{\varphi r}(s) = -0.159(s - 0.4919)(s + 0.3005), \\ b_{\psi r}(s) = -0.078(s + 0.1785)(s^2 + 0.2586s + 1.324), \\ b_{\varphi f}(s) = 0.402(s + 0.4501)(s + 0.03056), \\ b_{\psi f}(s) = -0.006(s - 0.9642)(s^2 + 0.1974s + 0.2361), \end{gathered}$$

For this simulation we assume that the stabilizing autopilot (6.24) has the following form

$$a_{ap}(s) = (s + 10), \quad b_{ap}(s) = 57(s + 0.5263).$$

Whereas the controller design is based on this linearized model, our simulations take into account the nonlinear dynamics of the rudder's and fins' steering machines [77, 5.6] with the maximal angles 40 degrees (rudder) and 35 degrees (fins) and the maximal rates 20 degree/s and 25 degree/s. [77, Appendix B.4].

To cope with saturations, we use the heuristical approach, called the automatic gain control (AGC) [22, 52] that decreases the actuator command in a smart way to ensure that saturation (of the actuator angle or rate) never occurs. In Fig. 6.5 we compare the commands from our OUC (whose design will be specified in Subsection 5.3) before and after the AGC algorithm. The rudder is not saturated, in this case the command remains unchanged. The fin angle's saturation is prevented by AGC.



(a) Rudder (b) Fins

Figure 6.5: Commands for the rudder (not saturated in the experiment) and fin angle (saturation prevented by AGC)

6.6.1 Wave disturbance

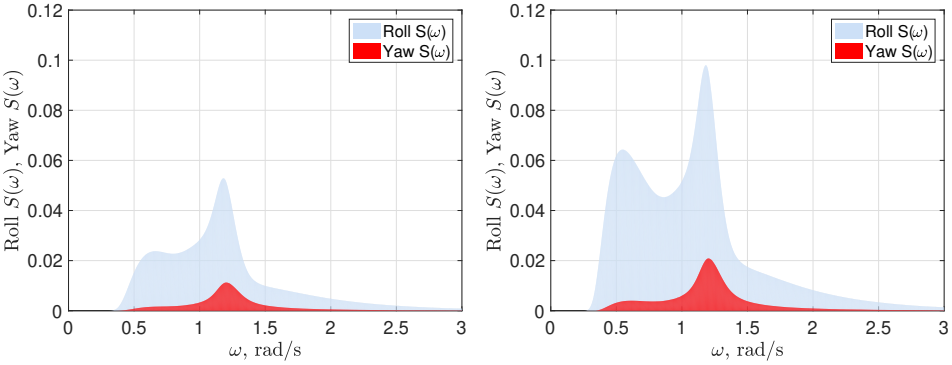
To obtain the proper time series of the polyharmonic approximation of the irregular wave (6.3) we use the methodology presented in [78, Sect. 4.2.5]. Note that the resulting realization is obtained for the *long-crested* irregular sea at 15 kts in beam seas for significant wave height of 3 m and the peak frequency 1.15 rad/s; the response amplitude operator has been taken from [78, Table B.9.]; and the number of sinusoidal component in the signal $p = 1000$. We consider the JONSWAP spectrum [25, Section 4.2.1] in the two cases: 1) significant wave height $1.5m$, which corresponds to the wave spectrum density with a single peak in Fig. 6.6(a); 2) significant wave height of 3 m, standing for the spectrum with two peaks in Fig. 6.6(b).

We use a very "rough" approximation of the signal (6.3) taking only one "dominating" sinusoidal component $p = 1$, corresponding to the peak of the power spectral density, i.e. $\omega_1 = 1.15$ rad/s. n the first case, we choose $p = 1$ and the frequency $\omega_1 = 1.15$ rad/s corresponds to the peak of the power spectral density. In the second case, $p = 2$ and we choose the peak frequencies are $\omega_1 = 1.15$ rad/s and $\omega_2 = 0.5$ rad/s.

6.6.2 The OUC design

The coefficients of the cost functional (6.22) are chosen as $\alpha = 5$, $\beta = 1$, $\gamma_1 = 20$ and $\gamma_2 = 4$.

In the design procedure from the previous sections, we choose the Hurwitz polynomial $\rho(s)$ of a simple structure, normalized (without loss of generality) in a



(a) Single-peak density (b) Double-peak density

Figure 6.6: Power spectral density of the wave disturbance

way that $\rho(0) = 1$:

$$\rho(s) = \frac{1}{\mu^9}(s+\mu)^9,$$

After this, we uniquely determine the matrix polynomial

$$r(s) = \begin{bmatrix} a_{11}s^2 + a_{12}s + a_{13} & a_{21}s^2 + a_{22}s + a_{23} \\ a_{31}s^2 + a_{32}s + a_{33} & a_{41}s^2 + a_{42}s + a_{43} \end{bmatrix},$$

where a_{ij}, $i = 1, \dots, 4$, $j = 1, 2, 3$ from the interpolation equations (6.20). According to Remark 6.5, the parameter μ has to be chosen in such a way that the transfer function $r(s)\Delta(s)/\rho(s)$ has a sufficiently narrow bandpass around the chosen frequencies ω_1 (first case) and ω_1, ω_2 (second case). After some tuning, we have chosen $\mu = 1.7$ as it gives a good (in this sense) performance.

6.6.3 Comparison with alternative controllers

We compare the designed OUC with other controllers, exploiting the linearized vessel's model. The of these controllers is the classical *LQR* control [25, Appendix D], designed to optimize the cost function

$$J_{LQR} = \int_0^\infty \left[\alpha' e_\varphi^2 + \beta' e_\psi^2 + \gamma_1' u_1^2 + \gamma_2' u_2^2\right] dt. \tag{6.25}$$

in the *absence* of disturbances. It should be noted that, in spite of similar cost functions, the behaviors of the two controllers are very different. Actually, the cost function (6.25) is in principle finite only for the disturbances vanishing as $t \to \infty$, and its optimality for $d \equiv 0$ does not guarantee any optimal performance

for the polyharmonic disturbance. For this reason, the choice of the coefficients $\alpha', \beta', \gamma_1', \gamma_2'$ is a delicate issue. We have tuned them in such a way that the LQR and OUC provide (approximately) same heading deviations (Fig. 6.9), which correspond to the coefficients $\alpha' = 5$, $\beta' = 1$, $\gamma_1' = 0.01$, $\gamma_2' = 0.001$ (notice that $\gamma_i' \neq \gamma_i$).

The conventional loop shaping controller [97] has been chosen as the third algorithm for comparison. This method ignores the yaw dynamics, working directly with the transfer function $W^0_{\varphi_u 2}$ (see 6.5). For the known dominating frequency of the disturbance, we select the structure of the notch filter centered at that point to damp the frequencies around it. The controller takes the following form

$$W_c(s) = \frac{u_1(s)}{e_\varphi(s)} = -10\frac{s^2 + 0.2(1.15)s + 1.15^2}{(s + 1.15)^2}.$$

6.6.4 Simulation results

The results of simulation are presented in Fig. 6.7–6.10, showing the dynamics of, respectively, the roll angle, the heading deviation and the rudder angle implementing aforementioned controllers. Notice that all controllers provide an average heading error 0.6 degree in the first case and 1 degree in the second case, whereas the roll damping errors visibly differ. As has been mentioned, we always combine the controllers with the AGC algorithm.

