ALIGNMENT CONTROL FOR OPTICAL COMMUNICATION BETWEEN UNDERWATER ROBOTS

By

Pratap Bhanu Solanki

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Electrical Engineering – Doctor of Philosophy

2021

ABSTRACT

ALIGNMENT CONTROL FOR OPTICAL COMMUNICATION BETWEEN UNDERWATER ROBOTS

By

Pratap Bhanu Solanki

Light-emitting diode (LED)-based optical communication is emerging as a promising low-power, low-cost, and high-data-rate alternative to acoustic communication for underwater applications. However, it requires a close-to-line-of-sight (LOS) link between the communicating parties. Achieving and maintaining the LOS is challenging due to the constant movement of underlying mobile platforms caused by propulsion and unwanted disturbances. In this dissertation, a novel, compact LED-based wireless communication system with active alignment control is presented that maintains the LOS despite the movement of the underlying platform. Multiple alignment control algorithms are developed for scenarios that range from a simple one-way two-dimensional (2D) setting to a practical three-dimensional (3D) bi-directional underwater setting.

An extended Kalman filter (EKF)-based approach is first proposed to estimate the relative orientation between the heading angle and the LOS direction, which is subsequently used for alignment control. The EKF uses only the measurement of light intensity from a single photodiode, where successive measurements are obtained via a scanning technique that ensures the full observability of the underlying system. The approach is first examined in a 2D setting, and then extended to the 3D scenario with improvements in both the hardware and the algorithm. The amplitude of the scanning is modulated according to the alignment performance to achieve a sound trade-off between estimation accuracy, signal strength, and energy consumption. The efficacy of the approach is tested and verified via simulation and on an experimental setup involving two robots with relative 3D motion.

The EKF approach uses an assumption that the relative motion between the robots is small, and consequently, requires the communicating robots to take the scanning in an alternating fashion for the convergence of the estimator. An alternative approach, first explored in the 2D setting, is developed that allows simultaneous, bi-directional alignment control for both parties. Because of the convex nature of the measured intensity functions, model-free approaches, including both hill-climbing (HC) and extremum-seeking (ES), are explored. The hill-climbing approach is found to be superior to the ES approach in terms of convergence time and computational efficiency. Theoretical analysis is provided for the hill-climbing approach that guarantees finite time convergence to an $O(\delta)$ neighborhood of the LOS, for control step size δ.

Finally, a model-free approach for the 3D setting is proposed that maximizes light intensity based on three consecutive intensity measurements from an equilateral triangle configuration. The efficacy of the approach is demonstrated experimentally, first with an underwater robot controlled by a joystick via LED communication and then with two robots performing bi-directional communication and tracking in an underwater setting.

Dedicated to my parents and my brother
for their love and support through this journey.

ACKNOWLEDGEMENTS

Throughout my Ph.D. journey, I have received a great deal of support and assistance.

Firstly, I would like to express my sincere gratitude to my advisor, Prof. Xiaobo Tan, for the continuous support of my Ph.D. study and related research, for his patience, motivation, and guidance. Besides my advisor, I would like to thank my entire thesis committee: Prof. Hassan Khalil, Prof. Ranjan Mukherjee, and Prof. Daniel Morris, for their insightful comments and encouragement. My sincere thanks also goes to Prof. Shaunak Bopardikar for the stimulating discussions and insights.

I thank John Thon for his fabrication expertise and his help in preparing mechanical parts, especially making waterproof enclosures for the robots. I also thank Austin Coha, Shaswat Joshi, Jason Greenberg, and Max Verboncoeur for their help in preparing the experimental setup and in performing experiments that are presented in this work. I thank Brian Fickies for his coordination and granting us seamless access to the IM Circle swimming pool facility at Michigan State University. I thank my fellow labmates for their support and for all the fun we have had in the last few years. Thanks also to all of the administrators and staff members in the ECE department who have helped me over the years.

I also want to acknowledge the funding support from National Science Foundation (IIS 1319602, IIP 1343413, CCF 1331852, ECCS 1446793, IIS 1734272, IIS 1848945) that made this work possible.

Most of all, I would like to thank my family: my parents and my brother for supporting my decision to come to the USA for pursuing the Ph.D., their patience, and providing me love, support, and encouragement throughout this journey. I want to praise and thank God, the almighty, who has granted countless blessings, knowledge, opportunity, strength, and perseverance in life and especially during this period of the pandemic.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

With the recent advancements in technology, the use of autonomous underwater vehicles (AUVs) has become increasingly popular for underwater exploration [56], with application to marine sciences, environmental engineering, and oil/gas exploration among others. One essential attribute of these robots, while they are deployed, is to stay connected with each other or with a base station via a wireless communication link. Due to the substantial attenuation of radio frequency signals in water [9], acoustic communication is currently the industry standard for underwater communication, with a range of up to tens of kilometers [53]. However, underwater acoustics suffers from shortcomings like latency, low data rates, and high power consumption [13]. Recently, optical wireless communication is emerging as a promising alternative or complementary solution to the acoustics communication for low-to-medium range data transfer applications, due to its properties such as low power, low cost, and high data rate [19, 29]. Figure 1.1 illustrates different promising application scenarios of wireless underwater optical communication technology.

Many of the recent works in optical communication systems focused on increasing the communication range and/or data rate using narrow beams (e.g., laser). Oubei *et al.* showed a 2.3 Gbits/s link over a distance of 7 m [37]. Liu *et al.* demonstrated 2.7 Gbits/s at 34.5 m using a 520 nm green laser diode (LD) [28]. Wu *et al.* used a 450-nm blueGaN laser diode (LD) directly modulated by pre-leveled 16-quadrature amplitude modulation (QAM) to achieve 12.2 and 5.6 Gbits/s data rate at a distance of 1.7 m and 10.2 m, respectively [55].

1.1 LED Optical Communication and the Problem of Directionality

Over the past few years, light-emitting diode (LED)-based optical communication has been proposed as a promising low-power, low-cost, high-rate solution for low-to-medium range underwater data transfer [18, 19, 29]. Several studies focused on increasing the range and data rates of LED communication. Brundage reported an optical communication system using a Titan blue

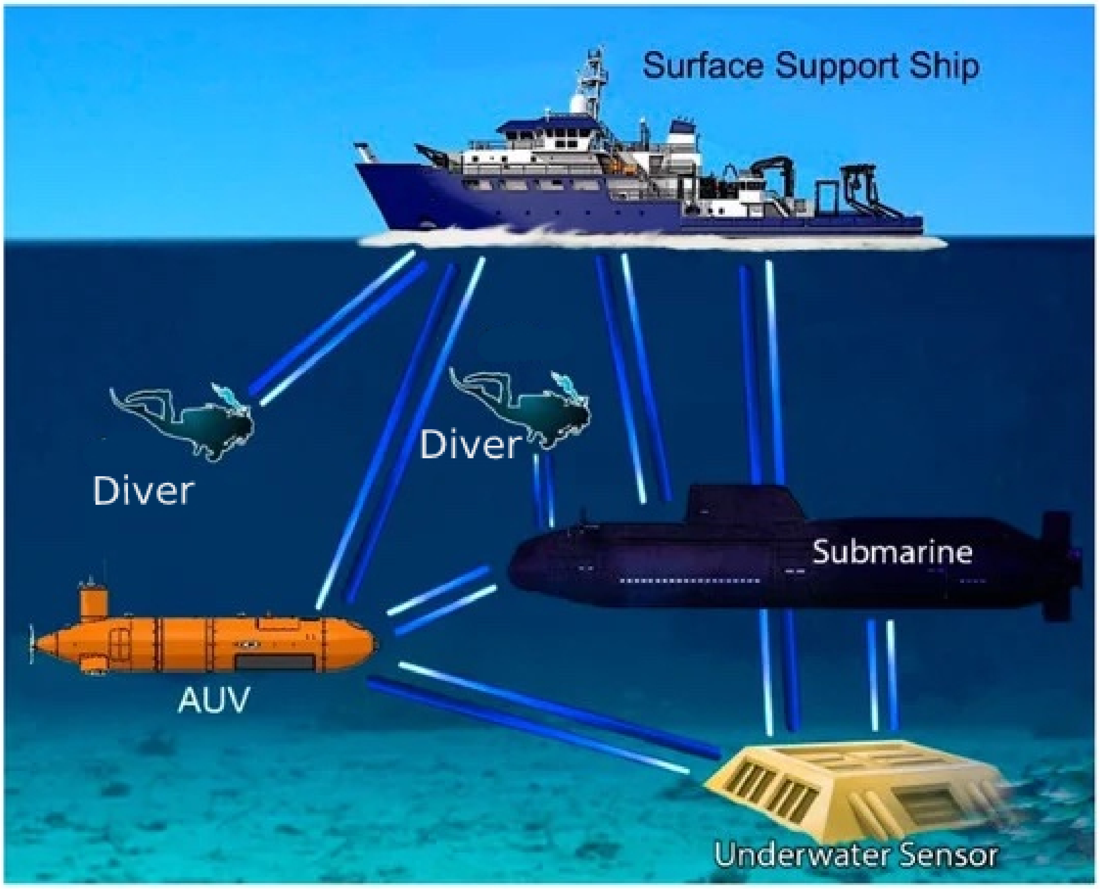


Figure 1.1: Illustration of application of underwater wireless optical communication in diverse scenarios (Source: Università degli Studi, Italy [42]).

lighting LED [8], which performed error-free communication over 1 Mbps at distances up to 13 m. Doniec and Rus demonstrated a bidirectional underwater wireless communication system called AquaOptical II [16], which used 18 Luxeon Rebel LEDs and an avalanche photodiode (APD) and operated over a distance of 50 m at a data rate of 4 Mbps.

An inherent challenge in wireless optical communication is that light signals are highly directional and thus close-to-line-of-sight links are required. For many intended applications involving mobile platforms (in particular, underwater robots), maintaining line of sight (LOS) is difficult due to constant movement of the platform caused by propulsion or ambient disturbances. Several approaches have been proposed to address the line-of-sight requirement in optical communication systems. Pontbriand *et al*. increased the field of view (FOV) of the receiver by using large-area photomultiplier tubes (~ 20 inch), to avoid the need for active pointing during communication [38]. Anguita *et al.* implemented a transmitter that used 12 LEDs arranged on a circle to transmit omnidirectionally in the plane [4], [5]. They tested their system at a rate of 100 kbps for distances of up to 2 m. Rust and Asada [41] used a high-power LED and 8 photodiodes arranged in a circle to control an underwater robot, where the system was able to transmit at 100 kbps at a distance of 23 m. Simpson *et al.* reported a system where the receiver had a 3-D spherical array of 7 lenses all focusing on a 2-D planar array of 7 photodiodes, and the transmitter consisted of a truncated hexagonal pyramid with 7 LEDs and 7 lenses [44]. Most of the aforementioned systems achieved the line-of-sight through redundancy in transmitters and/or receivers, which resulted in larger footprint, higher cost, and higher complexity.

Modulating-retro-reflector (MRR)-based asymmetric pointing systems have been demonstrated [14,17,21,36], where an MRR terminal is attached to a mobile robot. Galvanic [33,57] and MEMS-based [21] laser scanners have been used at the active stationary nodes for the initial search of the mobile platform. The approaches mentioned above work well when one of the communicating agents is stationary and has access to high power; however, this configuration does not apply to two mobile communicating robots, especially when they have limited computing and power resources. Soysal *et al.* used quadrant photo-detector [52] to simultaneously obtain azimuthal and elevation errors, and used Kalman Filter to predict the errors and henceforward control the alignment. However, a quadrant detector is a combination of 4 photo-diodes arranged in a quadrant manner, which is also, in a way redundancy in the hardware of the system. Theoretical work involving formulation of a doubly stochastic space-time Poisson process model of light was used to derive a linear-quadratic-Gaussian (LQG) controller for active pointing in the single-directional [24] and bi-directional [23] communication settings. However, the controller needs access to the outputs from each unit of a 2-dimensional grid that is obtained by partitioning the aperture of a receiver photo-diode. A recent study [10] involves using extremum-seeking control for alignment between a laser transmitter and a photo-diode mounted on mobile-robots, where the pointing mechanism is assisted by a camera that computed the alignment error.

1.2 Overview of Contributions

The contributions of this research reside in the design and development of a novel, compact LED-based wireless communication system that maintains the LOS despite the movement of the underlying platform. The novelty of the system lies in its simple design that uses only a single photo-diode and a single LED. The signal from the photo-diode is used for both alignment control and communication. Figure 1.2 illustrates the extraction of signal strength and information from a signal from the photo-diode circuit. The output from the photo-diode is fed into a voltage comparator and a low pass averaging filter. The output of the comparator is then used as the received signal for communication, which contains the encoded information in a bit sequence format. The output of the filter gives an average intensity measurement that serves as the signal strength, which is used in alignment algorithms.

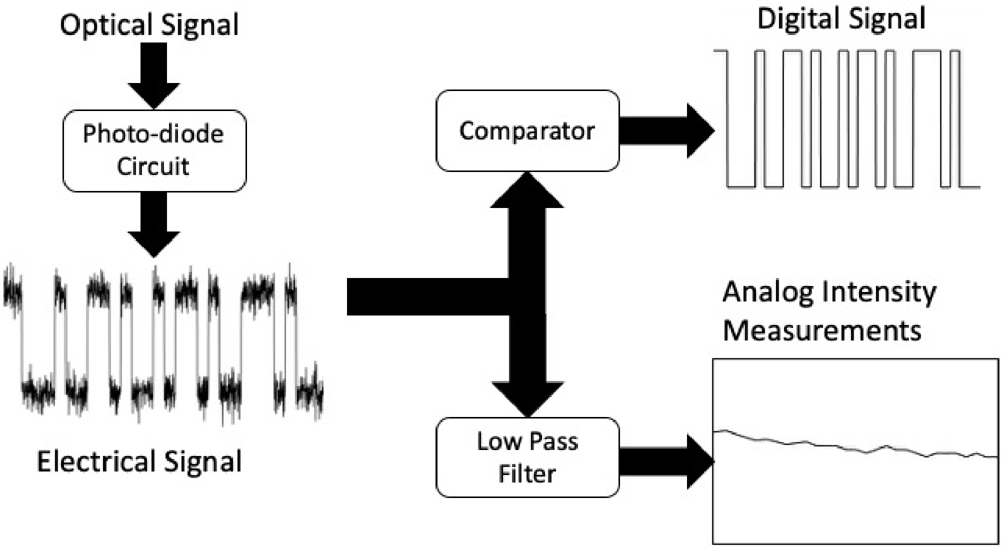


Figure 1.2: Illustration of the extraction of signal strength and information from optical signal incident on a photo-diode.

Al-Rubaiai and Tan [1,2] achieved active alignment control using single pair of LED and photo-diode, each of which was mounted on a base that could rotate independently of the underlying robotic platform. This system also used redundancy in time by taking three measurements at three different orientations and then computing a desired orientation based on the interpolation of the measurements. This alignment control approach is rudimentary and requires abrupt and large rotation of the communication device for the signal strength probing, and thus is not energy-efficient and creates unwanted mechanical vibrations. In this work, we use a principled approach for active alignment where the system of transmitter and receiver first is modeled as a dynamical system. Multiple alignment control algorithms are then developed for the system for scenarios that range from a simple one-way two-dimensional (2D) setting to a practical three-dimensional (3D) bi-directional underwater setting. Following are the details of the contributions of this work.

1.2.1 Alignment Control in the 2D Space

Firstly, we start with a two dimensional setting, where the receiver and the transmitter can only move on a plane. We propose an extended Kalman filter (EKF) for estimating the angle between the receiver orientation and the line connecting the receiver and the transmitter, which is then used to adjust the receiver orientation towards the LOS. We note that Kalman filter and EKF have been proposed in optical beam steering in the context of laser-based free space optical communication, where the laser beam is considered as a single line and thus simple geometric relationships can be used to relate the measurement to the receiver/transmitter configuration [52, 57]. For example, in their simulation study, Soysal and Efe considered a quadrant photo-detector as the measurement device, which was assumed to produce signals directly proportional to azimuth and elevation errors [52]. Yoshida and Tsujimura used a two-dimensional position-sensitive device (PSD) and the detected beam spot position was geometrically related to the relative position and orientation between the transmitter and the receiver [57]. These approaches do not apply to our setting because of the diffusive nature of LED and the use of single photo-detector in the proposed work.

We now briefly summarize our EKF-based alignment control approach. Based on a light intensity model, we first formulate an estimation problem, where the receiver estimates both its relative orientation to the transmitter and a quantity related to the overall light intensity at the receiver site. The rotating base is then instructed to move towards alignment based on the estimated relative orientation. Due to the nonlinear nature of the observation function, an EKF is adopted for the state estimation. In order to ensure convergence of the EKF, light intensity measurements taken at two consecutive steps in the scanning motion are used in each state update. The feedback control algorithm then updates the orientation bias in the angular scan motion based on the state estimate.

Preliminary versions of parts of this work were presented at the 2016 American Control Conference [45] and the 2016 IEEE International Conference on Advanced Intelligent Mechatronics [48]. More comprehensive results were published in IEEE/ASME Transactions on Mechatronics [46], where the tracking performance of the EKF approach was compared with two alternative schemes, hill-climbing and three-point averaging is, in the presence of the measurement noise. Hill-climbing is a widely used, computationally efficient algorithm for optimization that locally updates the solution in the direction of higher objective function [40]. Since better alignment between the receiver and the LED leads to higher measured light intensity, the hill-climbing algorithm simply directs the receiver to keep moving in the direction of higher light intensity. The three-point-averaging algorithm [1] computes the next orientation of the receiver based on the weighted-average of three orientations: no change, a fixed rotation to the right, and a fixed rotation to the left, where the measured intensities at these orientations are used as weights. A performance metric is designed to evaluate and compare the three algorithms in terms of tracking effectiveness, where a range of measurement noise levels is considered. For each of the noise levels, multiple runs of simulation and the corresponding experiments have been performed to assess the average performance and simultaneously alleviate the effects of stochasticity on the results. It is found that the EKF algorithm significantly outperforms the alternatives in the presence of measurement noise. Chapter 2 describes the details of this preliminary work.

1.2.2 Active Alignment Control in 3D Space

Since in a realistic underwater scenario, the robots move in a three-dimensional (3D) space, and each needs to transmit and receive. Hence, we propose a bidirectional active LOS-alignment system for mobile robots moving in a 3D scenario. In this setup, each robot is equipped with a transceiver, consisting of a photo-diode and an LED, attached to a two-degree-of-freedom (DOF) rotational system. We propose an extended Kalman filter (EKF)-based alignment control approach, where the estimates of azimuthal and elevation angles between the transceiver orientation and the LOS direction are used to adjust the transceiver orientation towards the LOS. Due to the use of a single photo-diode, only a single measurement is available at a time, which is not sufficient to estimate all the states of the system. To address this issue, we propose a circular scanning technique that ensures the system observability by taking three consecutive measurements of light intensity from non-coplanar directions.

Once the estimate of the relative angle between each agent's orientation and the LOS is known, a proportional-integral (PI) control algorithm is then used to generate the control inputs to drive the orientation of the transceiver towards the LOS. These control terms are translated into the corresponding motor commands for the rotational system. Additionally, to increase signal strength at the steady-state and to save energy, the amplitude of the circular scanning is adjusted based on the estimation covariance. Furthermore, to enable two-way alignment, the aforementioned approach is alternated between the robots, where each robot takes turns to conduct its circular scan while the other robot is paused. The efficacy of the proposed method is evaluated in both simulation and experiments, where we also implement an extremum-seeking (ES) approach [25] for comparison with the proposed approach. The ES approach is well studied for real-time optimization, and due to the unimodal nature of light-intensity function (see (3.2) and Figure 3.3 for details), it is applicable to our setup and is thus chosen as the benchmark algorithm for comparison. The simulation results demonstrate the functional limitation of the ES-based approach, as it becomes unstable at low distances. Furthermore, we perform simulations and experiments for the scenario where the robots move with a range of relative speeds. The results validate the efficacy of our approach at relatively low speeds between the robots and illustrate the challenges when the relative motion gets pronounced.

The details of this work are provided in Chapter 3. Simulation results of the proposed approach were presented at the 2017 ASME Dynamic Systems and Control Conference [51]. A more refined approach with experimental results was presented at 2018 IEEE International Conference on Robotics and Automation [49].

1.2.3 Bi-directional Alignment Approach for 2D Space

The EKF approach uses an assumption that the relative motion between the robots is small, and consequently, requires the communicating robots to take the scanning in an alternating fashion for the convergence of the estimator. In this work, we demonstrate bi-directional active beam tracking between two parties in a 2D setting. It is desirable to achieve LOS without relying on communication between the agents as the quality of communication link itself depends on the LOS. This work proposes, analyzes, and evaluates a hill-climbing based computationally efficient scheme for two agents in a plane to achieve and maintain LOS from arbitrary allowable initial configurations.

In this work we consider the optical alignment question as a two-agent cooperative-control problem. Each of the agents is assumed to have a local measurement of its output function that denotes the received optical power in terms of states x_1 and x_2 of the underlying system, with $x_i,\ i \in \{1, 2\}$, denoting the heading orientation of an agent i. The two output functions are non-conflicting; both have the global maximum at origin, which corresponds to the LOS configuration. Based on the setup, the agents do not have access to their own orientations. Furthermore, since the communication relies on the LOS, the alignment algorithm assumes that the agents do not have any communication with each other. Additionally, the agents act independently and simultaneously at each time step, which eliminates candidate solutions based on sequential actions of the agents that can greedily optimize their instantaneous output functions. We propose a computationally efficient hill-climbing based control scheme to update the heading orientations x_1 and x_2 of the agents, which only uses the current and previous measurements obtained by an agent. Furthermore, for a particular case, when the measurement functions can be approximately characterized in a Gaussian form, the proposed scheme guarantees that the headings of the agents are within a specified neighborhood of the LOS configuration within a finite number of steps.

The problem in this work can be considered as a special case of a multi-agent optimization problem in which each agent seeks to optimize their own cost (reward) function that depends on the state of the other agents. Several reported approaches based on distributed optimization rely on assumptions of the local cost functions being convex and the agents communicating their states/estimate of global state vectors as per a topology [30, 34]. Several approaches related to distributed sensing and estimation with cooperative control in multi-agent systems are reported in [11,20,26,35]. Collaborative source seeking via circular formation control is reported in [7,32], where agents share their measurements and locations with their neighbors to estimate the gradient of the underlying signal profile to steer the formation to the source of the signal. Passivity-based tools have also been studied for multi-agent synchronization and extremum-seeking problems [6,12]. However, these aforementioned works require connected network with information exchange between the agents or, in certain cases, access to the gradient of the local cost function with respect to the state. In multi-agent game-theoretic formulations, gradient play is a popular technique that converges to a Nash equilibrium for the game under mild technical assumptions [27,43]. However, these techniques require that each agent has access to the gradient of its own cost function.

The contributions of this work are three-fold. First, we propose a computationally efficient novel local control law for a class of systems that encompasses our free-space optical communication experimental setup, and only requires the information of the current and the immediately preceding reward function measurements. Second, under certain assumptions on the form of the measurement functions, we prove that, from any admissible initial values of the states, the proposed control law ensures that both agents reach a specified limiting set, which contains the global optimal, in a finite number of steps (cf. Theorem 4.2.1 for additional details). For the choice of Gaussian measurement functions derived from our setup, we obtain stronger convergence results (cf. Theorem 4.2.2). Third, the approach's effectiveness is evaluated in simulation and experiments, where the algorithm is tested with relative motion between the agents. Simulation results demonstrate the superiority of the proposed approach in terms of convergence speed, robustness to unknown disturbances, and handling large initial conditions against two alternative approaches. A 2D version of extremum-seeking (ES) control algorithm [25] is used as a benchmark approach for comparing the proposed method in this paper. The algorithm exhibits relatively slower convergence due to significant time spent in exploration by each agent. The other approach considered for comparison is the EKF based alignment approach [46]. However, the EKF formulation's assumptions of quasi-static dynamics are violated in the bi-directional setting, and consequentially, the algorithm fails to converge from arbitrarily large initial conditions.

Preliminary results of this work were presented at the 2020 IEEE International Conference on Advanced Intelligent Mechatronics [47], where the special case of Gaussian form measurement functions was considered and the studies were limited to simulation setting. The details of this work are provided in Chapter 4.

1.2.4 Bi-directional Alignment Approach for 3D Space and Experiments in Underwater Scenario

In this work, we extend the approach to bi-directional active beam tracking between two parties to the 3D scenario. Inspired by the aforementioned model-free bidirectional approach for the 2D setting, where maximizing own measurements by each agent leads to convergence to a neighborhood of the LOS, we propose a novel triangular-exploration algorithm where an agent continuously maximize its own local light intensity measurements, to achieve the LOS with the other communicating agent in the 3D setting.

The proposed algorithm moves the transceiver pointing direction in an equilateral triangular grid pattern and guarantees the pointing direction to be consistent with the local gradient direction. The method only requires the light intensity to be a unimodal function of the angle offsets from the LOS, and does not require an explicit model for implementation. Furthermore, the approach works directly for the setting of bi-directional communication. The effectiveness of the approach is first evaluated in a simulation setup of two robots by comparison with the EKF-based approach and the ES approach. Simulation results show that the proposed approach is effective in terms of its convergence speed for a wide range of relative speed and distance between the robots. The performance of the approach is further assessed against the EKF and the ES approach on an experimental setup, where the results further support the superiority of the proposed approach.

The efficacy of the approach and the overall communication system is further demonstrated in the underwater setting where communication is performed simultaneously with the alignment control. First, one-way communication and alignment is demonstrated on a setup where a human operator wirelessly controls a robot in the underwater scenario using an LED communication-based joystick, followed by demonstration of bi-directional alignment control and communication between two underwater robots. Preliminary results for the uni-directional setup were presented at 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) [50]. Chapter 5 provides the details of this work.

CHAPTER 2

EKF-BASED ALIGNMENT CONTROL IN THE 2D SPACE

In this chapter, we present a novel, compact LED-based communication system with active alignment control, in a two-dimensional (2D) setting, that maintains the LOS despite the underlying platform movement. An extended Kalman filter-based algorithm is proposed to estimate the angle between the receiver orientation and the receiver-transmitter line, which is used subsequently to adjust the receiver orientation. The algorithm uses only the measured light intensity from a single photodiode, where successive measurements are obtained via a scanning technique that also ensures the observability of the system. A simple proportional controller is designed for alignment. The effectiveness of the proposed active alignment algorithm is verified in simulation and experiments. In particular, its robustness in the presence of measurement noise is demonstrated via comparison with two alternative algorithms that are based on hill-climbing and three-point-averaging, respectively.

The organization of the chapter is as follows. In Section 2.1, the design and hardware implementation of the LED communication system is described. In Section 2.2 the model for the received light intensity is presented, followed by a state-space reformulation for the purpose of algorithm development. In Section 2.3 the estimation and tracking control algorithms are described. Simulation setup and results are presented in Section 2.4, while experimental setup and results are discussed in Section 2.5. Chapter summary is provide in Section 2.6

2.1 System Design and Implementation

An LED-based optical communication system mainly consists of two parts, the transmitter and the receiver. The transmitter converts the electrical signal into an optical signal. That signal passes through the medium and is picked up by the receiver. The receiver detects the optical signal and converts it back into an electrical signal for data processing. In addition to the transmitter and the receiver, the proposed system includes a mechanism for rotating the transmitter/receiver, to maintain communication despite the movement of the underlying robotic platform. Considering the intended applications, small footprint and low power consumption are among the major design constraints.

The role of the optical transmitter is to convert the electrical signal into light pulses. Since the signal attenuation underwater is minimum in the wavelength range of 400-500 nm [54], an off-the-shelf blue LED (Cree XR-E Series LED from Cree Inc) is chosen. which provides 30.6 lumens at 1 A and requires 3.3 volts. It comes assembled with a heat sink. A circuit is designed to modulate the LED (turning it on and off) in correlation with binary data [2]. A photodiode from Advanced Photonix (part number PDB-V107) is chosen for the receiver, and it has high quantum efficiency at 410 nm, low dark current, and fast rise time (20 ns). A 12 V reverse bias across the photodiode is used to increase the bandwidth and quantum efficiency [3]. A trans-impedance amplifier is used to convert the photodiode current signal into a voltage signal, which then goes through a filter for noise reduction.

The components of the transmitter and the receiver are placed on two printed-circuit boards (PCBs). The first PCB board is 2 inches in diameter and has two holes in the middle to attach set screw hubs for connecting to a motor shaft. The second PCB board has a rectangular shape with size of 1 inch $\times$ 2 inch, which holds the LED and the photodiode, and it is mounted perpendicularly to the first circular board by using four 90-degree header pins (see Figure 2.1).

There are 8 pins in the PCB circuits involving the power supply, the transmitted signal, and the received signal. These pins are connected by wires, which would be twisted when the PCBs are rotated. To address this problem, a slip ring (MT007 from MOFLON), an electromechanical device that allows the transmission of power and electrical signals from a stationary to a rotating structure, is adopted. A motor is used to rotate the device. We initially used a mini DC motor equipped with a shaft encoder, but later switched to a stepper motor due to the higher control precision of the latter.

2.2 Modeling

In this section, we first review a light intensity model and then formulate the state-space model for an estimation problem, where, without the loss of generality, a scenario of two robots is

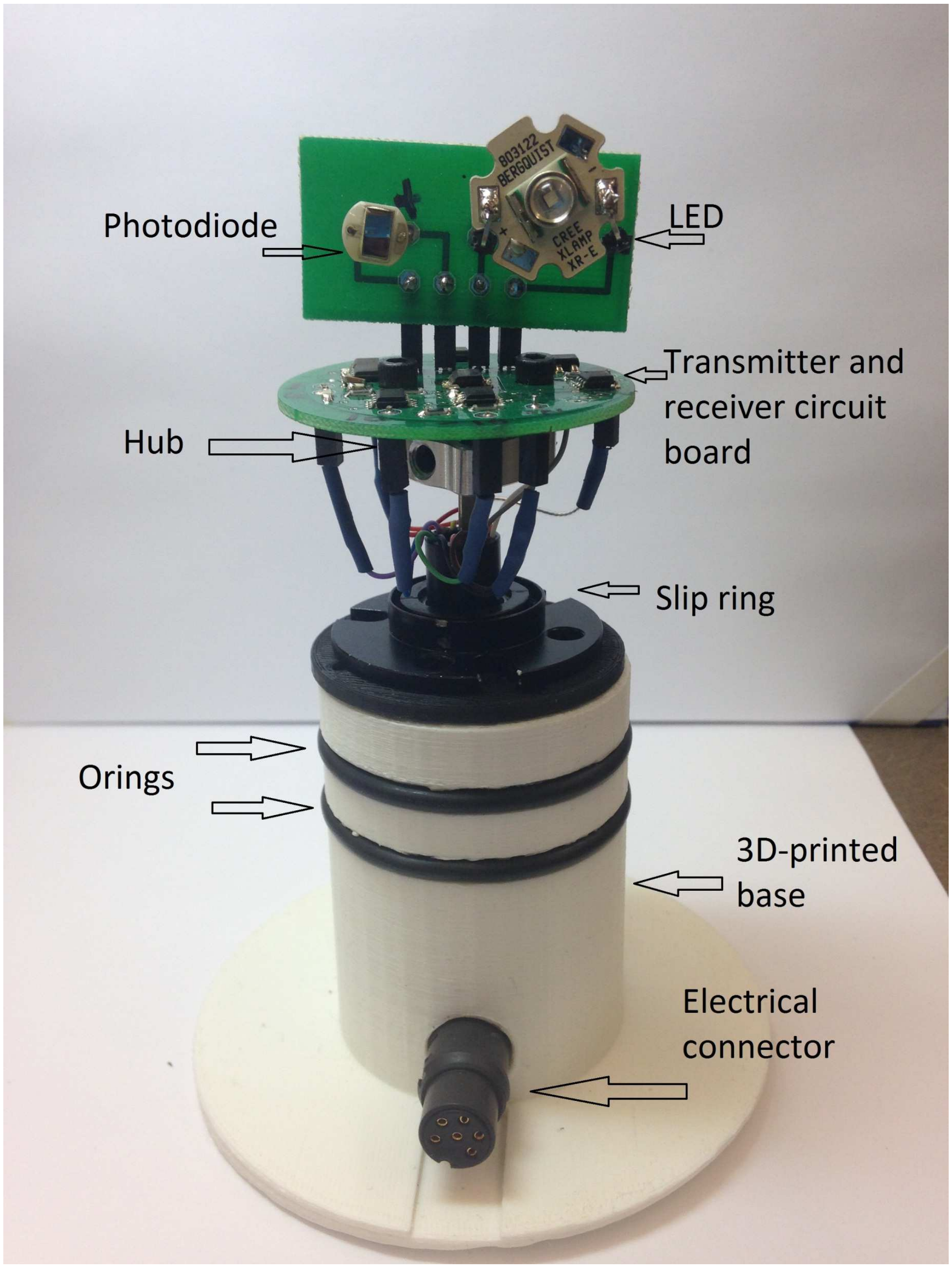


Figure 2.1: A prototype of LED optical communication module with a rotational base.

considered. In addition, in this work, we consider that the communicating parties are on the same plane.

2.2.1 Light Intensity Model

The model adopted here largely follows [15] with minor adjustments to suit the experimental prototype used in this work. The model takes into account all stages of the transmitter and receiver circuits, including LED, lens, photodiode, and amplifiers. The model mainly describes the effect of relative position and orientation between the transmitter and the receiver on the signal strength. See Figure 2.2 for an illustration of the variables of interest, including transmission angle γ, transmission distance d and the angle of incidence ϕ.

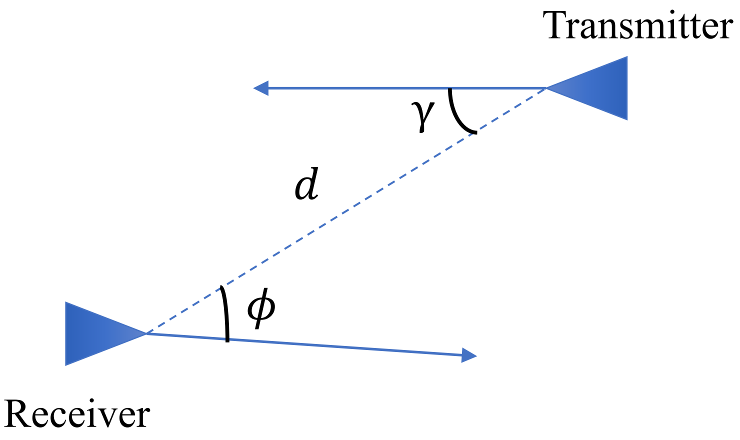


Figure 2.2: Illustration of the relative position and orientation between the transmitter and the receiver.

The transmitter LED has an angular intensity distribution which is rotationally symmetric about the LED's normal ($\gamma = 0°$). So if we know the intensity of the LED along the normal, we can compute the intensity at other points at the same radial distance based on spatial intensity curve I_γ, which represents the light intensity at a unit distance for different transmitter angles. I_γ is maximum at $\gamma = 0°$, and it rolls off as γ increases. Typically, I_γ can be obtained either directly from the LED vendor or measured experimentally.