As one can see, the proposed OUC controller demonstrates the best performance among all comparing controllers, although it utilizes the control more actively. All controllers show low influence on the heading deviation; however, the performance of the roll damping is different. The loop shaping controller shows the weakest damping ability, despite the fact that it utilize the fins more than optimal regulators. This simulation demonstrates the benefits of the combined rudder-fin control strategy avoiding undesired interaction between these actuators. The LQR shows the least control efforts providing the decent roll damping performance. The main disadvantage of the LQR is the lack of knowledge about the structure of the disturbance signal. For this simulation we have considered the worst case scenario when the peak of the spectral density is located close to the natural roll frequency of the vessel; however, if it is not the case the roll damping performance may significantly degrade. A key advantage of the OUC is the simplicity of the design procedure, which does not require to know the exact shape of the wave's spectral density and, furthermore, allows to find the explicit dependence of the controller's parameters from the coefficients of the cost functional. One can adjust the coefficients of the RRS controller "on the fly", using e.g. some adaptive estimator of the disturbance's spectrum [10, 13].

6.7 Conclusions

In this chapter, we offer a novel approach to the design of the roll damping system for marine vessels, based on the idea of optimal universal controllers (OUC). Unlike the existing approaches, such a controller does not require the full information about the wave's spectral density, but only the knowledge of its dominant frequencies.

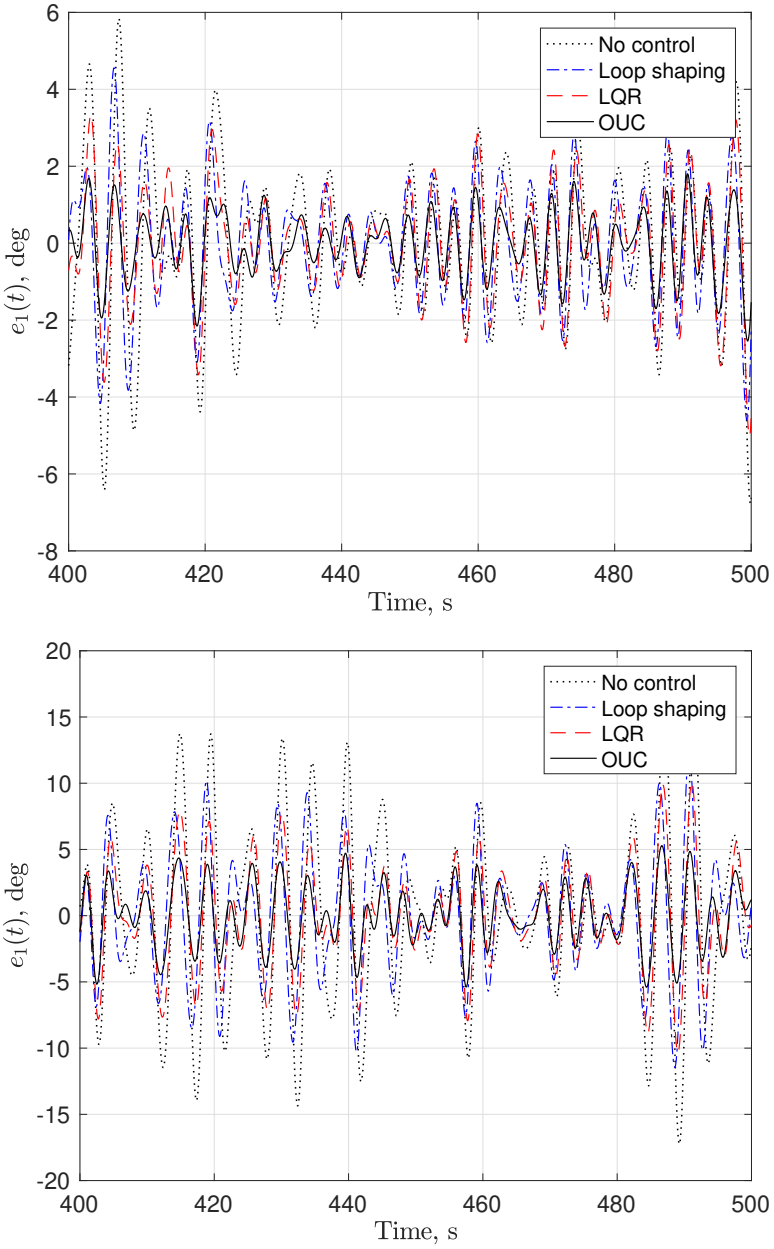


Figure 6.7: Roll angle: single-peak case (top) vs. double-peak case (bottom). Standard deviations (STDs) in the first case: 2.8803 (no control), 1.0562 (OUC), 1.8395 (LQR), 2.0666 (loop shaping). STDs in the second case: 5.0714 (no control), 2.0858 (OUC), 3.2486 (LQR), 3.5967 (loop shaping).