To describe the extinction of the light signal we will adopt Beer's Law [31], which is used in understanding the attenuation in physical optics. Let c be the attenuation coefficient for the medium in which the light transmits. We assume that the coefficient is uniform across the entire length of transmission. Beer's law gives the exponential signal degradation at distance d caused by absorption:

$$A = e^{-cd} \tag{2.1}$$

By combining the effect of spherical spreading with exponential decay, we get the equation of the irradiance reaching the receiver site:

$$E_\gamma(d) = I_\gamma e^{-cd}/d^2 \tag{2.2}$$

where I_γ denotes the angular intensity distribution of transmitter (combination of the LED and the lens), which characterizes the light signal strength for different transmission angles at a unit distance. Finally, we need to consider the effect of angle of incidence ϕ, which is basically the angle made by the receiver normal with the line connecting the receiver to the transmitter. From [2], the power incident on the detector can be computed based on the signal irradiance at the detector position:

$$P_{\text{in}} = E_\gamma(d) A_0 g(\phi) \tag{2.3}$$

where A_0 denotes the detector area and $g(\phi)$ characterizes the dependence of the received light intensity on the incidence angle ϕ. The term $g(\phi)$ is setup-dependent. For the receiver used in this work, we have found the function $g(\phi)$ using Gaussian curve fitting of the normalized measurement data (Figure 2.3) collected at different orientations of the receiver. The resulting $g(\phi)$ takes the form of a bimodal Gaussian function:

$$g(\phi) = a_1 e^{-(\frac{\phi - b_1}{c_1})^2} + a_2 e^{-(\frac{\phi - b_2}{c_2})^2} \tag{2.4}$$

where a_1 = 0.6682, b_1 = 7.752, c_1 = 148.8, a_2 = 0.3340, b_2 = -13.57, c_2 = 325.8 are the curve fitting parameters. The parameters b_1 and b_2 are relatively close to each other (over the range of

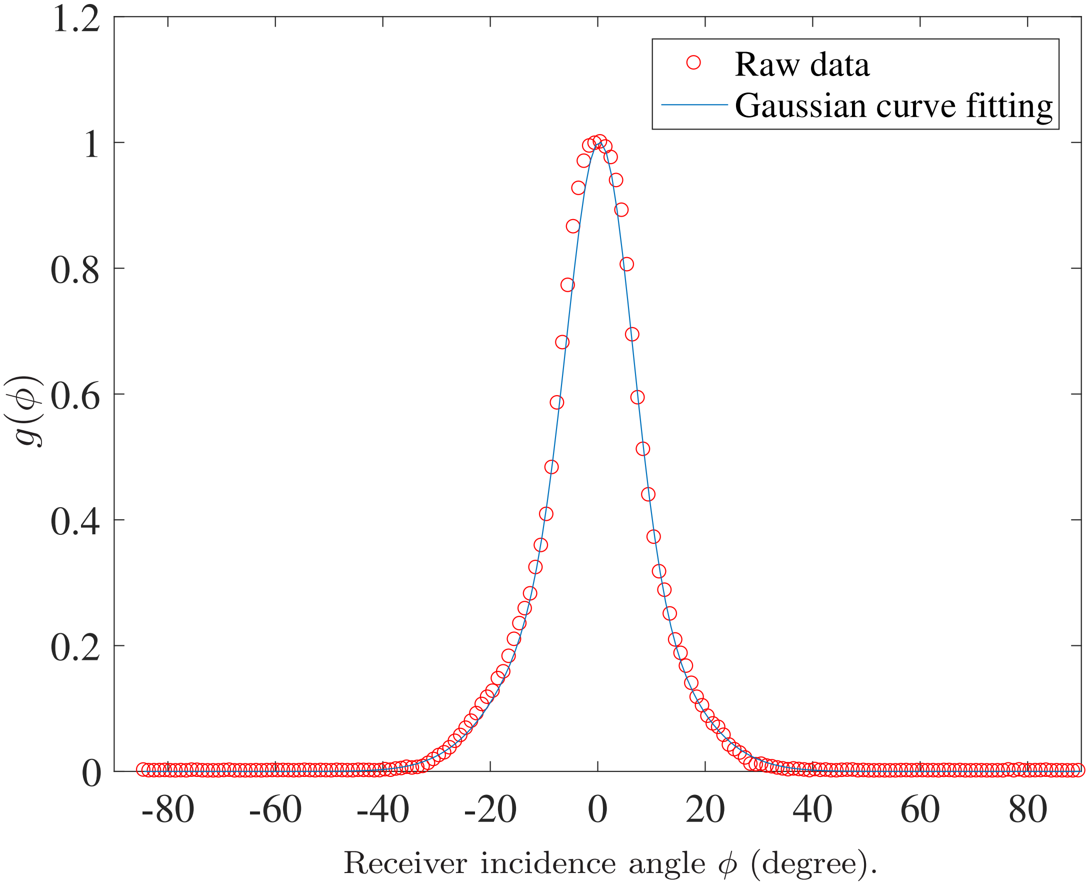


Figure 2.3: Gaussian curve fitting for the function g for the photodiode used in this work.

$[-180°, 180°]$), so the resulting sum of the two modes has a single peak, as shown in Figure 2.3. The curve fitting could be done using a single Gaussian mode but having one extra Gaussian mode gives significantly better fitting.

As the light arrives on the receiver photodiode, the photodiode produces a current, which gets filtered and amplified, to be processed by an analog-digital converter. After all the stages, the full signal strength model can be summarized as

$$V_d = C_p I_\gamma e^{-cd} g(\phi)/d^2 \tag{2.5}$$

where V_d is the voltage signal and C_p is a constant of proportionality, which depends on the area of receiver photo-diode and various parameters associated with the filter and amplifier circuits.

2.2.2 State-space Problem Formulation

From Figure 2.2 and Eq. (2.5), we can see that there are three independent variables, γ, d and ϕ, that characterize the received light intensity. One could take these three variables as the states to be estimated by the system, and then try to drive them towards their desired values through control, if that is possible. However, often times the underlying robotic platforms are engaged in other tasks and may not constrain or modify their motions to accommodate communication. For example, it may not be possible to change distance d in a desirable way for communication since that would involve the movement of the robots. What is much more practical is to control the receiver angle ϕ, since it is a completely local decision due to the independent rotation base for the transceiver. In a two-way communication setting, since the transmitter and the receiver on each robot are pointing in the same direction, adjusting ϕ to zero on each robot automatically aligns each transmitter with the line connecting two robots. In light of this discussion, we can combine terms involving γ and d in a single variable and define the state variables as

$$\mathbf{x} \triangleq \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \triangleq \begin{bmatrix} C_p I_\gamma e^{-cd}/d^2 \\ \phi \end{bmatrix} \tag{2.6}$$

The value of x_1 is dependent on the distance and the transmission angle. In a typical scenario, the receiver does not have information about how the transmitter and its underlying robotic platform move. So in our case, we will assume that the relative dynamics between the two communicating robots is slow enough (quasi-static) that it can be captured with a Gaussian process. In particular, the dynamics of the states defined in (2.6) can be represented in the discrete-time domain as

$$\mathbf{x}_k \triangleq \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} = \begin{bmatrix} x_{1,k-1} + w_{1,k-1} \\ x_{2,k-1} + u_k + w_{2,k-1} \end{bmatrix} \tag{2.7}$$

where k is the time index, and $w_{1,k}$ and $w_{2,k}$ are the process noises, assumed to be independent, white, Gaussian noises. These noise terms, to some extent, account for the slow dynamics of x_1 and x_2, which are not modeled explicitly. The term u_k is the control input through which the receiver angle is changed.

The kth measurement $V_{d,k}$ can be expressed in terms of the state variables, where an additive white Gaussian noise v_k, assumed to be independent from the process noises, is included:

$$V_{d,k} = x_{1,k} g(x_{2,k}) + v_k \tag{2.8}$$

Given the measurement, the goal is to estimate $x_{1,k}$ and $x_{2,k}$, based on which the control u_k is designed to drive x_2 towards 0°, which is the orientation with the maximum light intensity.

2.3 Estimation and Alignment Algorithms

Given that the measurement model (2.8) is nonlinear, a discrete time extended Kalman filter (EKF) [39] is explored for solving the estimation problem. From the (linear) state equation (2.7), the A and B matrices are:

$$\begin{cases} A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{cases} \tag{2.9}$$

So that the system dynamics can be written as:

$$\mathbf{x}_k = A\mathbf{x}_{k-1} + Bu_{k-1} + \mathbf{w}_{k-1} \tag{2.10}$$

with $\mathbf{w}_k = \left[w_{1,k}, w_{2,k}\right]^T$. Given that the output function in (2.8) is nonlinear, denoting the system's linearized output matrix at kth time instant as $C_k \triangleq C(\mathbf{x}_k)$, one can express the observability matrix at that time instant as [39]

$$O_k = \begin{bmatrix} C_k \\ C_{k+1}A \end{bmatrix} = \begin{bmatrix} C_k \\ C_{k+1} \end{bmatrix} \tag{2.11}$$

If the observability matrix O_k has a full rank of 2, the state estimation error under the EKF will be exponentially bounded in mean square and bounded with probability one under proper conditions [39]. A sufficient condition for O_k to be full rank is to make C_k a rank 2 matrix, which is only possible with at least two independent measurements of the light intensity. One could use two receivers with different orientations to address this problem, but that would increase the complexity and cost of the system. Instead, we introduce a scanning technique, where the motor of the rotating base is commanded to oscillate around a mean position. This mean position, which is what the control input modulates, is considered to be the state variable x_2 from here on.

Figure 2.4 provides an outline of the proposed method. At each iteration, the states x_1 and x_2 are updated according to the system dynamics. The scanning term is added to account for the actual orientation of the receiver. The light intensity measured by the receiver is used by the EKF to update the state estimates. Next, the estimate $\hat{x}_2$ is used to compute the control term. We note that the focus of this work is on the use of nonlinear estimation and basic feedback concepts to enable active alignment between the receiver and the LED. Therefore, exploration of advanced controllers is beyond the scope of this work; for simplicity of implementation and presentation, a proportional controller is adopted. The final command sent to the motor is the sum of control term and the difference between the last two consecutive scanning terms.

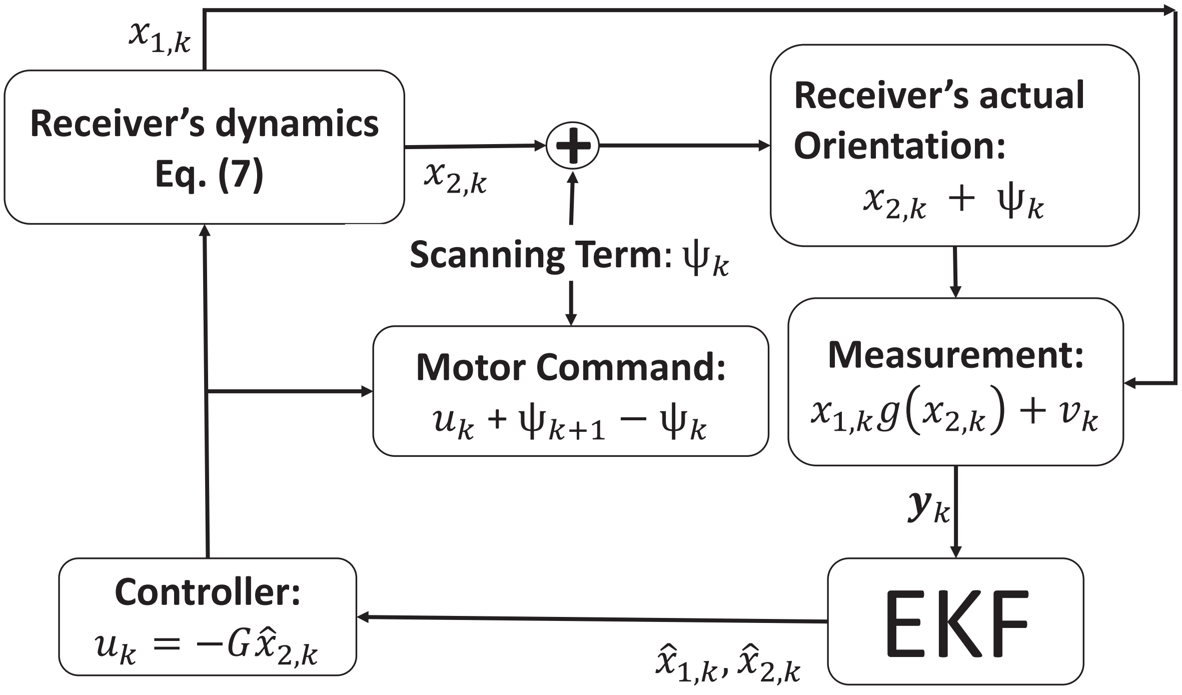

$x_{1,k}$
Receiver's dynamics Eq. (7)
$x_{2,k}$
+
Receiver's actual Orientation: $x_{2,k} + \psi_k$
Scanning Term: ψ_k
Motor Command: $u_k + \psi_{k+1} - \psi_k$
Measurement: $x_{1,k}g(x_{2,k}) + v_k$
y_k
Controller: $u_k = -G\hat{x}_{2,k}$
$\hat{x}_{1,k}, \hat{x}_{2,k}$
EKF

Figure 2.4: Block diagram illustrating the proposed method.

Figure 2.5 illustrates the scanning technique. The receiver oscillates through a defined array of angles Ψ = { ψ_1, ψ_2, ψ_3 ... ψ_n }, which contains predefined angles used for scanning. In our case

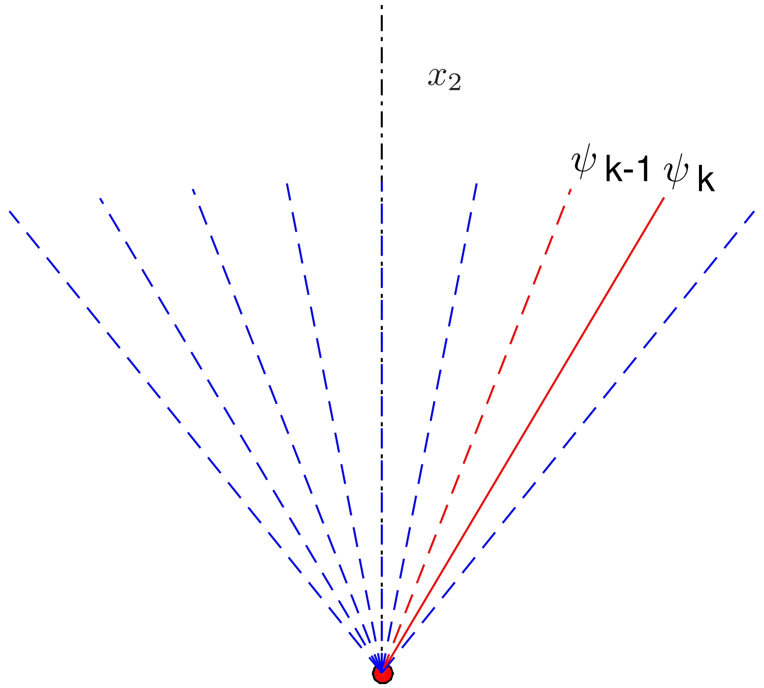


Figure 2.5: Illustration of the receiver scanning sequence, with mean x_2 and last two angles of scanning ψ_k and ψ_{k-1}.

$\Psi = \{-2°, -4°, -6°, -8°, -10°, -8°, -6°, -4°, -2°, 0°, 2°, 4°, 6°, 8°, 10°, 8°, 6°, 4°, 2°, 0° \}$. In each iteration, one ψ_k is chosen from this array sequentially. The measurement is taken at each ψ_k, and the last two measurements at ψ_k and ψ_{k-1} form our output vector **y**:

$$\mathbf{y}_k = \begin{bmatrix} x_{1,k} g(x_{2,k} + \psi_k) + v_k \\ x_{1,k-1} g(x_{2,k-1} + \psi_{k-1}) + v_{k-1} \end{bmatrix} \tag{2.12}$$

Using the dynamics equation (2.7) and the measurement equation (2.12), an EKF can be implemented. The complete algorithm is explained as follows.

There are three covariance matrices, namely, P, Q and R, associated with an EKF. P is the conditional error covariance matrix and P^f represents the forecast of the covariance matrix. P^f needs to be initialized as a positive definite matrix. The initial value of the estimate of x_2 can be taken as 0°. The initial value of the estimate of x_1 depends on the maximum possible value of intensity. A good choice of the initial estimate $\hat{x}_1$ would be from 1/3 to 2/3 of the maximum intensity value. Q is the process noise covariance matrix, and R is the measurement noise covariance matrix. At step k,

1. Prediction phase: Both state estimates ($\hat{x}$) and error co-variance matrix (P^f) are predicted:

$$\hat{\mathbf{x}}_k^f \triangleq \begin{bmatrix} \hat{x}_{1,k}^f \\ \hat{x}_{2,k}^f \end{bmatrix} = \begin{bmatrix} \hat{x}_{1,k-1} \\ \hat{x}_{2,k-1} + u_{k-1} \end{bmatrix} \tag{2.13}$$

$$P_k^f = AP_{k-1}A^T + Q \tag{2.14}$$

where $\hat{x}_{m,k}^f$ denotes the estimate of the mth state at kth interval and the superscript f stands for 'forecast'.

2. Estimated output: From (2.12), the estimated output can be written as

$$\hat{\mathbf{y}}_k \triangleq \begin{bmatrix} \hat{y}_{1,k} \\ \hat{y}_{2,k} \end{bmatrix} \triangleq \begin{bmatrix} \hat{x}_{1,k}^f g(\hat{x}_{2,k}^f + \psi_k) \\ \hat{x}_{1,k-1} g(\hat{x}_{2,k-1} + \psi_{k-1}) \end{bmatrix} \tag{2.15}$$

With (2.13), one can write

$$\begin{aligned} \hat{\mathbf{y}}_k &= h(\hat{x}_{1,k}^f, \hat{x}_{2,k}^f) \\ &\triangleq \begin{bmatrix} \hat{x}_{1,k}^f g(\hat{x}_{2,k}^f + \psi_k) \\ \hat{x}_{1,k}^f g(\hat{x}_{2,k}^f - u_{k-1} + \psi_{k-1}) \end{bmatrix} \end{aligned} \tag{2.16}$$

The term u_{k-1} would be expressed in terms of the state variables later. Now the linearized observation matrix C_k can be computed as:

$$C_k = \frac{\partial h(\hat{x}_{1,k}^f, \hat{x}_{2,k}^f)}{\partial \hat{\mathbf{x}}_k^f} = \begin{bmatrix} C_{k,1,1} & C_{k,1,2} \\ C_{k,2,1} & C_{k,2,2} \end{bmatrix} \tag{2.17}$$

where,

$$\begin{aligned} C_{k,1,1} &= g(\hat{x}_{2,k}^f + \psi_k) \\ C_{k,1,2} &= \hat{x}_{1,k}^f g'(\hat{x}_{2,k}^f + \psi_k) \\ C_{k,2,1} &= g(\hat{x}_{2,k}^f - u_{k-1} + \psi_{k-1}) \\ C_{k,2,2} &= \hat{x}_{1,k}^f g'(\hat{x}_{2,k}^f - u_{k-1} + \psi_{k-1}) \end{aligned}$$

with $g'(\cdot)$ being the derivative of $g(\cdot)$with respect to its argument.

3. Update/analysis phase:

$$K_k = P_k^f C_k^T (C_k P_k^f C_k^T + R)^{-1} \tag{2.18}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^f + K_k(\mathbf{y}_k - \hat{\mathbf{y}}_k) \tag{2.19}$$

$$P_k = (I - K_k C_k) P_k^f \tag{2.20}$$

It is to be noted that P_k and $\hat{\mathbf{x}}_k$ without any superscripts denote the updated values after the analysis phase.

4. Finally, the control is computed as

$$u_k = -G\hat{x}_{2,k} \tag{2.21}$$

where G is a positive gain, which, is motivated by the goal of driving the mean of scan x_2 to zero. The final rotation angle sent to the motor is $u_k + \psi_{k+1} - \psi_k$, which will be used to update the receiver angle at time $k + 1$.

Since the algorithm is based on EKF, the convergence depends mainly on two factors: the full rank condition of the observability matrix O_k of the linearized system (recall Eq. (2.11), and the initial conditions of the state estimates, which were already discussed earlier. Since the full rank condition of O_k is ensured by the non-singularity of the output matrix C_k, we consider the determinant of C_k

$$\begin{aligned} |C_k| = {} & \hat{x}_{1,k}^f g(\hat{x}_{2,k}^f + \psi_k) g'(\hat{x}_{2,k}^f - u_{k-1} + \psi_{k-1}) \\ & -\hat{x}_{1,k}^f g(\hat{x}_{2,k}^f - u_{k-1} + \psi_{k-1}) g'(\hat{x}_{2,k}^f + \psi_k) \end{aligned}$$

Using (2.13) and (2.21), we obtain

$$u_{k-1} = -\frac{G\hat{x}_{2,k}^f}{1 - G}$$

which implies

$$\hat{x}_{2,k}^f - u_{k-1} = \frac{\hat{x}_{2,k}^f}{1 - G}$$

and thus

$$|C_k| = \hat{x}^f_{1,k} g(\hat{x}^f_{2,k} + \psi_k) g'(\frac{\hat{x}^f_{2,k}}{1-G} + \psi_{k-1})$$
$$-\hat{x}^f_{1,k} g(\frac{\hat{x}^f_{2,k}}{1-G} + \psi_{k-1}) g'(\hat{x}^f_{2,k} + \psi_k)$$

Since x_1 represents the light intensity at the receiver site (which is in general different from the measured intensity by the receiver), it is always positive – if x_1 were zero, there would not be any measured signal even if the receiver is perfectly pointing at the transmitter and the algorithm would be stopped. Therefore, it is reasonable to assume $\hat{x}^f_{1,k} > 0$. So the only possibility for $|C_k|= 0$ is then

$$g(\hat{x}^f_{2,k} + \psi_k) g'(\frac{\hat{x}^f_{2,k}}{1-G} + \psi_{k-1})$$
$$-g(\frac{\hat{x}^f_{2,k}}{1-G} + \psi_{k-1}) g'(\hat{x}^f_{2,k} + \psi_k) = 0$$

or

$$\frac{g'(\hat{x}^f_{2,k} + \psi_k)}{g(\hat{x}^f_{2,k} + \psi_k)} = \frac{g'(\frac{\hat{x}^f_{2,k}}{1-G} + \psi_{k-1})}{g(\frac{\hat{x}^f_{2,k}}{1-G} + \psi_{k-1})} \tag{2.22}$$

One can show that the function $\mathbf{G} = \frac{g'(\cdot)}{g(\cdot)}$ is monotonously decreasing in our domain of interest: $(-180°, 180°)$. Hence (2.22) would be true if and only if

$$\hat{x}^f_{2,k} + \psi_k = \frac{\hat{x}^f_{2,k}}{1-G} + \psi_{k-1} \tag{2.23}$$

which implies

$$\hat{x}^f_{2,k} = \frac{(1-G)}{G}(\psi_k - \psi_{k-1}) \tag{2.24}$$

Since

$$|\hat{x}^f_{2,k}| < 180 \text{ and } |\psi_k - \psi_{k-1}| = 2$$

a sufficient condition for guaranteeing that (2.24) does not hold and thus C_k is non-singular, is

$$\left|\frac{1-G}{G}\right| < 90, \text{ or } G > \frac{1}{91}$$

Here, we get a very relaxed criterion on G. So for our simulation and experiments, we used $G = 0.5$.

2.4 Simulation Results

In this section, we verify the effectiveness of our algorithm through Matlab simulation. In addition, we introduce two alternative algorithms, followed by a comparison of EKF with the two algorithms.

To further explore the effect of unmodeled system dynamics (which were ignored in the algorithm development, considering that the receiver typically would not have access to the motion information of the transmitter), we have included some arbitrary dynamics for the system in the simulation. Specifically, the system state evolves according to the following,

$$\begin{cases} x_{1,k} = x_{1,k-1} + w_{1,k-1} \\ x_{2,k} = x_{2,k-1} + \beta + u_{k-1} + w_{2,k-1} \end{cases} \tag{2.25}$$

where β is an unknown constant disturbance. The term β corresponds to the relative angular movement between the transmitter and the receiver, and it simulates the scenario where the transmitter revolves around the receiver while shining directly at it. Based on the model and the algorithms described earlier, the simulation is conducted with parameters listed in Table 2.1. Note that we have used different Q and R values for EKF, than the system's noise covariance matrices, as it is shown earlier [45] that using scaled-up noise covariance matrices for EKF implementation gives an improvement in convergence performance.

Figure 2.6 shows the results obtained from a simulation run with $\beta = 1.2$. From Figure 2.6, it can be seen that the estimated states converge to the neighborhood of the original states in about 2 seconds and remain there throughout the run. We note that the estimated value for the state x_2 is slightly lower than the actual. Most likely this can be attributed to the positive bias term β in the system dynamics, which constantly produces a shift of receiver orientation in the positive direction. Note that the oscillations in the measurements are attributed to the scanning motion of the receiver.

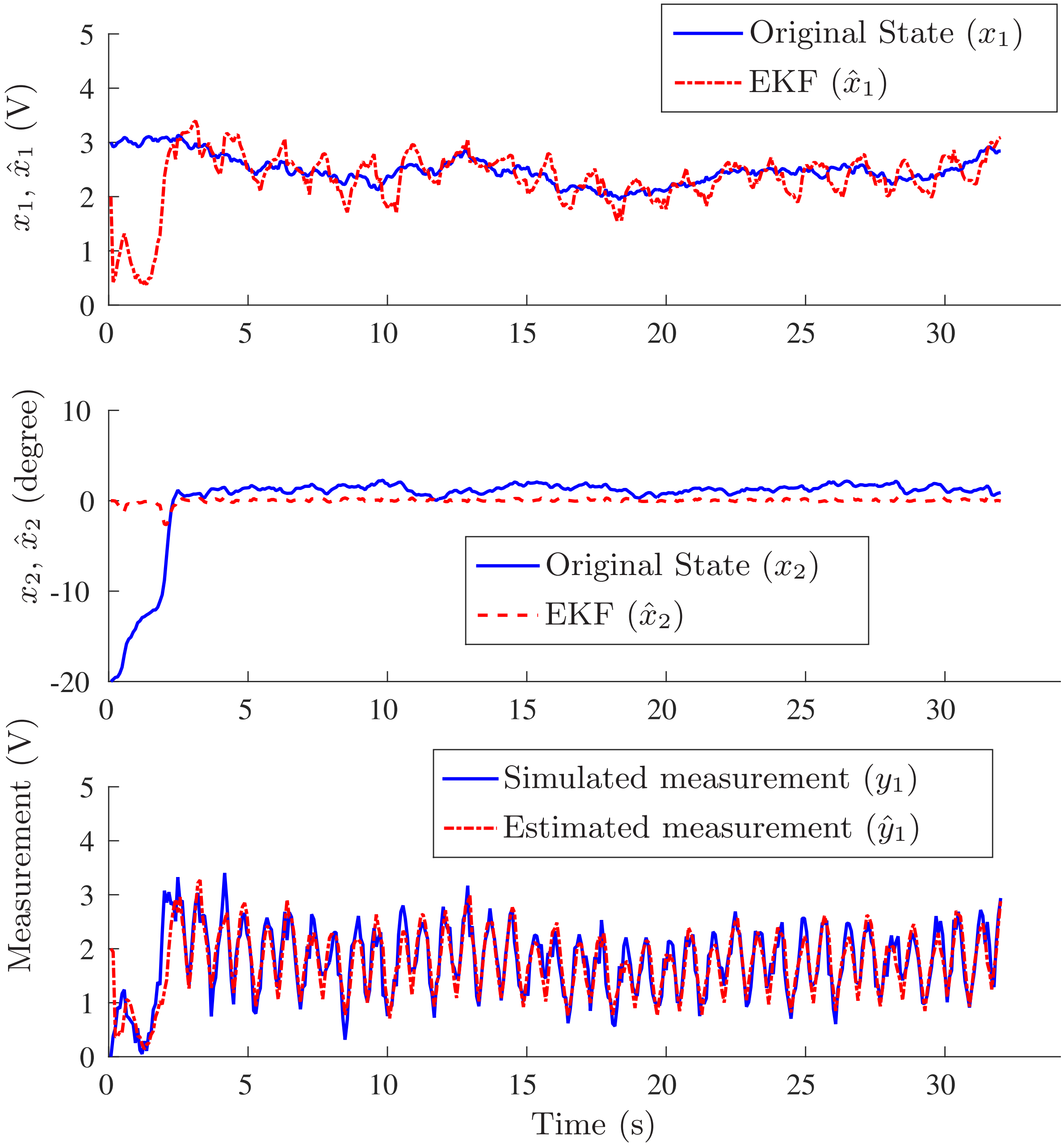


Figure 2.6: Simulation results of EKF when the x_2 dynamics contains an unknown constant disturbance $\beta = 1.2°/\text{s}$.

Table 2.1: Parameters associated with EKF implementation in the simulation.

Parameter	Description	Value
$\hat{\mathbf{x}}_\mathbf{0}$	Initial state estimate	$[2, 0]^T$
P_0^f	Initial error covariance matrix	$\begin{bmatrix} 100 & 0 \\ 0 & 1000 \end{bmatrix}$
Q_{sys}	System's process noise covariance matrix	$\begin{bmatrix} 0.0025 & 0 \\ 0 & 0.01 \end{bmatrix}$
Q	EKF's process noise covariance matrix	$\begin{bmatrix} 0.25 & 0 \\ 0 & 1 \end{bmatrix}$
R_{sys}	System's measurement noise covariance matrix	$\begin{bmatrix} 0.04 & 0 \\ 0 & 0.04 \end{bmatrix}$
R	EKF's measurement noise covariance matrix	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
G	Proportional controller gain	0.5

And we note that the actual mean (state x_2) converges to the neighborhood of zero in about 2 - 3 seconds. This corresponds to the alignment of the mean direction of the receiver to the direction that faces the transmitter.

To explore the limit of algorithm's assumption on quasi-static dynamics, the constant disturbance term β is increased to an extent where the tracking fails. Figure 2.7 shows the states when the EKF algorithm stops working and the angle of incidence (x_2) starts going unbounded. This corresponds to the receiver's direction moving away from the transmitter-facing direction.

Next, we compare the performance of the EKF-based algorithm with two alternative algorithms: hill-climbing and three-point-averaging. In particular, we explore the performance of the algorithms in the presence of measurement noises. In hill-climbing, the receiver starts with an orientation, measures the light intensity and rotates by angle $\kappa = 2°$ in either the clockwise or counterclockwise direction. It then measures the new light intensity. If the latter is higher than the previous value, it will rotate by κ again in the same direction; otherwise it will rotate in the opposite direction.

For the three point-averaging algorithm [2], the receiver performs a clockwise rotation and then a counter-clockwise rotation by an angle κ from the original location. Meanwhile, it takes the light intensity measurements at each step (V_1, V_2, V_3), where V_1, V_2, V_3 represent the voltages at the

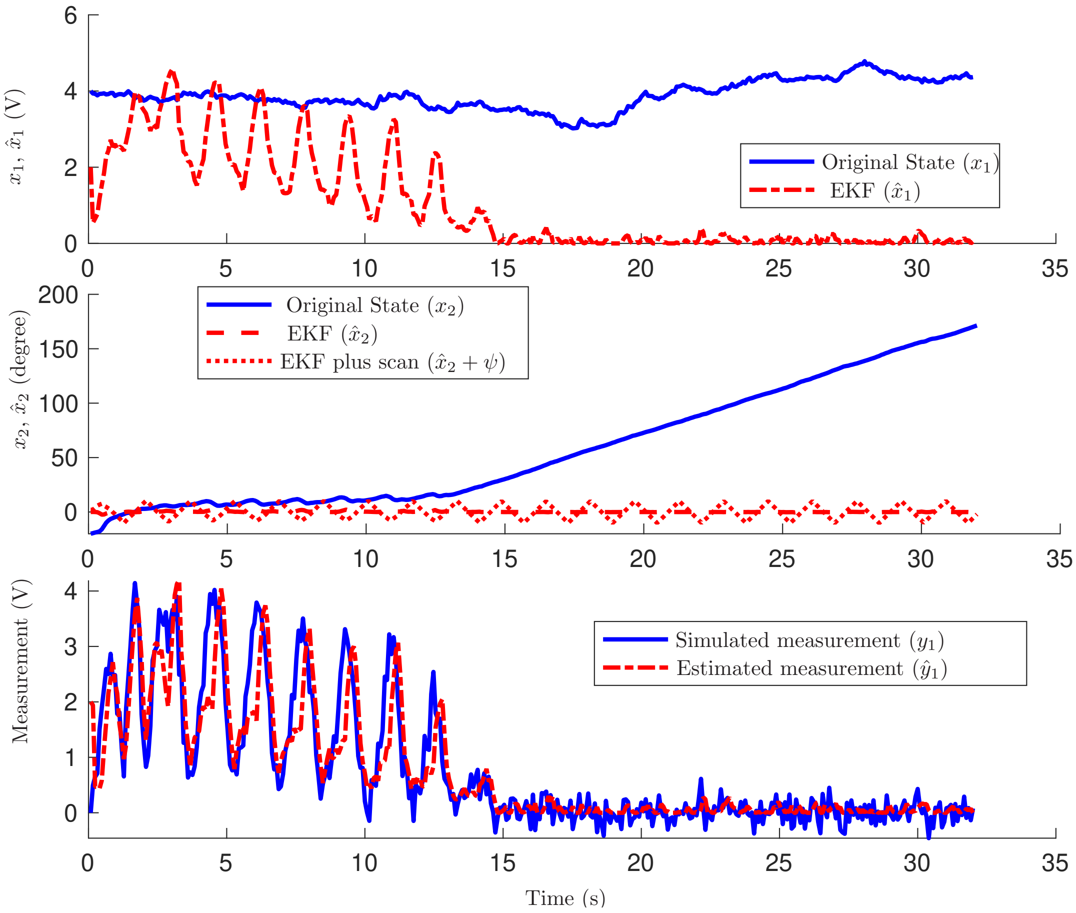


Figure 2.7: Simulation results of EKF method's failure when the unknown constant rate disturbance β is increased to 8°/s.

clockwise rotation, counter-clockwise rotation, and original location ($\kappa = 0$), respectively. The new turning angle κ_p of the rotating base is calculated by taking a weighted average of these signals at three steps:

$$\kappa_p = \frac{\kappa V_1 - \kappa V_2}{V_1 + V_2 + V_3} \tag{2.26}$$

The algorithm is implemented in Matlab for simulation, with turning angle $\kappa = 2^\circ$. To quantify the alignment control performance, two metrics are considered based on the angle of incidence (ϕ) and the clean light intensity measurement (light intensity value uncorrupted by noise v_k, defined as 'clean measurement'), respectively. Note that the algorithms use the noise-corrupted measurement for alignment control, and the clean measurement is used only for performance evaluation. We define the 'tracking percentage' as the fraction of time when the system is in the tracking zone, where the latter could be determined based on either the angle of incidence or the clean measurement. In particular, we consider a threshold of 15° for the angle of incidence ϕ : if ϕ is outside the range $[-15, 15]^{\circ}$, it would be considered out of the tracking zone. From Figure 2.3, we can see that this angular threshold corresponds to about 20% of the maximum level of the intensity, which we will use as a threshold for the 'clean measurement' for determining whether the receiver orientation is in the tracking zone. For a maximum intensity of 3 V, the corresponding intensity threshold for tracking is 0.6 V.