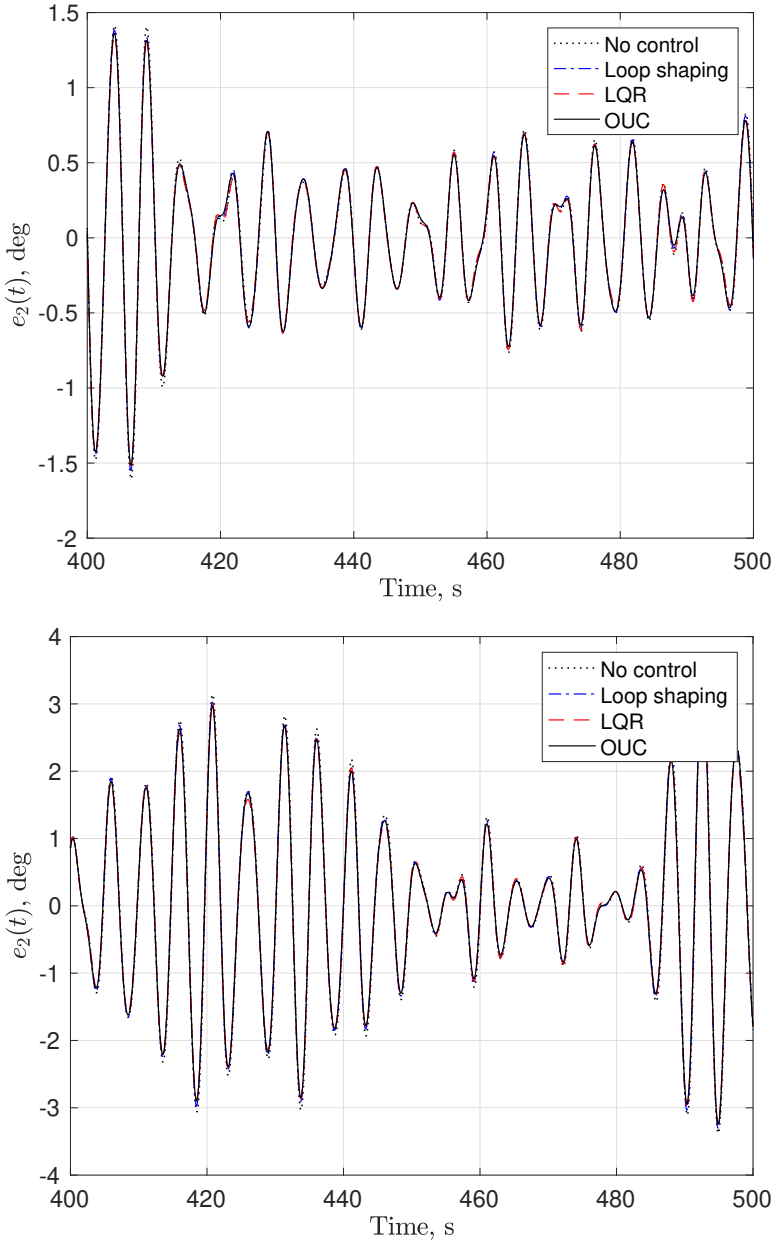


Figure 6.8: Heading error: single-peak case (top) vs. double-peak case (bottom). STDs in the first case: 0.6120 (no control), 0.5835 (OUC), 0.5825 (LQR), 0.5965 (loop shaping). STDs in the second case: 1.0570 (no control), 1.0092 (OUC), 1.0050 (LQR), 1.0300 (loop shaping).

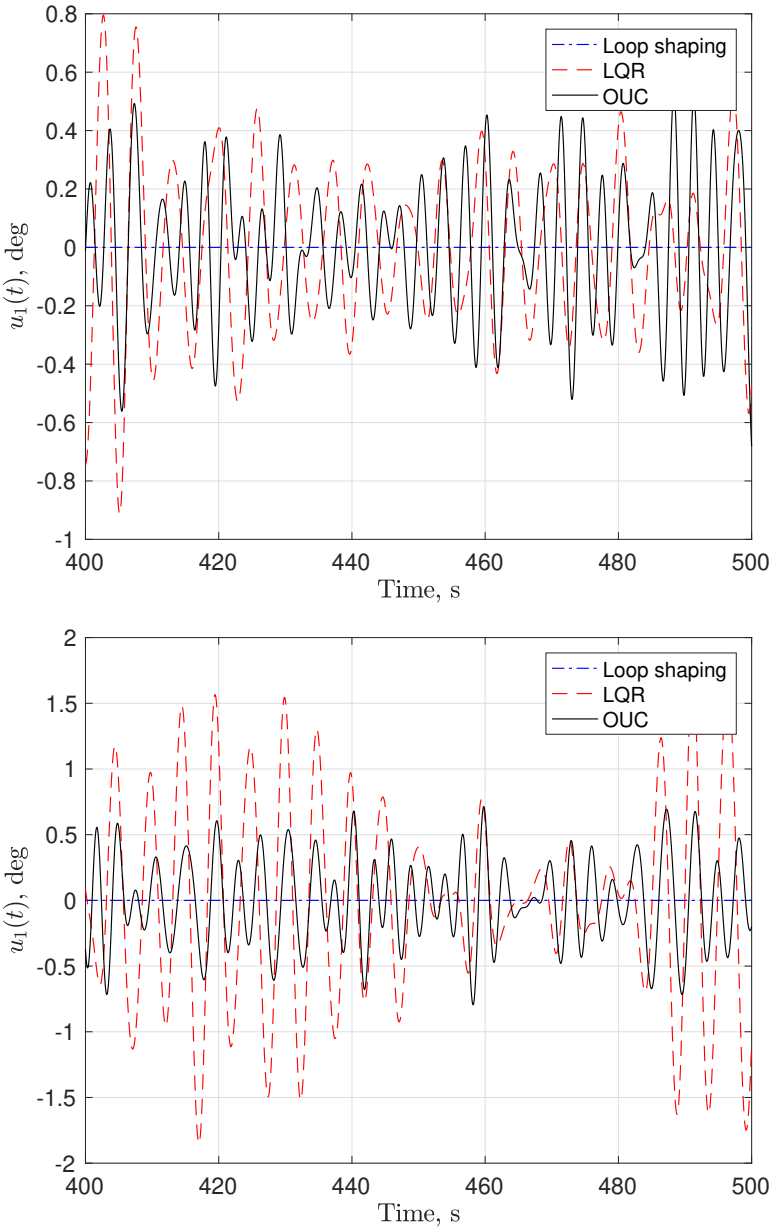


Figure 6.9: The rudder angle: single-peak case (top) vs. double-peak case (bottom)