For the comparison of alignment control performance, each algorithm is run with a series of noise levels for the measurement. As the noise level increases, the stochasticity in the tracking percentage increases. Hence in simulation, for each noise level, results of 1000 runs have been combined. In each run, the tracking percentage is computed and then the average over the 1000 runs is obtained to get an estimate of the expected value. The standard deviation in each case is also computed to capture the variation among the runs. Figure 2.8 shows the plots for the mean of each algorithm with standard deviation as error bars. For clarity, the standard deviation in error bars is scaled down by 5 times. For each of the algorithm, the tracking percentage is computed by the two methods (based on the angle of incidence (ϕ), legend '-angular', and the light intensity measurement, legend 'measure') mentioned earlier.

The purpose of using two methods to compute tracking is to show that there is a high amount of correlation between the tracking percentages computed by these two methods. So that in the case of experiments, even if the angular data is not available, we can confidently rely on the tracking percentage generated by the intensity measurement data. From Figure 2.8, one can easily see that the tracking performance of the EKF-based algorithm is much better than that of the other two. The tracking remains 100% even at the noise level of 1.0 and it decreases gradually after that, which gives a good range of operation. The other two algorithms perform well under low noise levels, but their performance degrades faster than the EKF at higher noise levels. It can also be observed

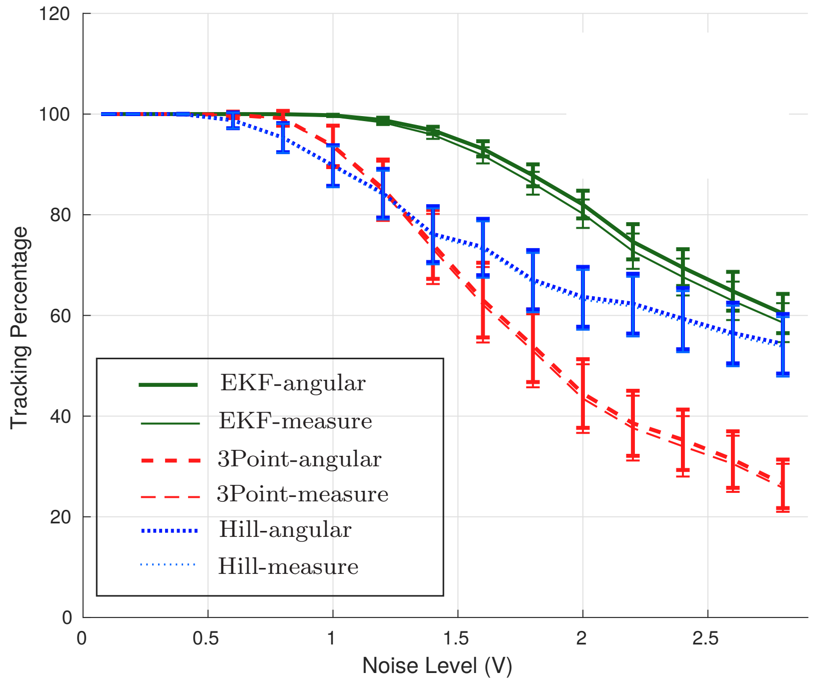


Figure 2.8: Simulation results on comparison of alignment control performance for the three methods, for different levels of measurement noise, when system states are evolved according to Eq.(2.25) with $\beta = 1.2$. Vertical bars denote the down-scaled standard deviations at each point. "Hill" represents the hill-climbing algorithm, and "3Point" represents the three-point-averaging algorithm.

that the standard deviation under the EKF-based algorithm is also much lower than the other two algorithms, which further proves its reliability under higher noise levels.

2.5 Experimental Results

In this section, we verify the efficacy of our algorithm by implementing cases similar to the simulation on an experimental setup. While the LED communication hardware design allows us to use both the receiver and the transmitter in the same module at once, so that two robots, each equipped with such a module, can communicate with each other both-ways, in this particular work we are focused on implementation of tracking algorithm on the receiver. Therefore, a separate light source is used as a transmitter. Figure 2.9 shows such a transmitter-receiver pair used in the experiments. The transmitter is mounted on a mobile robot to facilitate relative motion with respect to the receiver. It is to be noted that the receiver in Figure 2.9 is a modified version of the device shown in Figure 2.1. Here, as mentioned earlier, a stepper motor is used instead of a DC motor. The stepper motor has a precise control over the angular position. For the real-time onboard implementation of all computations, Intel Edison® mini-computer board is used. It is equipped with 500 MHz Atom 2-Core CPU and 1 GB of LPDDR3 RAM. The hardware specifications are sufficient for the real-time computation required for our algorithm, and each EKF iteration takes about 50 ms to complete.

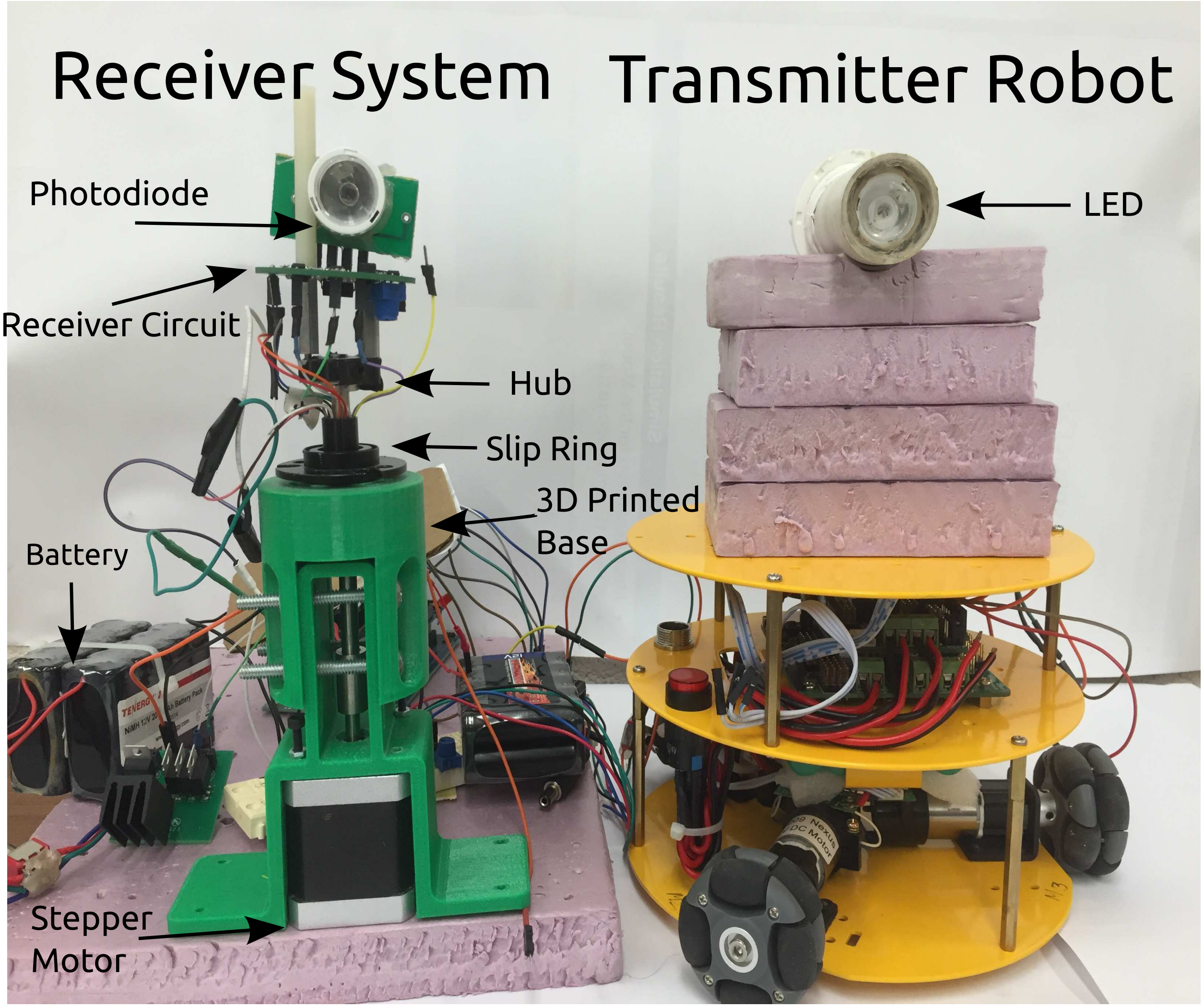


Figure 2.9: Complete setup: Receiver(left) on a rotation base and transmitter LED on a mobile robot (right).

A mobile robot equipped with the transmitter revolves around the static receiver at a distance of 1.25 m while facing the receiver (Figure 2.10). Here the transmitter robot is hard-coded to follow the circular path centered around the receiver, which not only ensures the distance d to be constant but also enables the transmitter to focus light on the receiver throughout the run ($\gamma \approx 0^o$). The robot's speed is fixed in such a way so that it revolves around the receiver at $1^\circ/\mathrm{s}$. On the receiver's end, an averaging filter is implemented on the on-board measurement of the light intensity. Currently, our experiments are conducted in the air so the noise is relatively small and can be removed by averaging. The averaged output of the filter is termed as 'clean measurement' for the experiments. Moreover, to implement a range of noises, an extra artificial Gaussian noise term is added to the clean measurement.

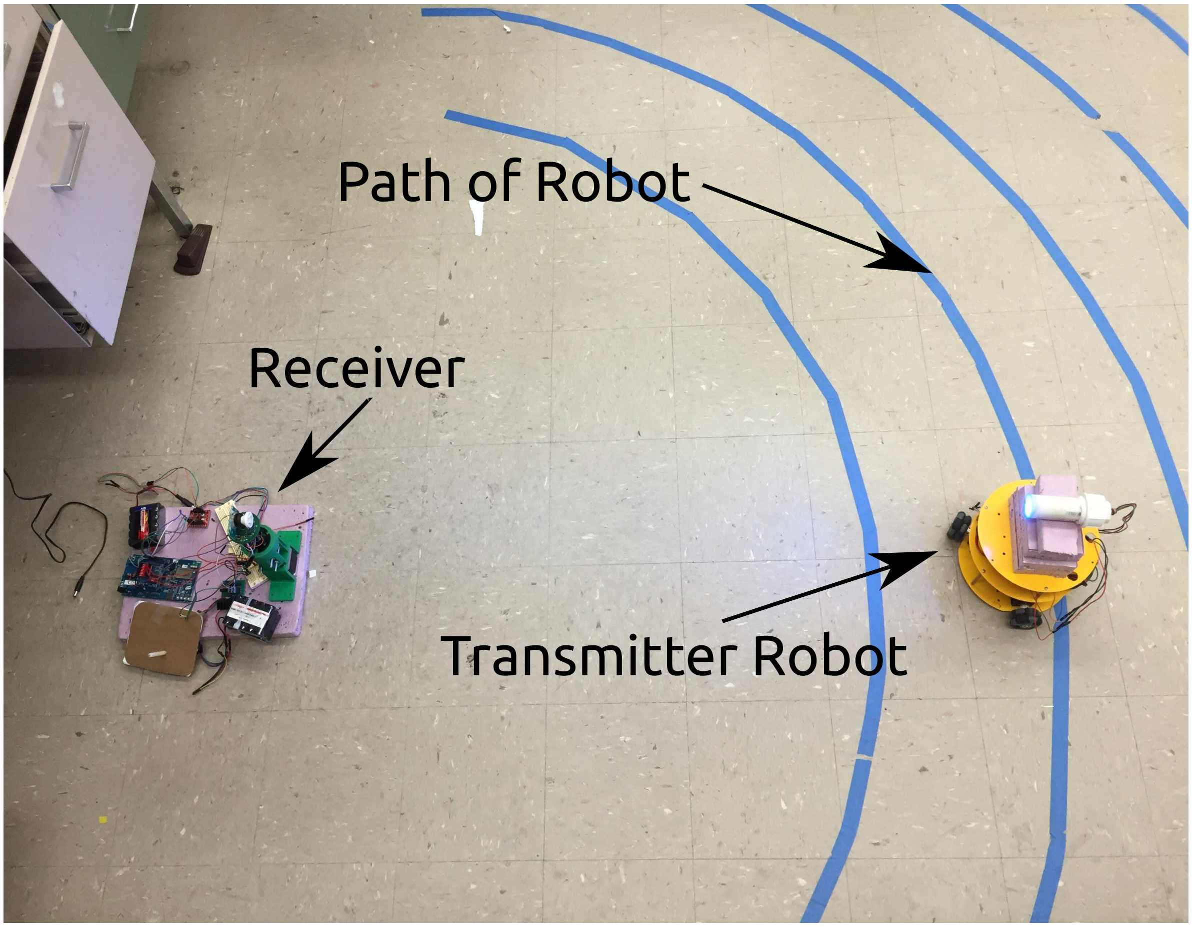


Figure 2.10: Experimental setup: Transmitter robot moving around static receiver, following the marker lines.

Figure 2.11 shows the evolution of state estimates and measurement output for a particular run with the additional, artificial noise level of 1.0 when the EKF-based algorithm is implemented.

Here we can see that the estimate of the mean of the scan ($\hat{x}_2$) goes to a bounded neighborhood of zero as well. It is to be noted that we do not have access to the original states of the system, hence only the estimates are plotted. However, while running the experiment, we have visually observed that after a few iterations, the mean angular position of the receiver-scan starts aligning itself with the line connecting the transmitter to the receiver, which implies that the real x_2 also converges to the neighborhood of 0°. Moreover, the estimate $\hat{x}_1$ stays within a relatively narrow range (according to our design of experiment), which should be close to the actual x_1. Hence, according to these observations, the estimates remain within a close range around the original states.

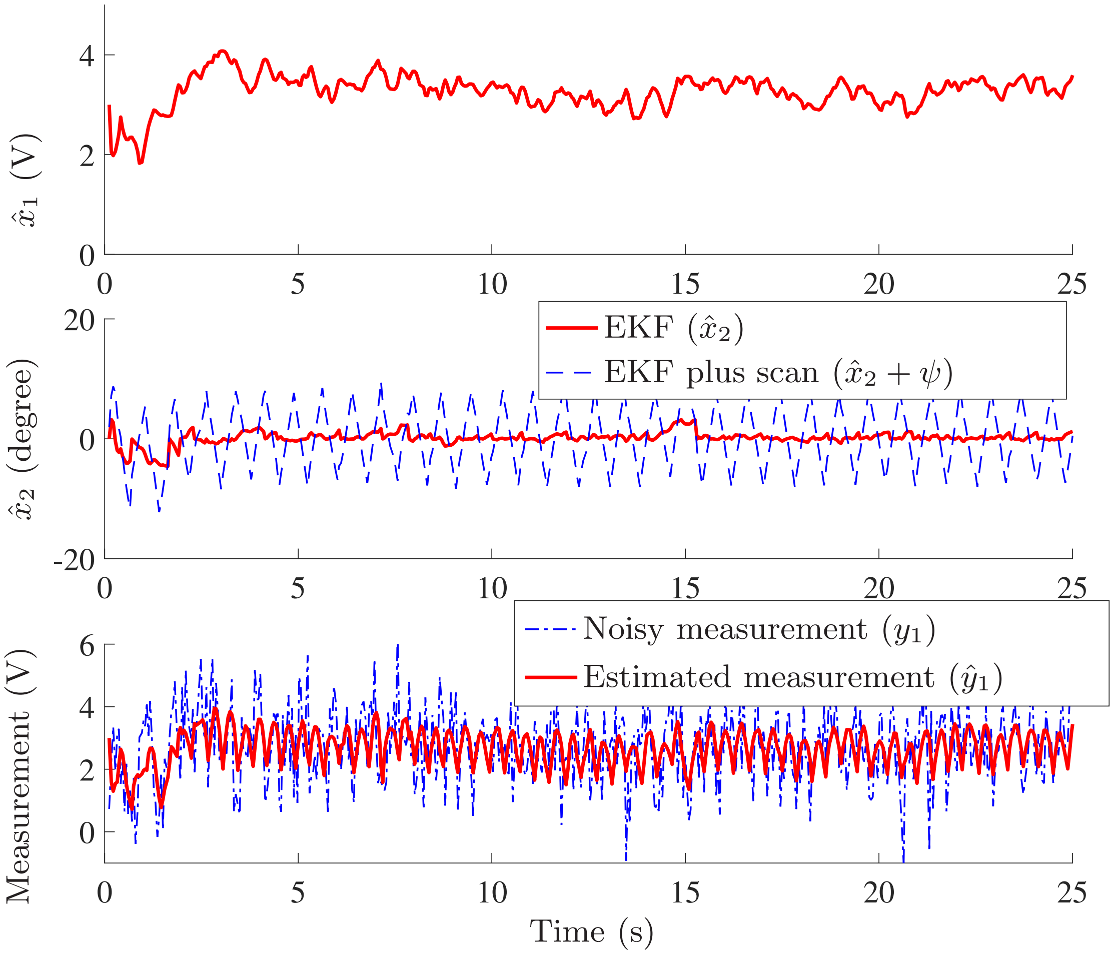


Figure 2.11: Experimental results when the transmitter robot moves around the receiver with an angular rate of about 1 degree/s. The measurements are corrupted with the noise level of 1.0.

To conduct a comparison between the three algorithms, we further perform multiple runs in the experiment. However, unlike simulation, it is not practical to do 1000 runs in the experiments for each case, and we limit the number of experiments to 10 runs. Moreover, when the transmitter robot moves along the circular path, its trajectory is not always consistent. For instance, sometimes it goes a little closer to the receiver and sometimes further. The data generated by this motion is good for qualitative demonstration but to have a fair quantitative comparison between the algorithms, we need a consistent system. Hence we choose to keep the transmitter robot static and introduce an unknown constant disturbance term, similar to β in Eq. (2.25), in the onboard program. This disturbance term forces the receiver to rotate away from the transmitter facing direction. Figure 2.12 shows the plot of the tracking percentage with scaled-down standard deviation error-bars over the range of noise level. It is to be noted that since the angular data is not available, the tracking is computed by thresholding of light intensity measurements. As we can see, the behaviors of the algorithms in the experiment are similar to those observed in simulation. and the EKF algorithm has higher tracking performance with gradual degradation as compared to other two algorithms. The variance in the tracking percentage of EKF is also lower than the other two alternative algorithms. Other than the comparative performance evaluation, if we consider the noise level of 1.0, which is one-third of the maximum intensity or 22dB SNR (Signal to noise ratio), we have more than 95% of tracking.