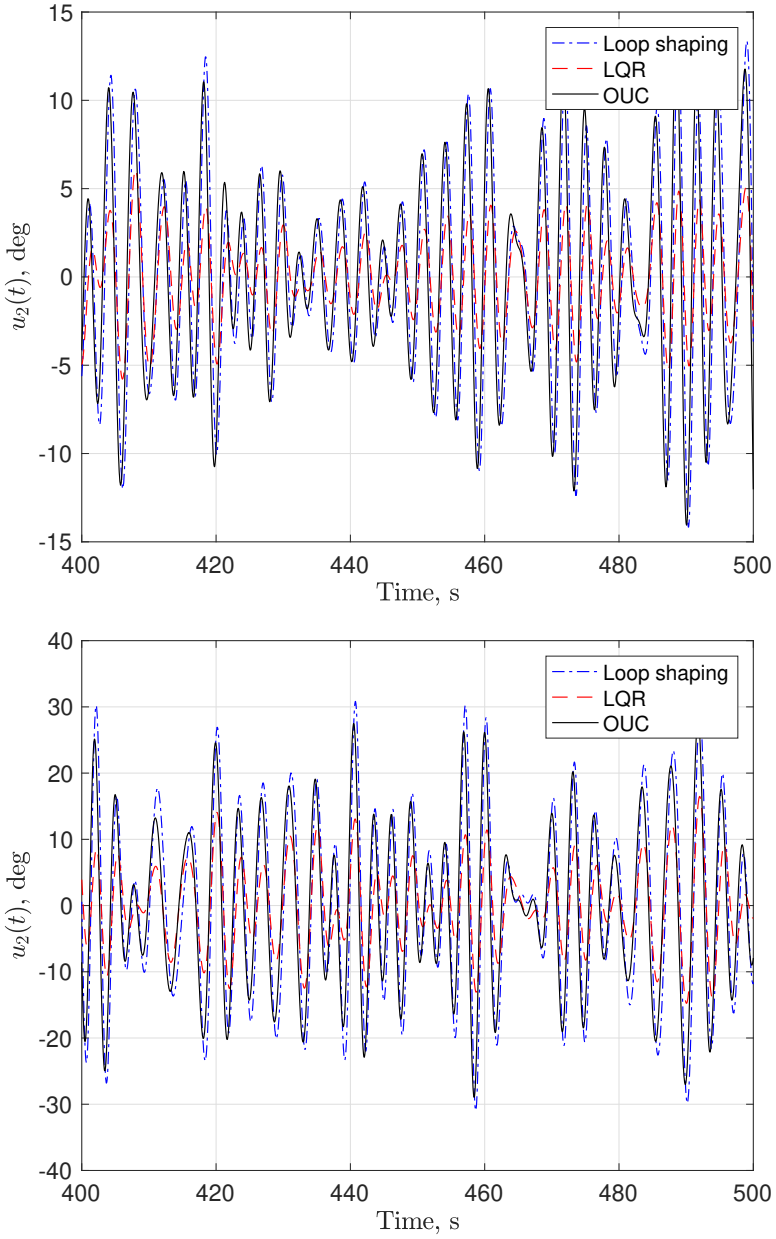


Figure 6.10: The fin angle: single-peak case (top) vs. double-peak case (bottom)

Chapter 7
Conclusions and future research

7.1 Conclusions

In this thesis we have studied two motion control problems: path following of mobile robots and roll damping for marine crafts.

In Chapter 2 , we have introduced some preliminary concepts required for the following chapters. By discussing the specific traits of ground, marine and aerial robots, we have justified our choice of the planar unicycle as the guidance model and have provided the formal definition of the path following control problem. Furthermore, we have proposed the procedure for constructing the guiding vector filed (GVF), that is the primary object of our design approach, along with some natural assumptions and useful lemmas needed of subsequent analyzes.

In Chapter 3 , we have proposed a novel algorithm for planar path following control of nonholonomic mobile robots exploiting the idea of the guiding vector field. The main advantage of our approach over the existing results is that the desired path can be an arbitrary smooth curve in its implicit form. We also have examined mathematical properties, such as stability and convergence rate, of the proposed algorithm and given global conditions for following asymptotically the desired path.

In Chapter 4 , we offer the extension of the guiding vector field algorithm for a moving path following control problem of nonholonomic robots in the presence of disturbances. In particular, we have considered the problem when the desired path is given with respect to the external coordinate frame that is uniformly moving. By using the appropriate change of coordinates, we have managed to reformulate the original moving path following problem as the motion control with respect to the stationary coordinate frame, but given in the relative coordinates.

In Chapter 5 , we have shown the results of practical experiments with real robots. These results illustrate the applicability of the path following control strategy to the different types of ground, marine, and aerial robots. Several use cases, such as the conventional path following for E-Puck robot and the fixed-wing airplane, moving path following for E-Puck robot, and coordinated path following for robotic fish, have confirmed the theoretical results of the previous chapters and demonstrated the remarkable performance of the GVF-based control algorithms.

In Chapter 6 , we have investigated the problem of the roll damping system for marine crafts. Unlike the existing methods using an approximation of the wave disturbance by the "colored noise", we have proposed a novel design approach based on the representation of wave influence by a polyharmonic process with the known spectrum, but uncertain amplitudes and phase shifts. We have utilized the idea of optimal universal controllers, developed in the works of V.A. Yakubovich, to construct the damping controller minimizing the average "power" of oscillations. The main advantage of the proposed controller is that it does not require the full information about the wave's spectral density, but only the knowledge of its dominant frequencies.

7.2 Recommendations for future research

The results presented in this thesis can be extended in several directions.

First, the idea of GVF steering can be applied to more general models than the simplest unicycle robot, considered in Chapters 2-4.

Second, the path following algorithm can be extended to the three-dimensional case. Some special results in this direction have been reported in [53, 101, 103]. The desired curvilinear path $\mathcal{P}$ in the three-dimensional space can be described as an intersection of two surfaces (Fig. 7.1)

$$\mathcal{P} \triangleq \{(x, y, z) : \varphi_1(x, y, z) = 0 \wedge \varphi_2(x, y, z) = 0\} \subset \mathbb{R}^3,$$

where $\varphi_i \in C^2(\mathbb{R}^3 \to \mathbb{R}), i = 1, 2$. The main design idea is that if the robot simultaneously approaches both surfaces, eventually it will be brought to the prescribed path $\mathcal{P}$. Thus, we may define the guiding vector field by

$$v = \tau - k_{n1} e_1 n_1 - k_{n2} e_2 n_2, \quad \tau = n_1 \times n_2$$

where $n_i = \nabla\varphi_i$ and $e_i = \varphi_i(x, y, z)$ are the "tracking errors" ($i = 1, 2$). Subsequent design of the controller, steering the robot to the GVF, is similar to the planar case (two angles are controlled instead of a single angle). The technical analysis of this algorithm requires, however, some extra assumptions of non-degeneracy and is beyond the scope of this thesis; also the convergence rate of the path following algorithm appears to be very sensitive to the choice of φ_1 and φ_2.

Third, the results of Chapter 4 can be extended to a broader class of time-varying vector fields [53], allowing to work with some all types of transformations (such as translation, rotation and scaling) of the predefined trajectory. Also, throughout this thesis, we assume the existence of a suitable state estimator that provides proper data; however, in practice, it may be impossible. Thus, the aggregation of the proposed guidance controllers with appropriate observers is an important part of

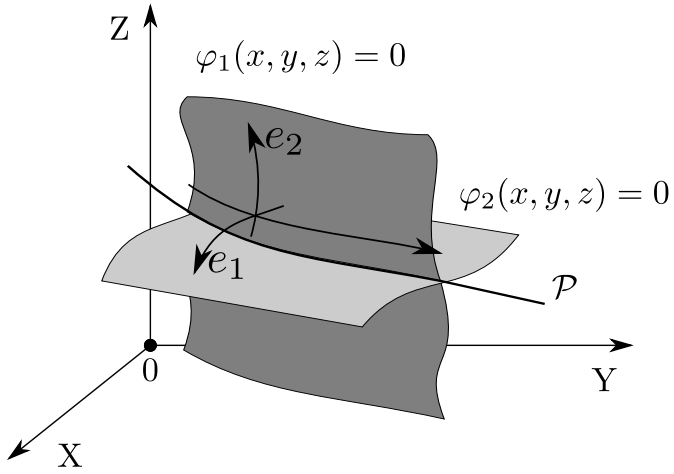


Figure 7.1: The 3D desired path as the intersection of two surfaces

the future considerations.

Fourth, in Chapter 6 we have addressed the problem of oscillation damping in the context of roll stabilization for marine crafts, however, the proposed approaches can also be used for active vibration control of aircraft [76] and automobiles [48].

Fifth, for the roll damping control, a reasonable topic of future research is to employ adaptive control methods to enable the controller's functioning in the fully uncertain environment, in particular, combining the roll damping controller with an estimator of the dominating encounter wave frequencies. Also, as it has been shown in Section 6.6, the proposed roll damping controller has been enhanced by an automatic gain controller (AGC) [22, 97, 98] preventing saturation of actuators. It could also be used with alternative gain scheduling techniques, e.g. time-varying gain reduction [52] or, more generally, advanced control allocation methods taking into account nonlinear dynamics of actuators [40, 105]. Mathematical analysis of the resulting nonlinear systems remains, however, a non-trivial problem for future research.

Finally, both motion control problems separately studied in this thesis can be merged and considered together, e.g. the path following control with simultaneous roll stabilization.

Bibliography

[1] Jürgen Ackermann, Jürgen Guldner, Wolfgang Sienel, Reinhold Steinhauser, and Vadim I. Utkin. Linear and Nonlinear Controller Design for Robust Automatic Steering. *IEEE Transactions on Control Systems Technology*, 3(1):132–143, mar 1995.

[2] A. Pedro Aguiar, João P. Hespanha, and Petar V. Kokotović. Path-following for nonminimum phase systems removes performance limitations. *IEEE Transactions on Automatic Control*, 50(2):234–239, feb 2005.

[3] Adeel Akhtar, Christopher Nielsen, and Steven L. Waslander. Path Following Using Dynamic Transverse Feedback Linearization for Car-Like Robots. *IEEE Transactions on Robotics*, 31(2):269–279, apr 2015.

[4] Brian Anderson and John B Moore. *Optimal control: linear quadratic methods*. Prentice Hall information and system sciences series, 1990.

[5] G. Antonelli. *Underwater Robots*. Springer Tracts in Advanced Robotics. Springer International Publishing, 2018.

[6] Ardupilot. UAV open-source project. http://ardupilot.org/.

[7] R. Balloch. Attitude sensors for DP. In *Proc. Dynamic Positioning Conf.*, Houston, TX, USA, 1998.

[8] Randal W. Beard, Jeff Ferrin, and Jeffrey Humpherys. Fixed wing UAV path following in wind with input constraints. *IEEE Transactions on Control Systems Technology*, 22(6):2103–2117, nov 2014.

[9] Randal W. Beard and Timothy W. McLain. *Small Unmanned Aircraft: Theory and Practice*, volume 36. Princeton University Press, 2013.

[10] Dennis J.W. Belleter, Roberto Galeazzi, and Thor I. Fossen. Experimental verification of a global exponential stable nonlinear wave encounter frequency estimator. *Ocean Engineering*, 97:48–56, mar 2015.

[11] Mogens Blanke, Jens Adrian, Knud-Erik Larsen, and Jens Bentsen. Rudder roll damping in coastal region sea conditions. In *IFAC Proceedings Volumes*, volume 33, pages 39–44. Elsevier, aug 2000.

[12] Mogens Blanke and Anders C Christensen. Rudder-roll damping autopilot robustness to sway-yaw-roll couplings. *10th Ship Control Systems Symposium, Ottawa 25-29 Oct. 1993.*, 1993.

[13] Alexey A. Bobtsov, Denis Efimov, Anton A. Pyrkin, and Ali Zolghadri. Switched algorithm for frequency estimation with noise rejection. *IEEE Transactions on Automatic Control*, 57(9):2400–2404, sep 2012.

[14] Morten Breivik and Thor I. Fossen. Principles of guidance-based path following in 2D and 3D. In *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference, CDC-ECC '05*, volume 2005, pages 627–634. IEEE, 2005.

[15] J. B. Carley. Feasibility study of steering and stabilising by rudder. In *Proc. 4th Ship control system symposium*, The Hague, The Netherlands, 1975.

[16] JB Carley and A Duberley. Design considerations for optimum ship motion control. In *Proceedings*, 1972.

[17] F.L. Chernousko. Modelling of snake-like locomotion. *Applied Mathematics and Computation*, 164(2):415 – 434, 2005. 12th International Workshop on Dynamics and Control.

[18] W.E. Cowley and T.H. Lambert. The use of the rudder as a roll stabiliser. In *Proc. 3rd Ship Control Syst. Symp.*, Bath, UK, 1972.

[19] P. Crossland. The effect of roll-stabilisation controllers on warship operational performance. *Control Engineering Practice*, 11(4):423–431, apr 2003.

[20] H.G. Hector Garcia de Marina, Yuri A. Y.A. Kapitanyuk, Murat Bronz, Gautier Hattenberger, and Ming Cao. Guidance algorithm for smooth trajectory tracking of a fixed wing UAV flying in wind flows. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5740–5745. IEEE, may 2017.

[21] GCHE Decroon, M Percin, BDW Remes, R Ruijsink, and C de Wagter. *The Delfly: Design, aerodynamics, and artificial intelligence of a flapping wing robot*. Springer, 2016.

[22] P G M der Klugt, P G M Van der Klugt, and P.G.M. Van der Klugt. *Rudder Roll Stabilization*. PhD thesis, TU Delft, 1987.

[23] Mohamed I. El-Hawwary and Manfredi Maggiore. Case studies on passivity-based stabilisation of closed sets. *International Journal of Control*, 84(2):336–350, feb 2011.

[24] Giuseppe Fedele and Andrea Ferrise. Non adaptive second-order generalized integrator for identification of a biased sinusoidal signal. *IEEE Transactions on Automatic Control*, 57(7):1838–1842, jul 2012.

[25] Thor I. Fossen. *Guidance and control of ocean vehicles*. Wiley, 1994.

[26] Thor I. Fossen and Kristin Y. Pettersen. On uniform semiglobal exponential stability (USGES) of proportional line-of-sight guidance laws. *Automatica*, 50(11):2912–2917, nov 2014.

[27] Thor I. Fossen, Kristin Y. Pettersen, and Roberto Galeazzi. Line-of-sight path following for dubins paths with adaptive sideslip compensation of drift forces. *IEEE Transactions on Control Systems Technology*, 23(2):820–827, mar 2015.

[28] Alexander L. Fradkov, Iliya V. Miroshnik, and Vladimir O. Nikiforov. *Nonlinear and Adaptive Control of Complex Systems*. Springer Netherlands, Dordrecht, 1999.

[29] Rajan J. Gill, Dana Kulic, and Christopher Nielsen. Spline path following for redundant mechanical systems. *IEEE Transactions on Robotics*, 31(6):1378–1392, dec 2015.

[30] Vincius M. Goncalves, Luciano C.A. Pimenta, Carlos A. Maia, Bruno C.O. Dutra, and Guilherme A.S. Pereira. Vector fields for robot navigation along time-varying curves in n-dimensions. *IEEE Transactions on Robotics*, 26(4):647–659, aug 2010.

[31] Graham C. Goodwin, Tristan Pérez, Maria Serón, and Ching Yaw Tzeng. On fundamental limitations for rudder roll stabilization of ships. *Proceedings of the IEEE Conference on Decision and Control*, 5:4705–4710, 2000.

[32] O. N. Granichin and T. A. Khantuleva. Adapting wing elements ("feathers") of an airplane in a turbulent flow with a multiagent protocol. *Automation and Remote Control*, 78(10):1867–1882, Oct 2017.