2.6 Chapter Summary

In this work, a compact LED-based communication system using a single receiver with active alignment control has been presented. For tracking, we have tested the applicability of simpler algorithms like the hill-climbing and the three-point averaging methods. These methods are good for low noise environments but their performance degrades steeply at higher noise scenarios. Whereas, a principled approach using state estimation and control like our proposed EKF-based alignment control algorithm not only gives comparable results in lower noise cases but performs robustly in the case of higher noise environment. In our approach, the motion of the transmitter

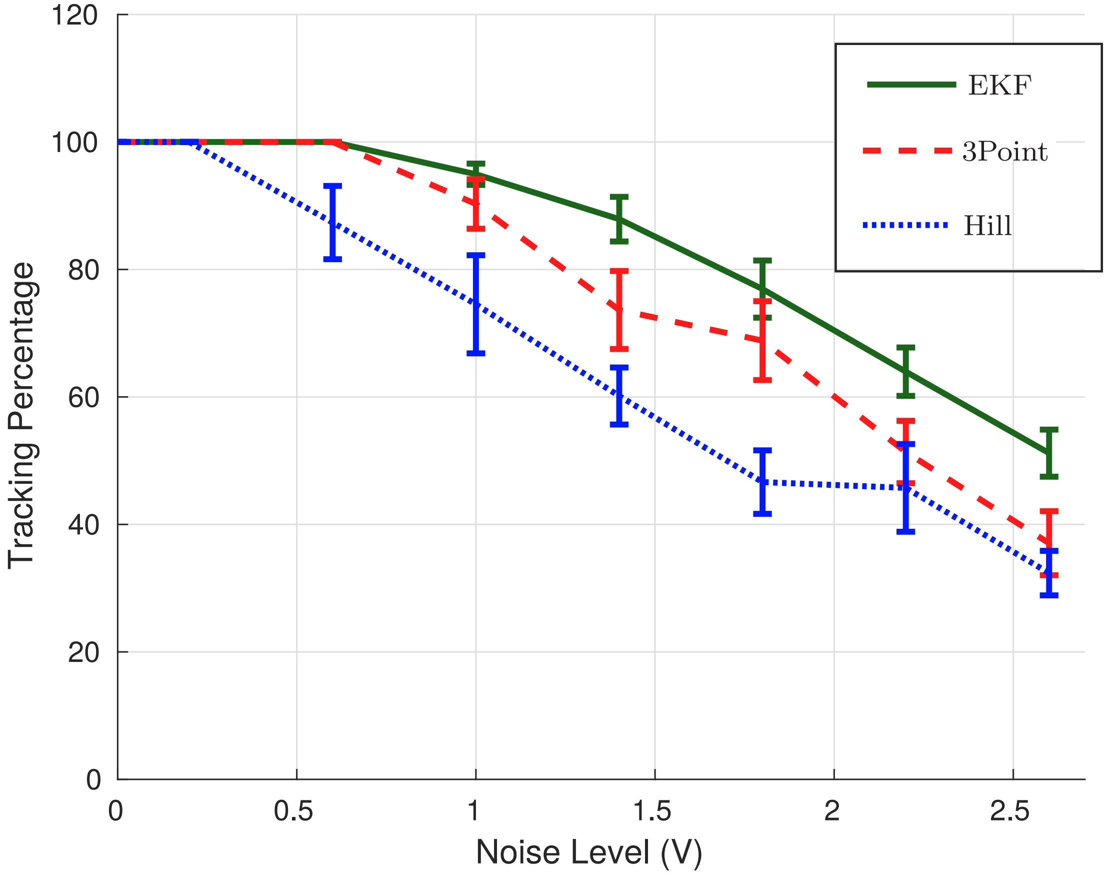


Figure 2.12: Experimental results on comparison of alignment control performance for the three methods, for different levels of measurement noise when the constant disturbance (β) is 2.8°/s. Vertical bars denote the down-scaled standard deviations at each point.

was assumed to be unknown and captured as part of a white Gaussian noise. A scanning technique is implemented to satisfy the observability criterion required for the EKF. A simple proportional controller is used for active alignment.

As mentioned in Section 2.3, the main focus of this work was to demonstrate the instrumental role of nonlinear estimation and feedback control in the active alignment of the LED and the receiver. In particular, this approach uses the measurement history to estimate the state variables and subsequently applies a control action based on the state estimate. It outperforms algorithms that simply react to the current measurement (such as the hill-climbing and three-point-averaging algorithms) in the presence of measurement noises, at a modest cost of implementation complexity.

CHAPTER 3

EKF-BASED ALIGNMENT CONTROL IN THE 3D SPACE

In this chapter, we extend the active alignment control approach to three-dimensional (3D) space. We propose an extended Kalman filter (EKF)-based alignment approach, where the estimates of azimuthal and elevation components of the heading bias with the LOS are used to correct the alignment. We introduce and implement a new circular scanning technique on a two-degree-of-freedom (DOF) rotational system, mounted on a robot, that enables consecutive independent measurements from a single photo-diode, which are necessary to satisfy the observability constraints of the EKF. Furthermore, we explore a synchronized alternating scheme to extend the approach to a system of two robots in a bi-directional setting, where both robots participate in the alignment scheme. The scanning amplitude is further adjusted based on the EKF estimation covariance, to balance the trade-off between estimation accuracy and actuation effort. We compare the proposed approach with an extremum-seeking (ES) approach in both simulation and experimentation, where a setup of two robots with relative 3D motion is considered. The presented results support the efficacy of the proposed method in the presence of slow relative motion between the robots, and demonstrate the superiority of the proposed approach over the ES method over a wide range of distances between the robots.

The organization for the rest of the chapter is as follows. In Section 3.1, the hardware setup of the system is discussed, followed by a discussion on the light signal strength model for state-space formulation. In Section 3.2, the EKF-based alignment algorithm is described, followed by a brief description of the ES-based method. Discussion on simulation setup and results are presented in Section 3.3, while experimental setup and results are reported in Section 3.4. Finally, concluding notes are provided in Section 3.5.

3.1 System Setup and Modeling

In this section, we first briefly describe the hardware-setup, which is essential in the subsequent mathematical modeling of the received light intensity. We then formulate the state-space representation of the system, which is the basis for EKF algorithm development.

3.1.1 System Setup

Figure 3.1 illustrates the transceiver mounted on the two-DOF mechanism, along with miscellaneous components of the setup. The transceiver comprises a transmitter, Cree XR-E Series LED with principal wavelength around 480 nm, and a receiver, photo-diode from Advanced Photonix (part number PDB-V107). Lens-optics is mounted on each of the devices to adjust the field of view. Further, to enable dynamic adjustment of transceiver pointing direction in 3D, a two-DOF rotation mechanism is used. The mechanism consists of two Dynamixel® servo motors; the first one, called *base servo*, adjusts the azimuthal orientation of the transceiver, and the second servo motor adjusts the elevation orientation of the transceiver. For the onboard computation and processing, we used a Beaglebone-Blue® board.

In our work, we consider the signal from a single photo-diode, which can be used for both alignment control and communication. We assume that the optical signal has an approximately equal distribution of low and high values, and hence the average intensity stays approximately equal to the half of the high value of the signal. This chapter focuses only on the alignment control aspect of the problem. Please see Chapter 5 for details on the simultaneous alignment and communication. Next, we briefly discuss the coordinate frames related to our physical setup and define the variables associated with the light signal strength model used in this work.

3.1.2 Coordinate Frames and Received Light Intensity Model

We consider a scenario of two robots, say $\mathbf{R}_1$ and $\mathbf{R}_2$, located at arbitrary positions in a 3D space. Each robot has its own set of two local frames of reference; without the loss of generality, we

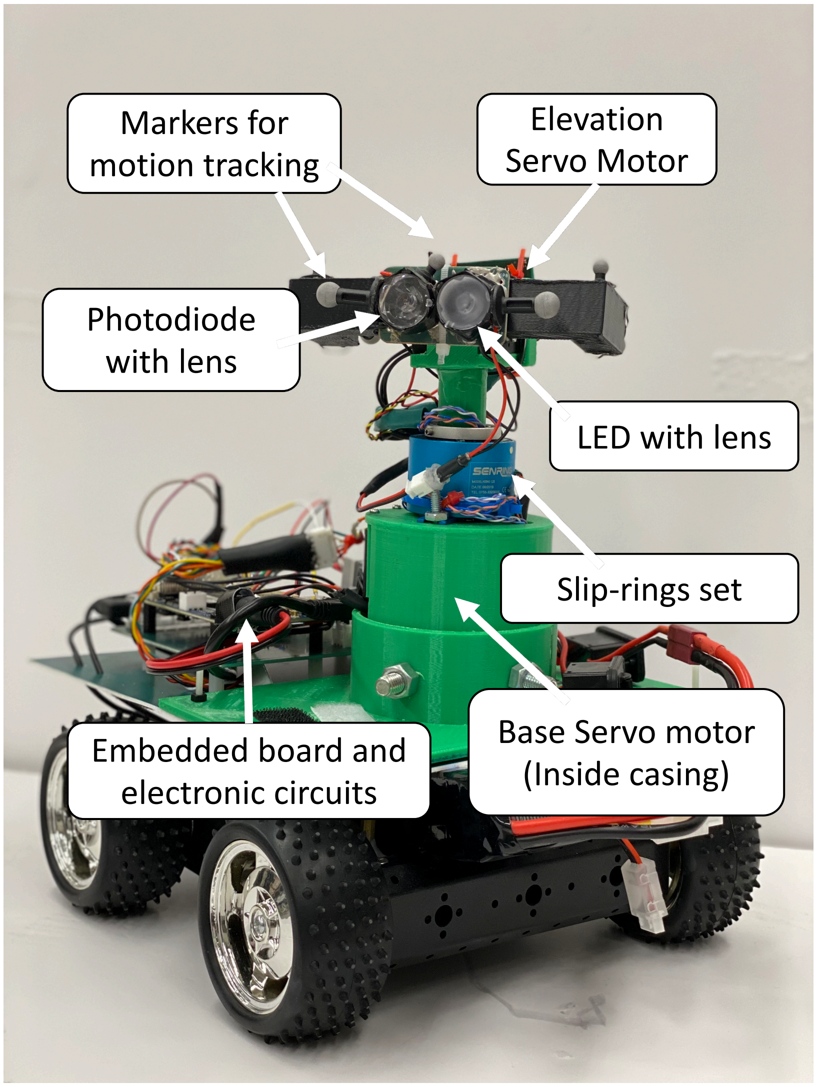

Markers for motion tracking
Elevation Servo Motor
Photodiode with lens
LED with lens
Slip-rings set
Embedded board and electronic circuits
Base Servo motor (Inside casing)

Figure 3.1: Hardware setup, where the transceiver (transmitter LED and receiver photo-diode) mounted on two-DOF active pointing mechanism is shown.

study the frames of $\mathbf{R}_1$ that help in developing the intensity model of the light transmitted by $\mathbf{R}_2$ and received by $\mathbf{R}_1$. Figure 3.2 illustrates two 3D coordinate frames determined by the relative orientation of the two robots, at robot $\mathbf{R}_1$'s receiver end. For modeling purposes, both the receiver and the transmitter of the same robot are considered to be on a single point called a *transceiver* and their normal axes are aligned. The coordinate frames are denoted as the *base frame* $(o-x'y'z')$ and the *transceiver frame* $(o-xyz)$, respectively. Both coordinate frames share the same origin, which is the location of the transceiver of $\mathbf{R}_1$. The following sequential rule defines the axes of the frames:

- y' is the rotor's axis of base stepper motor of $\mathbf{R}_1$.
- x is the heading direction (normal) of the receiver of $\mathbf{R}_1$.
- Both z and z' are identical and are chosen as a perpendicular axis to the plane containing x and y'.
- x' and y are decided by the right-hand rule.

Figure 3.2 also describes all the associated variables of interest in our analysis. The parameter d is the distance between the transceivers of two robots. The line connecting the two robots' transceivers represents the desired direction of LOS. The angle between the transmitter orientation of $\mathbf{R}_2$ and the LOS is the transmission angle γ. The angles ϕ and θ represent two orthographic projections of the angle between the LOS and the pointing direction x, where θ denotes the elevation, and ϕ denotes the azimuthal orientation in the transceiver frame.

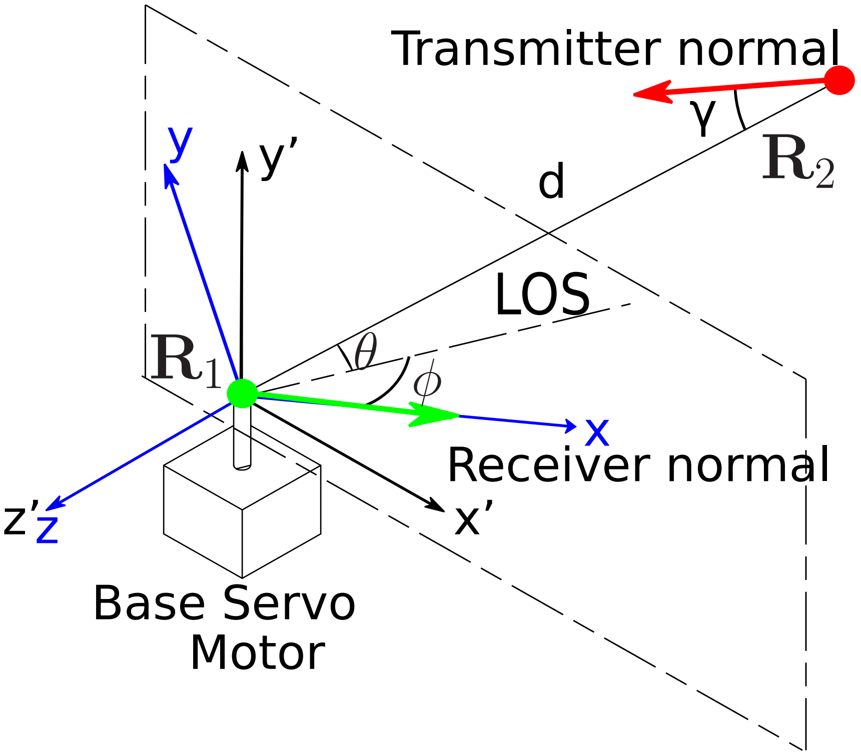


Figure 3.2: Illustration of two local 3D coordinate frames and the associated variables to define the relative position and orientation between the transceivers of two robots.

The light signal strength model here is derived from Section 2.2, with minor modifications to accommodate the 3D setting. From Eq. (2.2), considering the effect of the area and orientation of receiver (combination of the lens and the photo-diode), the final received power is:

$$P_{\text{in}} = E_{\gamma}(d) A_0 f(\phi, \theta) \tag{3.1}$$

where A_0 denotes the lens aperture area and $f(\phi, \theta)$ characterizes the effect of incidence angles ϕ and θ on the received intensity of light. The term $f(\phi, \theta)$ depends on the combination of the lens and the photo-diode used in the receiver. To characterize the function $f(\phi, \theta)$ for our receiver setup, we first collected the measurement data at different values of ϕ and θ at a distance of 1 m. Figure 3.3 shows the collected light intensity data. The collected data is observed to be circularly symmetric about the peak. Henceforth, for practical usage and simplicity in the formulation, $f(\phi, \theta)$ is approximated by a one-dimensional function of the receiver's heading angle with respect to the LOS, which is the equivalent angle combining ϕ and θ:

$$f(\phi, \theta) = g\left(\arccos\left(\cos\phi \cos\theta\right)\right), \tag{3.2}$$

where $g(\cdot)$ is the Gaussian function obtained by performing curve-fitting on the normalized version measurement data, the expression of $g(\cdot)$ is provided in Eq. (2.4) in Chapter 2. The final measurement model can be summarized as

$$V_d = \mathbb{C}_p I_{\gamma} e^{-cd} g(\arccos\left(\cos\phi \cos\theta\right))/d^2 \tag{3.3}$$

where V_d is the signal strength in volts, and $\mathbb{C}_p$ is a proportionality constant, which depends on the lens aperture of the receiver along with various parameters associated with the photo-diode's signal filter and amplifier stages.