[33] Gerald Hearns and Mogens Blanke. Quantitative Analysis and Design of a Rudder Roll Damping Controller. In *IFAC Proceedings Volumes*, volume 31, pages 105–110. Elsevier, oct 1997.

[34] M. A. Hinostroza, Weilin Luo, and C. Guedes Soares. Robust fin control for ship roll stabilization based on L2-gain design. *Ocean Engineering*, 94:126–131, jan 2015.

[35] Andre Hladio, Christopher Nielsen, and David Wang. Path following for a class of mechanical systems. *IEEE Transactions on Control Systems Technology*, 21(6):2380–2390, nov 2013.

[36] Isaac Horowitz and Marcel Sidi. Optimum synthesis of non-minimum phase feedback systems with plant uncertainty. *International Journal of Control*, 27(3):361–386, mar 1978.

[37] M. Hou. Parameter Identification of Sinusoids. *IEEE Transactions on Automatic Control*, 57(2):467–472, feb 2012.

[38] Michael Hoy, Alexey S. Matveev, and Andrey V. Savkin. Algorithms for collision-free navigation of mobile robots in complex cluttered environments: A survey. *Robotica*, 33(3):463–497, mar 2015.

[39] Alberto. Isidori. *Nonlinear control systems*. Springer, 1995.

[40] Tor A. Johansen and Thor I. Fossen. Control allocation - A survey. *Automatica*, 49(5):1087–1103, may 2013.

[41] Tor A. Johansen, Thomas P. Fuglseth, Petter Tøndel, and Thor I. Fossen. Optimal constrained control allocation in marine surface vessels with rudders. *Control Engineering Practice*, 16(4):457–464, apr 2008.

[42] Y.A. Kapitanyuk, H.G.D. Marina, A.V. Proskurnikov, and M. Cao. Guiding vector field algorithm for a moving path following problem. *IFAC-PapersOnLine*, 50(1), 2017.

[43] Y.A. Kapitanyuk, A.V. Proskurnikov, and M. Cao. Optimal controllers for rudder roll damping with an autopilot in the loop. *IFAC-PapersOnLine*, 49(23), 2016.

[44] Y.A. Kapitanyuk, A.V. Proskurnikov, and M. Cao. A Guiding Vector-Field Algorithm for Path-Following Control of Nonholonomic Mobile Robots. *IEEE Transactions on Control Systems Technology*, 26(4), 2018.

[45] Yu. A. Kapitanyuk and S. A. Chepinsky. Control of mobile robot following a piecewise-smooth path. *Gyroscopy and Navigation*, 4(4):198–203, oct 2013.

[46] Yuri A. Kapitanyuk, Sergey A. Chepinskiy, and Aleksandr A. Kapitonov. Geometric path following control of a rigid body based on the stabilization of sets. In *IFAC Proceedings Volumes (IFAC-PapersOnline)*, volume 47, pages 7342–7347. Elsevier, 2014.

[47] Yuri A. Kapitanyuk, Anton V. Proskurnikov, and Ming Cao. Optimal Universal Controllers for Roll Damping. *Ocean Engineering (under review)*, 2019. online at https://arxiv.org/abs/1707.07277.

[48] H. Karkosch, F. Svaricek, R. A. Shoureshi, and J. L. Vance. Automotive applications of active vibration control. In *1999 European Control Conference (ECC)*, pages 581–585, Aug 1999.

[49] E. Kelasidi, A.M. Kohl, K.Y. Pettersen, B.H. Hoffmann, and J.T. Gravdahl. Experimental investigation of locomotion efficiency and path-following for underwater snake robots with and without a caudal fin. *Annual Reviews in Control*, 46:281 – 294, 2018.

[50] Hassan K. Khalil. *Nonlinear systems*. Prentice Hall, 2002.

[51] Sergey A. Kolyubin and Anton S. Shiriaev. Dynamics-consistent motion planning for underactuated ships using virtual holonomic constraints. In *2014 Oceans - St. John's, OCEANS 2014*, pages 1–7. IEEE, sep 2015.

[52] T. Laudval and T.I. Fossen. Nonlinear rudder-roll damping of non-minimum phase ships using sliding mode control. In *Proc. of the European Control Conference, Brussel, Belgium*, pages 1689–1694. IEEE, jul 1997.

[53] Dale A. Lawrence, Eric W. Frew, and William J. Pisano. Lyapunov Vector Fields for Autonomous Unmanned Aircraft Flight Control. *Journal of Guidance, Control, and Dynamics*, 31(5):1220–1229, sep 2008.

[54] Anders Lindquist and Vladimir A. Yakubovich. Optimal damping of forced oscillations in discrete-time systems. *IEEE Transactions on Automatic Control*, 42(6):786–802, jun 1997.

[55] Anders Lindquist and Vladimir A. Yakubovich. Universal regulators for optimal tracking in discrete-time systems affected by harmonic disturbances. *IEEE Transactions on Automatic Control*, 44(9):1688–1704, 1999.

[56] Zhiquan Liu, Hongzhang Jin, Michael J. Grimble, and Reza Katebi. Ship forward speed loss minimization using nonlinear course keeping and roll motion controllers. *Ocean Engineering*, 113:201–207, feb 2016.

[57] A.E.J.M. Lloyd. Roll stabilisation by rudder. In *Proc. 4th Ship control system symposium*, The Hague, The Netherlands, 1975.

[58] M. S. Longuet-Higgins. The effect of non-linearities on statistical distributions in the theory of sea waves. *Journal of Fluid Mechanics*, 17(3):459–480, nov 1963.

[59] L. Marconi and A. Isidori. Mixed internal model-based and feedforward control for robust tracking in nonlinear systems. *Automatica*, 36(7):993–1000, jul 2000.

[60] Osama A. Marzouk and Ali H. Nayfeh. Control of ship roll using passive and active anti-roll tanks. *Ocean Engineering*, 36(9-10):661–671, jul 2009.

[61] Alexey S. Matveev, Michael Hoy, Jayantha Katupitiya, and Andrey V. Savkin. Nonlinear sliding mode control of an unmanned agricultural tractor in the presence of sliding and control saturation. *Robotics and Autonomous Systems*, 61(9):973–987, sep 2013.

[62] Alexey S. Matveev, Michael C. Hoy, and Andrey V. Savkin. Extremum Seeking Navigation Without Derivative Estimation of a Mobile Robot in a Dynamic Environmental Field. *IEEE Transactions on Control Systems Technology*, 24(3):1084–1091, may 2016.

[63] Alexey S. Matveev, Kirill S. Ovchinnikov, and Andrey V. Savkin. A method of reactive 3D navigation for a tight surface scan by a nonholonomic mobile robot. *Automatica*, 75:119–126, jan 2017.

[64] Alain Micaelli, Claude Samson, et al. Trajectory tracking for unicycle-type and two-steering-wheels mobile robots. *Tech. Rep. No. 2097, INRIA*, 1993.

[65] Francesco Mondada, Michael Bonani, Xavier Raemy, James Pugh, Christopher Cianci, Adam Klaptocz, Stephane Magnenat, Jean-Christophe Zufferey, Dario Floreano, and Alcherio Martinoli. The e-puck, a robot designed for education in engineering. In *Proc. of the 9th Conf. on Auton. Robot Syst. and Competitions*, pages 59–65, 2009.

[66] P. Morin and C. Samson. Trajectory tracking for non-holonomic vehicles: overview and case study. In *Proceedings of the Fourth International Workshop on Robot Motion and Control (IEEE Cat. No.04EX891)*, pages 139–153. IEEE, 2004.

[67] D.R. Nelson, D.B. Barber, T.W. McLain, and R.W. Beard. Vector Field Path Following for Miniature Air Vehicles. *IEEE Transactions on Robotics*, 23(3):519–529, jun 2007.

[68] V. Nicolau, C. Miholcă, D. Aiordachioaie, and E. Ceangă. QFT autopilot design for robust control of ship course-keeping and course-changing problems. *Control Engineering and Applied Informatics*, 7(1):44–56, 2005.

[69] Hyondong Oh, Seungkeun Kim, Hyo Sang Shin, and Antonios Tsourdos. Coordinated standoff tracking of moving target groups using multiple UAVs.

IEEE Transactions on Aerospace and Electronic Systems, 51(2):1501–1514, apr 2015.

[70] Tiago Oliveira, A. Pedro Aguiar, and Pedro Encarnacao. Moving Path Following for Unmanned Aerial Vehicles with Applications to Single and Multiple Target Tracking Problems. *IEEE Transactions on Robotics*, 32(5):1062–1078, oct 2016.

[71] Giuseppe Oriolo, Alessandro De Luca, and Marilena Vendittelli. WMR control via dynamic feedback linearization: Design, implementation, and experimental validation. *IEEE Transactions on Control Systems Technology*, 10(6):835–852, nov 2002.

[72] Derek A. Paley, Fumin Zhang, and Naomi Ehrich Leonard. Cooperative control for ocean sampling: The glider coordinated control system. *IEEE Transactions on Control Systems Technology*, 16(4):735–744, jul 2008.

[73] Dimitra Panagou and Vijay Kumar. Cooperative visibility maintenance for leader-follower formations in obstacle environments. *IEEE Transactions on Robotics*, 30(4):831–844, aug 2014.

[74] Paparazzi. UAV open-source project. http://wiki.paparazziuav.org/.

[75] Sanghyuk Park, John Deyst, and Jonathan How. A New Nonlinear Guidance Logic for Trajectory Tracking. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Reston, Virigina, aug 2004. American Institute of Aeronautics and Astronautics.

[76] J. T. Pearson, R. M. Goodall, and I. Lyndon. Active control of helicopter vibration. *Computing Control Engineering Journal*, 5(6):277–284, Dec 1994.

[77] Tristan Perez. *Ship Motion Control*. Springer, 2005.

[78] Tristan Perez and Mogens Blanke. Ship roll damping control, apr 2012.

[79] Rafael Potrie and Pablo Monzón. Local implications of almost global stability. *Dynamical Systems*, 24(1):109–115, mar 2009.

[80] A. V. Proskurnikov and V. A. Yakubovich. Universal controllers in model matching optimal control problems for unknown external signals. *Journal of Computer and Systems Sciences International*, 51(2):214–227, apr 2012.

[81] Anton V. Proskurnikov. Universal controllers of V.A. Yakubovich: a systematic approach to LQR problems with uncertain external signals. *IFAC-PapersOnLine*, 48(11):557–562, jan 2015.

[82] A.V. Proskurnikov and V.A. Yakubovich. Universal regulators for optimal tracking of stochastic signals with an unknown spectral density. *Doklady Mathematics*, 74(1):147–151, jan 2006.

[83] Alessandro Renzaglia, Christophe Reymann, and Simon Lacroix. Monitoring the evolution of clouds with UAVs. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume 2016-June, pages 278–283. IEEE, may 2016.

[84] Claude Samson. Path Following And Time-Varying Feedback Stabilization of a Wheeled Mobile Robot. *Second International Conference on Automation, Robotics and Computer Vision*, pages 1–14, 1992.

[85] Claude Samson. Control of Chained Systems Application to Path Following and Time-Varying Point-Stabilization of Mobile Robots. *IEEE Transactions on Automatic Control*, 40(1):64–77, 1995.

[86] M. M. Seron, J. H. Braslavsky, P. V. Kokotović, and D. Q. Mayne. Feedback limitations in nonlinear systems: From Bode integrals to cheap control. *IEEE Transactions on Automatic Control*, 44(4):829–833, apr 1999.

[87] M. T. Sharif, G. N. Roberts, and R. Sutton. Sea-trial experimental results of fin/rudder roll stabilisation. *Control Engineering Practice*, 3(5):703–708, may 1995.

[88] Bruno Siciliano and Oussama Khatib, editors. *Springer Handbook of Robotics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[89] D. Soetanto, L. Lapierre, and A. Pascoal. Adaptive, non-singular path-following control of dynamic wheeled robots. In *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, volume 2, pages 1765–1770. IEEE, 2003.

[90] Jakob Stoustrup, H.H. Niemann, and M. Blanke. Roll damping by rudder control-a new H/sub /spl infin// approach. In *Proceedings of IEEE International Conference on Control and Applications CCA-94*, volume 2, pages 839–844. IEEE, 1994.

[91] Knut Streitlien, George S. Triantafyllou, and Michael S. Triantafyllou. Efficient foil propulsion through vortex control. *AIAA Journal*, 34(11):2315–2319, 1996.

[92] P. B. Sujit, Srikanth Saripalli, and Joao Borges Sousa. Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicless. *IEEE Control Systems*, 34(1):42–59, feb 2014.

[93] S. Surendran, S. K. Lee, and S. Y. Kim. Studies on an algorithm to control the roll motion using active fins. *Ocean Engineering*, 34(3-4):542–551, mar 2007.

[94] Antonios. Tsourdos, Brian White, and Madhavan. Shanmugavel. *Cooperative Path Planning of Unmanned Aerial Vehicles, ser. Advances in Unmanned Marine Vehicles*. Wiley, 2011.

[95] G. Ullrich and P.A. Kachur. *Automated Guided Vehicle Systems: A Primer with Practical Applications*. Springer Berlin Heidelberg, 2014.

[96] Luis Ariel Valbuena Reyes and Herbert G. Tanner. Flocking, Formation Control, and Path Following for a Group of Mobile Robots. *IEEE Transactions on Control Systems Technology*, 23(4):1268–1282, jul 2015.

[97] J. van Amerongen, P. G.M. van der Klugt, and H. R. van Nauta Lemke. Rudder roll stabilization for ships. *Automatica*, 26(4):679–690, jul 1990.

[98] J. van Amerongen, H.R. van Nauta Lemke, and P.G.M van der Klugt. Rudder Roll Stabilization controller design based on an adaptive criterion. In *1986 American Control Conference*, pages 237–242. IEEE, jun 1986.