3.1.3 State-space Problem Formulation

We observe from Figure 3.2 and Eq. (3.3), that the received signal strength depends on the four independent variables, γ, d, ϕ and θ. In light of the discussion provided in Section 2.2.2, we can

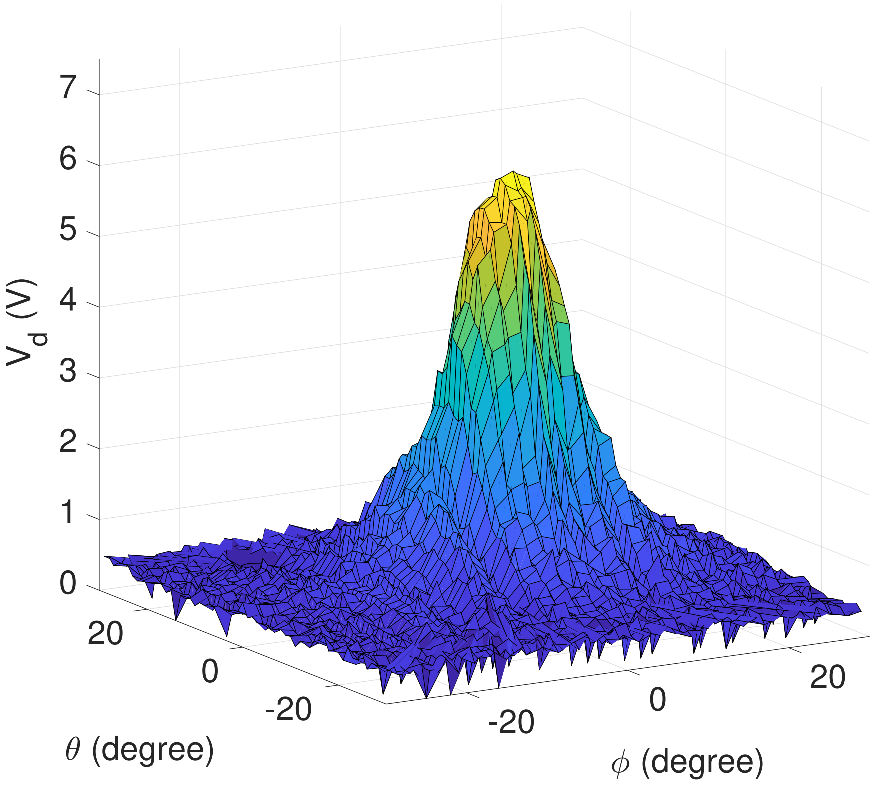


Figure 3.3: Light intensity data for Gaussian curve fitting to approximate $f(\phi, \theta)$.

merge the terms comprising γ and d in a single variable and define the state variables as

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \triangleq \begin{bmatrix} \mathbb{C}_p I_\gamma e^{-cd}/d^2 \\ \phi \\ \theta \end{bmatrix} \tag{3.4}$$

The states in $\mathbf{x}$ depend on the distance and relative orientation between the transceivers of the two robots. In a typical scenario, a robot does not have the exact information about the motion of the other robot. So we will assume that the relative motion between the two communicating robots is slow enough that a constant model with Gaussian noises can capture the relative dynamics. In particular, the dynamics for the state variables defined in (3.4) can be represented in the discrete-time domain as

$$\mathbf{x}_k = \begin{bmatrix} x_{1,k} \\ x_{2,k} \\ x_{3,k} \end{bmatrix} = \begin{bmatrix} x_{1,k-1} + w_{1,k-1} \\ x_{2,k-1} + u_{2,k-1} + w_{2,k-1} \\ x_{3,k-1} + u_{3,k-1} + w_{3,k-1} \end{bmatrix} \tag{3.5}$$

where $w_{1,k}$, $w_{2,k}$ and $w_{3,k}$ represent process noise terms. We assume these noises to be independent, white, and Gaussian. These noise terms, up to a certain extent, account for the slow dynamics of $\mathbf{x}_k$, which we do not explicitly model. The terms $u_{2,k}$ and $u_{3,k}$ are the control inputs for adjusting the transceiver's orientation. These two inputs are defined in the transceiver frame and thus require a rotational transformation to convert them into the base frame, in order to generate motor commands. The details of the transformation are discussed later in Appendix A. The system output at time instant k, the measurement $V_{d,k}$, can be expressed as:

$$V_{d,k} = x_{1,k} g(\arccos\left(\cos x_{2,k} \cos x_{3,k}\right)) + v_k \tag{3.6}$$

where v_k is an additive white Gaussian noise, assumed to be independent of the process noise terms.

Given a sequence of measurements, the objective is to estimate $\mathbf{x}_k$, which is then used to design the control term $\mathbf{u_k} = \left[u_{2,k}, u_{3,k}\right]^T$, in order to drive x_2 and x_3 (termed as *angular states*) towards 0°, the configuration corresponding to the LOS.

3.2 Estimation and Alignment Algorithms

In this section, we first examine the observability of the formulated state-space model and then introduce the circular scanning technique. Moving further, we discuss the implementation of the EKF algorithm with a few refinements and the extension to the bi-directional setting, followed by a brief description of the implementation of the extremum-seeking control algorithm.

3.2.1 Observability of the System

For the nonlinear measurement model (3.6), an extended Kalman filter [39] is explored for state estimation. From the (linear) dynamic equation (3.5), we define

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \; B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{3.7}$$

Then equation (3.5) can be re-written as:

$$\mathbf{x}_k = A\mathbf{x}_{k-1} + B\mathbf{u}_{k-1} + \mathbf{w}_{k-1}, \tag{3.8}$$

with $\mathbf{w}_k = \left[w_{1,k}, w_{2,k}, w_{3,k}\right]^T$. Denoting $C_k \triangleq C(\mathbf{x}_k)$ as the system's linearized output matrix at the kth time instant, the nonlinear observability matrix at that time instant, obtained from [39], can be expressed as

$$O_k = \begin{bmatrix} C_k \\ C_{k+1}A \\ C_{k+2}A^2 \end{bmatrix} = \begin{bmatrix} C_k \\ C_{k+1} \\ C_{k+2} \end{bmatrix}. \tag{3.9}$$

The matrix O_k needs to be full rank at each time instant k, for the system to be observable [39], which is a necessary condition for the stability of the EKF estimate. A possible way to satisfy this criterion is to ensure the linear-independence of three consecutive C_k's with each other, which requires at least three successive measurements of light intensity to be independent. In our system, we ensure the linear-independence by taking the measurements from three non-planar pointing directions of the transceiver. Henceforth, we introduce a circular scanning technique, where the pointing direction of the transceiver moves in a circular manner centered around a mean orientation. This mean orientation, which is what the control input modulates, is considered as the new modified states (x_2, x_3).

The scanning pattern depends on two parameters: scanning amplitude δ_r and angular step δ_ψ (see Figure 3.4), where δ_r modulates the radial angular displacement of the measurement orientation from the mean orientation. The parameter δ_ψ is the angular separation between the two successive measurement orientations. The term ψ_k accounts for the relative angular position of a measurement orientation with respect to the mean (x_2, x_3) orientation at the kth instant and it has two orthogonal components α_k and β_k as shown in (3.10):

$$\begin{cases} \psi_k = \psi_{k-1} + \delta_\psi \\ \beta_k = \delta_r \cos(\psi_k), \\ \alpha_k = \delta_r \sin(\psi_k). \end{cases} \tag{3.10}$$

Figure 3.4 explains the scanning technique. The pointing orientation moves around the mean (x_2, x_3) in successive angular steps of size δ_ψ. The measurement obtained at each ψ_k is considered as our output $\mathbf{y}_k \in \mathbb{R}^1$:

$$\mathbf{y}_k = x_{1,k} g\left(\xi\left(x_{2,k}, x_{3,k}, \beta_k, \alpha_k\right)\right) + v_k, \tag{3.11}$$

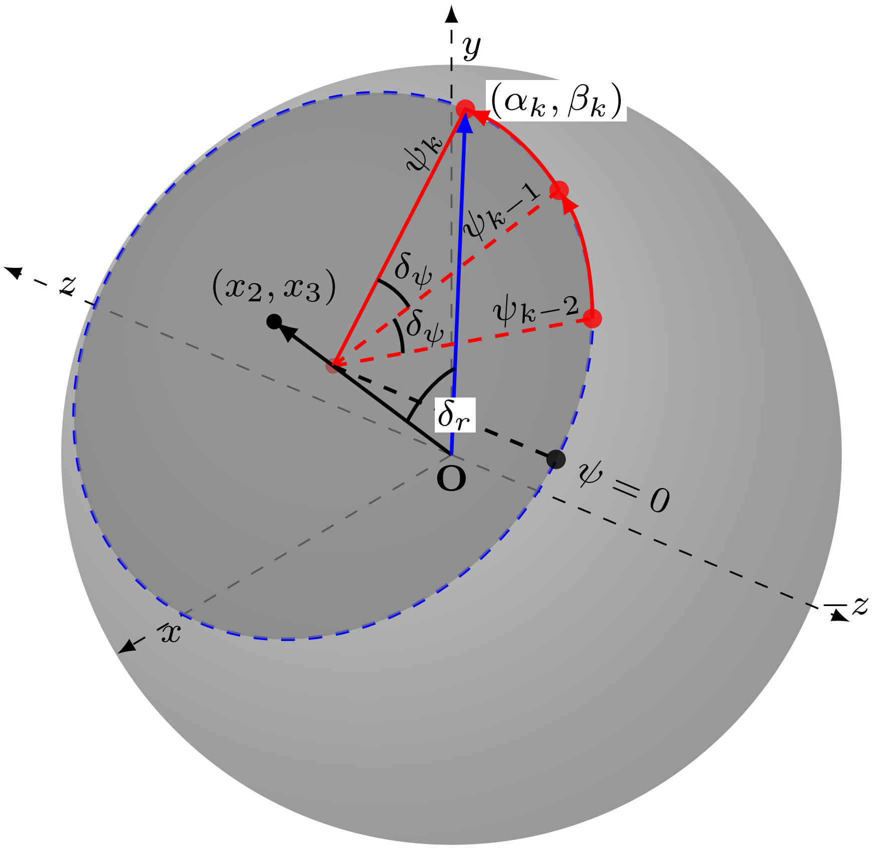


Figure 3.4: Illustration of the circular scanning sequence, with mean pointing orientation (x_2, x_3) and the last three angular positions of scanning ψ_k, ψ_{k-1} and ψ_{k-2}.

where $\xi(\cdot)$ computes the net angle between the LOS direction and the current pointing direction, which depends on both the mean and the scanning terms. The details of the function $\xi(\cdot)$ are covered in Appendix B.

3.2.2 Implementation of Extended Kalman Filter

With the system dynamics (3.8) and the output equation (3.11), an extended Kalman filter (EKF) can now be implemented. The complete algorithm description is as follows.

The EKF has three covariance matrices: $P \in \mathbb{R}^{3\times3}$, $Q \in \mathbb{R}^{3\times3}$, and $R \in \mathbb{R}^1$. P denotes the conditional error covariance matrix, Q denotes the process noise covariance matrix, and R is the measurement noise covariance. At step k,

1. **Prediction phase** : The state estimates and error covariance matrix are predicted as:

$$\hat{\mathbf{x}}_k^f = \begin{bmatrix} \hat{x}_{1,k}^f \\ \hat{x}_{2,k}^f \\ \hat{x}_{3,k}^f \end{bmatrix} = \begin{bmatrix} \hat{x}_{1,k-1} \\ \hat{x}_{2,k-1} + u_{2,k-1} \\ \hat{x}_{3,k-1} + u_{3,k-1} \end{bmatrix}, \tag{3.12}$$

$$P_k^f = AP_{k-1}A^T + Q, \tag{3.13}$$

where $\hat{x}_{n,k}^f$ represents the estimate of nth state at the kth time interval and the superscript f stands for ‘forecast’ of the associated entities.

2. **Output estimation**:

From (3.11), we get

$$\hat{\mathbf{y}}_k \triangleq h(\hat{\mathbf{x}}_k^f) = \hat{x}_{1,k}^f g\left(\xi\left(\hat{x}_{2,k}^f, \hat{x}_{3,k}^f, \beta_k \alpha_k\right)\right) \tag{3.14}$$

Now the matrix $C_k \in \mathbb{R}^{1\times 3}$ can be computed as:

$$C_k \triangleq \frac{\partial h(\hat{\mathbf{x}}_k^f)}{\partial \hat{\mathbf{x}}_k^f} = \left[g(\hat{\xi}_k^f), \frac{-\hat{x}_{1,k} g'(\hat{\xi}_k^f)}{\sqrt{1-(\hat{\chi}_k^f)^2}} \left[\frac{\partial \hat{\chi}_k^f}{\partial \hat{x}_{2,k}}, \frac{\partial \hat{\chi}_k^f}{\partial \hat{x}_{3,k}} \right] \right], \tag{3.15}$$

where $g'(\cdot)$ indicates the derivative of $g(\cdot)$ and

$$\hat{\xi}_k^f \triangleq \xi\left(\hat{x}_{2,k}^f, \hat{x}_{3,k}^f, \beta_k, \alpha_k\right) = \arccos(\chi_k^f), \text{ and}$$
$$\hat{\chi}_k^f \triangleq \chi\left(\hat{x}_{2,k}^f, \hat{x}_{3,k}^f, \beta_k, \alpha_k\right).$$

The details of the functions $\chi(\cdot)$, $\xi(\cdot)$, and their derivatives are discussed in Appendix B.

3. **Analysis/update phase**:

$$\begin{cases} K_k = P_k^f C_k^T (C_k P_k^f C_k^T + R)^{-1} \\ \hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^f + K_k(\mathbf{y}_k - \hat{\mathbf{y}}_k) \\ P_k = (I_3 - K_k C_k) P_k^f \end{cases} \tag{3.16}$$

Here $K_k \in \mathbb{R}^{3\times 1}$ denotes the filter gain. The matrix I_3 denotes the 3 × 3 identity matrix and it is to be noted that P_k and $\hat{\mathbf{x}}_k$ with no superscripts denote the final estimated values of the kth step.

Using the state estimates, the control terms can now be computed as

$$\begin{cases} u_{2,k} = -G_P\hat{x}_{2,k} - G_I\hat{z}_{2,k} \\ u_{3,k} = -G_P\hat{x}_{3,k} - G_I\hat{z}_{3,k} \end{cases} \tag{3.17}$$

where the terms $\hat{z}_{2,k}$ and $\hat{z}_{3,k}$ are the integrals of the estimates $\hat{x}_{2,k}$ and $\hat{x}_{2,k}$, respectively, which are defined by:

$$\hat{z}_{i,k} \triangleq \sum_{n=0}^{k-1} T\hat{x}_{i,n} = \hat{z}_{i,k-1} + T\hat{x}_{i,k-1},\ i \in \{2,3\} \tag{3.18}$$

where T is the sampling time. The positive constants G_P and G_I account for the proportional and integral gains of the PI-controller, respectively. The gains are designed to ensure the closed-loop stability of the two-states; for more details, see [49].

Note that the above calculation is conducted in the local transceiver frame of $\mathbf{R}_1$, and the control terms need to be translated to the base frame to generate final commands for motors. The details of the command translation are discussed in Appendix A.

Moreover, when the mean of the scan achieves a steady state, there are still oscillations in the intensity measurements due to the scanning motion. These oscillations limit the average intensity below the available maximum. This difference of intensity can be critical during actual communication, as lower light intensity leads to a weaker signal to noise ratio (SNR) and results in a higher bit error rate. To avoid such cases and additionally, to reduce the power consumed in the scanning motion [51], we propose an proportional scanning technique. Here, the amplitude of scanning is made proportional to the time-average of the norm of the sub-matrix of P (denoted as P_{ang}) corresponding to the angular components, where

$$P_{ang} = \begin{bmatrix} P_{2,2} & P_{2,3} \\ P_{3,2} & P_{3,3} \end{bmatrix} \tag{3.19}$$

with $P_{i,j}$ denotes the entry in the ith row and jth column of the P matrix:

$$\delta_{r,k} = \max\left(\delta_l, \min\left(K_\delta \sum_{i=k-n_s+1}^{k} \|P_{ang,i}\|_2, \delta_h\right)\right) \tag{3.20}$$

with the 2-norm $\|\cdot\|_2$ being the largest eigenvalue of a matrix, and δ_h and δ_l indicate the upper and lower bounds of the scanning amplitude, respectively. K_δ denotes the proportional gain and n_s is the number of iterations in an active scanning period, which is equal to $\frac{360^o}{\delta_\psi}$. The lower bound is implemented due to the constraint of the three independent measurements from the observability criterion and hence, the scanning amplitude cannot be made zero.

3.2.3 Extension to the Bidirectional Scenario

So far, we have discussed the formulation and approach for a single directional scenario where the receiver tries to align itself towards the direction of the maximum light intensity. Now, consider the bidirectional case of two robots, where each of the robots is equipped with a transceiver. Here we cannot implement the aforementioned algorithm on each robot as the simultaneous scanning would violate the quasi-static assumption on the state x_1 of each of the robot. Therefore, we implement an alternating pausing-scanning approach. Here one robot starts with the active scanning phase while the other with the passive pausing phase and they alternate afterward. The active and passives phase are described as follows.

1. **Active phase:** In this phase the robot completes a circular scan of the EKF approach presented earlier. The total number of steps in one circular scan equals to $360°/\delta_\psi$. After one circle of the scan, the robot points to the center of the scan and switches to the passive phase.

2. **Passive phase:** In this phase a robot fixes its orientation; scanning amplitude (δ_r) and the control gains (G_p,G_I) both are zero. Due to the zero scanning radius, the observability matrix of the system has rank one and hence the original system is not observable. Therefore, we implement the EKF only for the state x_1 in this phase, detailed as follows:

$$\hat{x}^f_{1,k} = \hat{x}_{1,k-1} \tag{3.21}$$

$$P^f_{k,1,1} = P_{k,1,1} + Q_{1,1} \tag{3.22}$$

$$K_k = P^f_{k,1,1} C_{k,1} (P^f_{k,1,1} C^2_{k,1} + R)^{-1} \tag{3.23}$$

$$\hat{x}_k = \hat{x}^f_k + K_k (\mathbf{y}_k - \hat{\mathbf{y}}_k) \tag{3.24}$$

$$P_{k,1,1} = (1 - K_k C_{k,1}) P^f_{k,1,1} \tag{3.25}$$

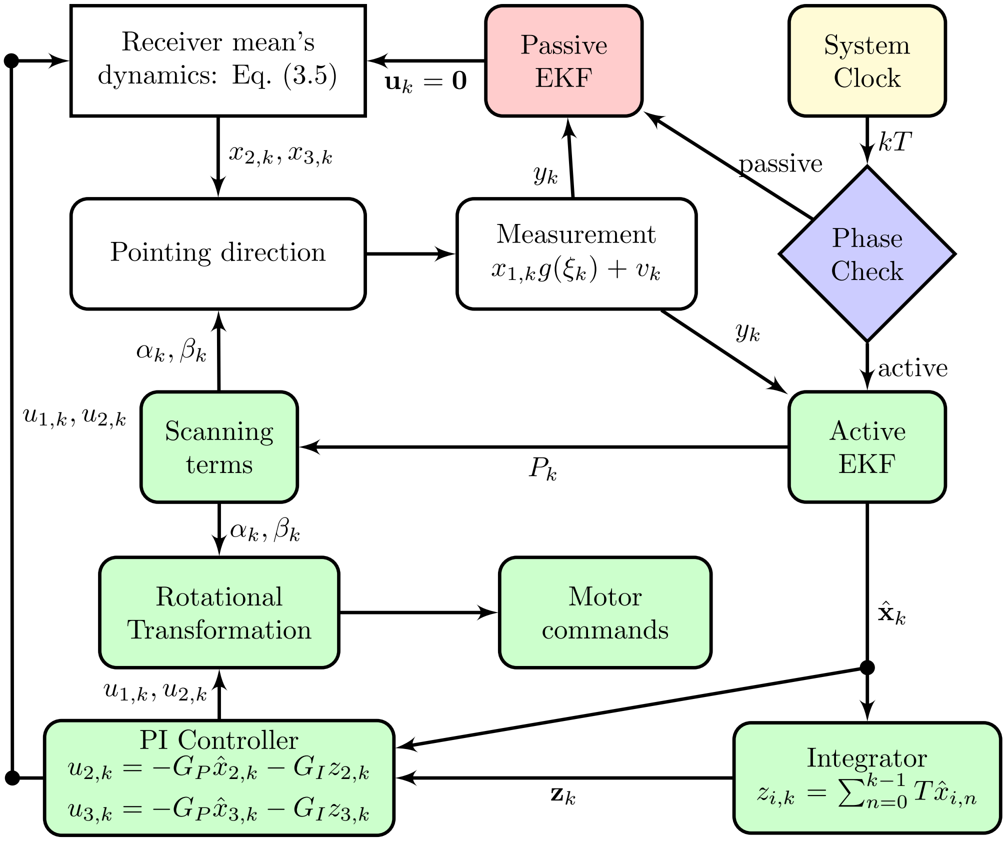


Figure 3.5: Block diagram summarizing the proposed method. All the shaded color nodes denote the steps known to the algorithm. The information at white color nodes is not available to the algorithm.