[99] Evgeny I. Veremey. Dynamical correction of control laws for marine ships' accurate steering. *Journal of Marine Science and Application*, 13(2):127–133, jun 2014.

[100] J. Wang, A.Y. Krasnov, Y.A. Kapitanyuk, S.A. Chepinskiy, Y. Chen, and H. Liu. Path following control algorithms implemented in a mobile robot with omni wheels. *Gyroscopy and Navigation*, 7(4), 2016.

[101] Jian Wang, Iurii A. Kapitaniuk, Sergey A. Chepinskiy, Dongliang Liu, and Aleksandr J. Krasnov. Geometric path following control in a moving frame. *IFAC-PapersOnLine*, 48(11):150–155, 2015.

[102] V. A. Yakubovich. Universal Regulators in Linear-Quadratic Optimization Problems. *Trends in Control*, pages 53–68, 1995.

[103] W. Yao, Y. A. Kapitanyuk, and M. Cao. Robotic path following in 3d using a guiding vector field. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 4475–4480, Dec 2018.

[104] J. Yuh, Giacomo Marani, and D. Richard Blidberg. Applications of marine robotic vehicles. *Intelligent Service Robotics*, 4(4):221, Jul 2011.

[105] Luca Zaccarian. Dynamic allocation for input redundant control systems. *Automatica*, 45(6):1431–1438, jun 2009.

Summary

In this thesis, several theoretical and practical problems related to motion control of ground robots, aerial vehicles and marine crafts are addressed.

First, we address the planar path following control problem by using the guiding vector field (GVF) based approach. In Chapter 2, we propose the procedure of constructing the vector field, using the implicit representation of the desired path (as an algebraic equation). Although the GVF may be undefined in some "degenerate" points, we prove that under natural assumptions (that hold for many practically important trajectories) almost all integral curves of the GVF avoid these exceptional points and converge to the desired trajectory. We demonstrate that a robot that ideally follows an integral curve will thus reach the desired path. In Chapter 3, we propose a control algorithm for a basic nonholonomic unicycle model that steers the robot along the trajectories of the GVF. We examine stability and convergence properties of this algorithm. Due to presence of degeneracy points in the vector field, in general it is usually impossible to prove global convergence to the desired trajectory. We however explicitly estimate a region of initial conditions, for which such a convergence can be established.

Second, we extend the aforementioned results to the problem of *moving path following*, where the desired path is defined with respect to an external moving reference frame and the robot is influenced by a constant (or slowly changing) external disturbance. Using the appropriate change of coordinates and a reasonable assumption on the magnitude of the disturbance one can transform the moving path following problem to the conventional path-following problem (Chapter 4).

Third, in Chapter 6 we address the problem of oscillation damping, specific for marine robotics (the methods developed to solve this problem, however, can be extended to oscillation damping in ground vehicles and aircrafts). Namely, the problem of rudder roll stabilization (RRS) is considered for a marine vessel equipped with a "rudder" (generally, a group of oscillators sufficient for the yaw steering), stabilizing fins and the autopilot that has to share the "rudder" with a roll

stabilization system. Using the theory of optimal universal controllers developed in the works of V.A. Yakubovich, we design a roll stabilizing controller, which guaranties optimal damping of roll oscillations for a wave induced disturbance from a given class (polyharmonic signals with a known spectrum, but uncertain amplitudes and phases). The optimality is understood in the sense of average "power" of oscillation.

Finally, the results of each chapter are confirmed by numerical simulations. Besides this, several experiments with real robots are reported (Chapter 5) that confirm the practical applicability of the proposed path-following algorithms. Chapter 7 is devoted to conclusions, and also proposes some directions for future research.

Samenvatting

In deze thesis worden verschillende theoretische en praktische problemen behandeld die gerelateerd zijn aan de motion control van landrobots, luchtvaartuigen en marinevoertuigen.

Ten eerste wordt het vlak padvolging probleem behandeld door gebruik te maken van de Guiding Vector Field (GVF) methode. In Hoofdstuk 2 ontwikkelen we een procedure voor het construeren van vectorvelden op basis van de impliciete representatie van het gewenste pad als een algebraïsche vergelijking. Hoewel de GVF niet gedefinieerd kan zijn in sommige "gedegenereerde" punten bewijzen we dat, onder natuurlijke aannames (die vaak gelden in de praktijk), bijna alle integrale curves van de GVF deze exceptionele punten ontwijken en convergeren naar het gewenste traject. We demonstreren dat een robot die een integrale curve ideaal volgt het gewenste pad bereikt. In Hoofdstuk 3 wordt een regeltechniek algoritme voorgesteld voor een anholonoom eenwieler model dat de robot aanstuurt langs de trajecten van de GVF. We bestuderen de stabiliteit en convergentie eigenschappen van dit algoritme. De aanwezigheid van gedegenereerde punten in het vectorveld maakt het, in het algemeen, onmogelijk om globale convergentie naar het gewenste traject te bewijzen. Wij maken echter explicite schattingen van een regio van initieel voorwaarden waarvoor een dergelijke convergentie kan worden vastgesteld.

Ten tweede wordt het hiervoorgenoemde resultaat uitgebreid voor het *bewegende padvolging* probleem waarin het gewenste pad gedefinieerd wordt aan de hand van een extern bewegend referentiekader, en de robot door een een constante (of langzaam veranderende) externe verstoring wordt beïnvloed. Door een toepasselijke verandering van coordinaten, en een redelijke aanname over de magnitude van de verstoring, kan het bewegende padvolging probleem getransformeerd worden naar het conventionele padvolging probleem (Hoofdstuk 4).

Ten derde wordt in Hoofdstuk 6 het oscillatie demping probleem specifiek voor marine robotica behandeld (hoewel de ontwikkelde methodes uitgebreid kunnen worden naar oscillatie demping voor landrobots en luchtvaartuigen). Het probleem

van Rudder Roll Stabilization (RRS) wordt onderzocht voor een marinevoertuig uitgerust met een “rudder” (een groep van oscillatoren voor gieren sturing), stabilisatie vinnen, en een autopilot die de rudder deelt met een slinger stabilisatie systeem. Door gebruik te maken van de optimale universele controllers ontwikkeld in de werken van V.A. Yakubovich, ontwerpen we een slinger stabilisatie controller die optimale demping garandeert voor slinger oscillaties veroorzaakt door een golf verstoring van een bepaalde klasse (polyharmonische signalen met een bekend spectrum maar onzeker amplitude en fase). De optimaliteit van deze methode is in termen van de gemiddelde “kracht” van de oscillatie.

Ten laatste worden de resultaten van elk hoofdstuk bevestigd door numerieke simulaties. Daarnaast worden verschillende experimenten met echte robots gerapporteerd (Hoofdstuk 5) die de praktische toepasbaarheid van de voorgestelde padvolging algoritmes bevestigen. Hoofdstuk 7 concludeert de thesis en stelt een aantal toekomstige onderzoeksrichtingen voor.