At each iteration, the other entries of $\hat{\mathbf{x}}_k$ and P_k are carried to the next iteration. This way at the beginning of an active phase the initial conditions for $\hat{x}_2$ and $\hat{x}_3$ are propagated from the end of the previous active phase. After waiting for the predetermined number of iterations required to complete a circular scan, the robot switches to the active phase.

Figure 3.5 summarizes the overall flow of our approach.

3.2.4 Benchmark Approach: Extremum-seeking (ES) Control

Here, we briefly discuss the implementation of the discrete-time ES control method that is used as a benchmark for comparison with our method [25]. The algorithm is typically used to optimize a function in real-time. The light intensity measurement from each robot is used as the function for maximization. The block diagram in Figure 3.6 illustrates the flow of the algorithm implemented on each of the robots. The plant represents the overall system of the two robots, the output from the plant $\mathbf{y}_{i,k}$ is passed through a high-pass filter, which is then multiplied by two separate perturbation signals $\delta_{ES}\sin(2\pi f_{p,i}Tk)$ and $\delta_{ES}\cos(2\pi f_{p,i}Tk)$ to generate the corresponding bias signals, $\xi_{2,k}$ and $\xi_{3,k}$, respectively. The terms δ_{ES} and $f_{p,i}$ represent the amplitude, and the frequency of the perturbation signals for robot $\mathbf{R}_i$, respectively. The bias signals are each multiplied by a gain of K_{ES}, and then added to their corresponding perturbation terms to generate the control terms $u_{2,k}$ and $u_{3,k}$.

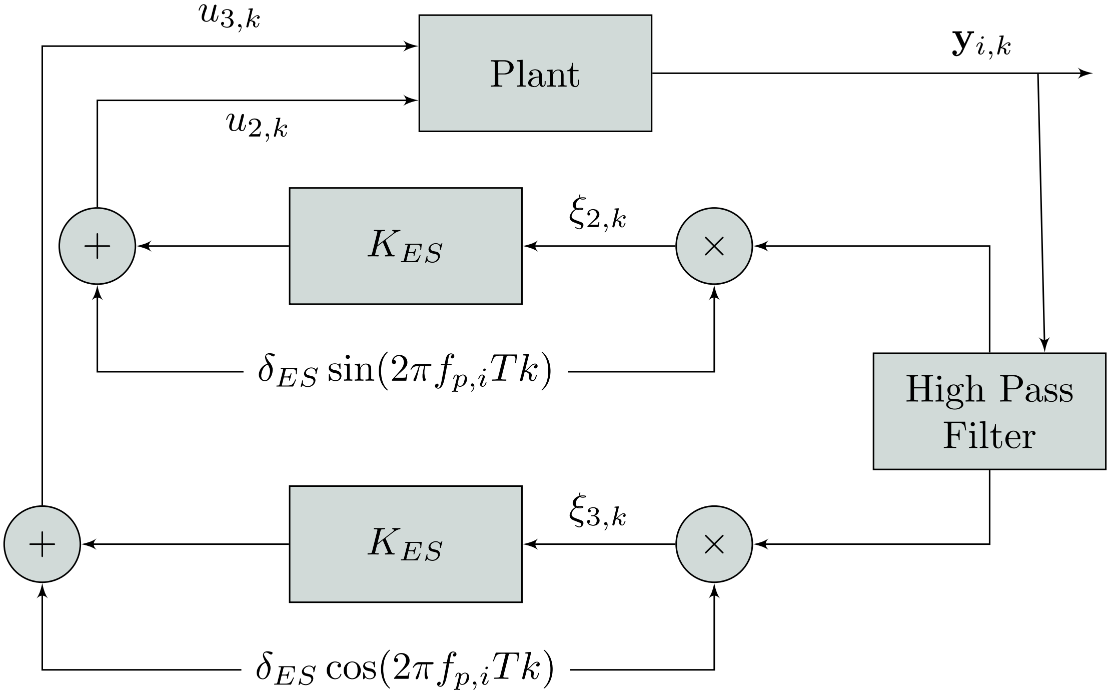


Figure 3.6: Block Diagram for extremum seeking control.

3.3 Simulation Results

In this section, we simulate the proposed approach and the ES control method for a two-robot scenario in MATLAB. The parameters used in the simulation are listed in Table 3.1. For EKF, the initial condition for the state estimates is chosen as $[0.5, 0, 0]^T$, where the first term is chosen as a positive value close to the expected voltage at the LOS, and the other two terms are each chosen to be zero as it is an unbiased initial condition. The matrices Q and R are the scaled version of the noise-covariance matrices of the system. In our prior work [45], we compared the performance of EKF over multiple values of scaling-coefficient of Q, and it is shown that the scaled versions improve the performance of the EKF algorithm. The exact values of the matrices are chosen empirically. The initial error-covariance P_0^f is chosen to be same as Q.

First, we consider the case where the two robots are stationary and separated about two meters apart in the 3D space, such that initially, none of them is aligned with the LOS. Figure 3.7 summarizes the evolution of the states and their estimates, the measured light intensity and its estimate, and the scanning/perturbation amplitude by each robot during the course of the algorithm execution. It is observed that, in the case of EKF, for both robots, the states x_2 and x_3, and their estimates converge to a neighborhood of zero in about forty seconds. For the ES, the states converge to a neighborhood of zero in about twenty seconds. The bottom sub-figures show the scanning amplitude values (δ_r) for EKF, along with the constant perturbation amplitude (δ_{ES}) for ES. They illustrate that, when a robot is in the active phase, it has a high value of δ_r, while the other robot's δ_r is zero, which signifies its passive phase. Initially, when the uncertainty in the estimates is high, the value of δ_r is high (4°), and when the states reach the steady-state, the value of δ_r in the active phase changes between 2° and 3°. It can be inferred from the plots that the alternating and proportional nature of the scanning technique in the EKF approach helps in

- achieving lower steady-state error and high-intensity values, and
- reducing the effort by the actuators,

as compared to the ES-based approach where the perturbation amplitude is constant, which results in the higher steady-state error and control effort.

Next, to assess the algorithms' repeatability and study the effect of the distance on the tracking performance, we have conducted multiple simulation runs with robots stay stationary for a range of distances. To characterize the efficiency of tracking, we consider a metric called *average pointing error* $\mathbf{E}$, which is the average of the heading offset angle ξ during a run from both of the robots,

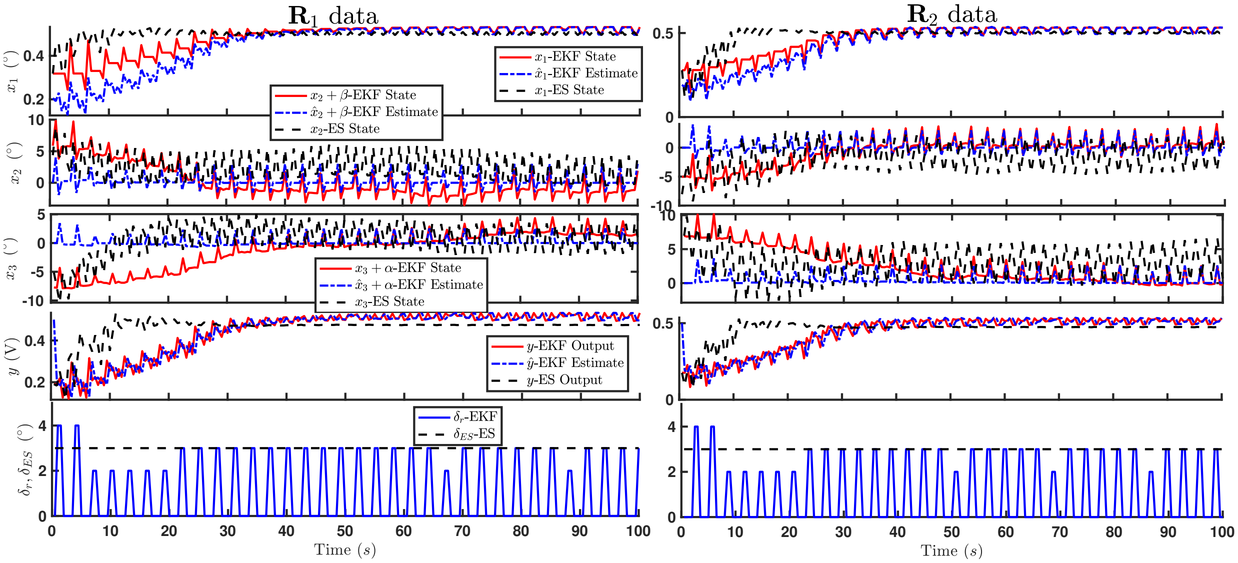


Figure 3.7: Plot of a simulation run illustrating the evolution of the three states and measured intensity for the EKF and the ES approach, and their EKF-estimates, and scanning/perturbation amplitude for each robot, when the robots are stationary. The angular states: x_2, x_3, and their estimates are augmented with scanning terms β and α to illustrate a fair comparison with the angular states of the ES approach.

calculated after the steady-state is achieved, namely

$$\mathbf{E} = \frac{1}{2(n_{max} - n_l + 1)} \sum_{i=1}^{2} \sum_{k=n_l}^{n_{max}} \xi_{i,k},$$

where n_{max} denotes the total number of iterations in each run (we use 200) and n_l is the time index of an iteration that is inside the steady-state (161 is used here to capture the last twenty percent of iterations). Furthermore, we consider another metric called *average intensity* **I**, which considers the average intensity measurement and is defined similarly to **E**:

$$\mathbf{I} = \frac{1}{2(n_{max} - n_l + 1)} \sum_{i=1}^{2} \sum_{k=n_l}^{n_{max}} \mathbf{y}_{i,k},$$

Figure 3.8 shows the effect of the increasing distance on the average intensity and the average pointing error. It also shows the curve of $\mathbf{I}_{\text{LOS}}$, which is the maximum attainable light intensity at a distance, achieved for the case of perfect LOS between the robots. $\mathbf{I}_{\text{LOS}}$ exhibits an inverse quadratic decrease as per Eq. (3.3). For EKF, the average intensity **I** exhibits a similar decreasing trend with the distance, and it stays close to $\mathbf{I}_{\text{LOS}}$. The pointing error **E** stays low for the distance from 1 m to 3 m, and then it starts increasing linearly with the distance. This degradation is due to the decreasing signal to noise ratio (SNR) as the intensity measurement decreases by the inverse square law with the distance while the level of the measurement noise stays constant.

For the ES algorithm, it is observed that at low distances, the pointing error becomes significantly high, and that results in a low average-intensity value. At a low distance, the product of gain K_{ES} and the high-pass-filtered output signal becomes high, and it results in the instability of the ES algorithm. It illustrates that the ES approach would require varied values of gain K_{ES} at different distances; however, it is not practical for a real scenario where the operating distance between the robots could change and may not be known for either robot. In the case of EKF, the estimate of the state x_1 accounts for the change in intensity due to the change in the distance, and hence the approach works for a wide range of the distances. For the current value of K_{ES}, the ES algorithm shows minimum pointing error between the distance of 2 m and 3 m, and the error starts increasing beyond three meters.

In a practical scenario, the robots would be moving; however, in our formulation of EKF, we assumed a quasi-static relative motion between the robots. This assumption would be violated when the robots move at a larger speed. Therefore, to explore the influence of robot movements on the LOS alignment performance, the algorithms are further tested with the robots moving at a range of speeds. Figure 3.9 illustrates the configuration of the motion of the two robots, which is based on the experimental setup to be discussed in Section 3.4. In the setup, the robot $\mathbf{R}_1$, which we call the *Rover*, moves on a horizontal plane, and the robot $\mathbf{R}_2$, which we call the *Elevator*, moves upward. Initially, both the robots are placed at the points of minimum distance $d_{\min}$ between the linear paths of them. Each of the robots moves at a certain speed, and since the directions of their motion are orthogonal to each other, the relative speed s is computed by taking the square-root of the sum of squares of their speeds ($\sqrt{s_1^2 + s_2^2}$) where s_i denotes the speed of the robot $\mathbf{R}_i$.

Next, we consider a range of relative speed values for the robots. At each speed, we perform 1000 runs and compute the average and the standard deviation of the pointing error $\mathbf{E}$. Due to the robots' moving nature, the distance between them changes, and hence the maximum attainable intensity changes through the course of a run, so the average intensity $\mathbf{I}$ is not considered in these

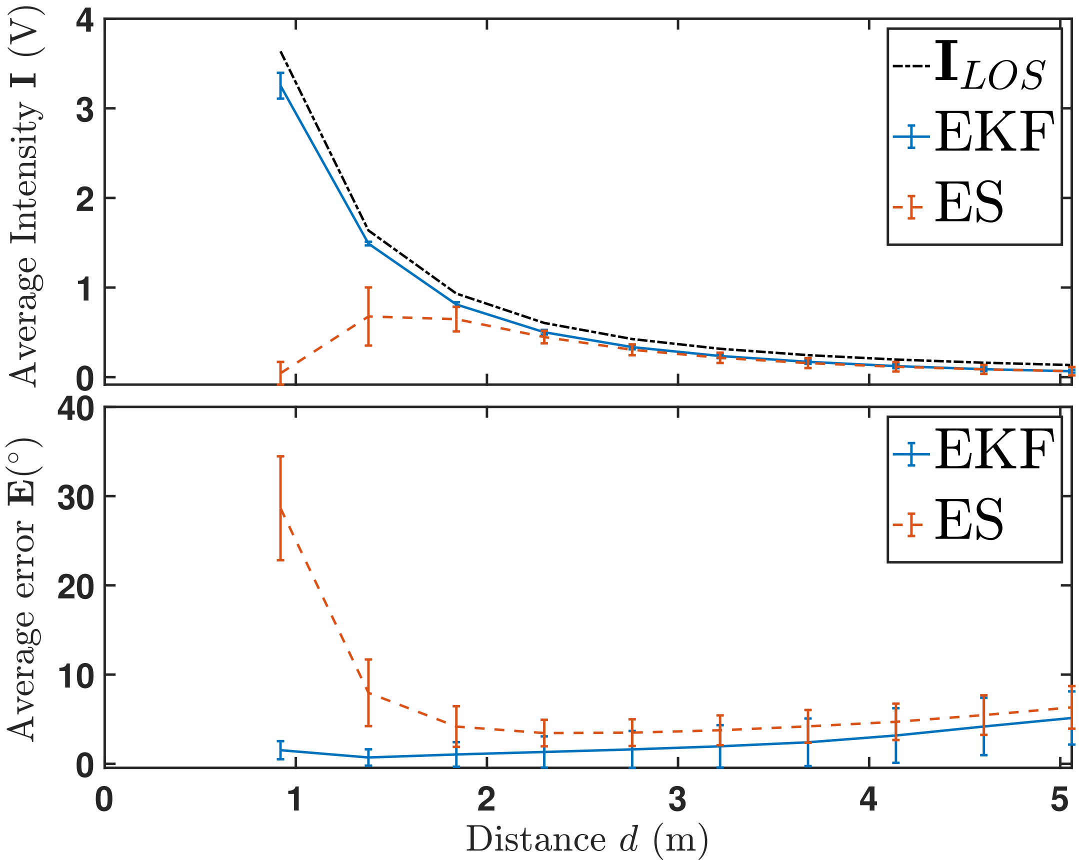


Figure 3.8: Tracking performance in terms of average intensity **I** and average error **E** in simulation over a range of distances between the robots. The error bars denote the standard-deviation. The intensity at LOS ($\mathbf{I}_{\mathrm{LOS}}$) is also shown for reference.

set of simulations. Figure 3.10 illustrates the average pointing error over the range of speed v. It is observed that at low speeds, the pointing error for EKF stays lower than the error for ES; however, as the speed increases, the error for EKF increases at a higher rate as compared to ES. Furthermore, the EKF algorithm stops converging at high speeds that results in high pointing errors. This behavior illustrates the limitation of the EKF algorithm when the quasi-static assumptions on the states are violated. For, ES algorithm, the error stays relatively low as the gain K_{ES} is finely tuned for the distance range of two to three meters (the distance d stays in this range when the robots move), so it exhibits fast convergence and hence can track the LOS, under the relatively fast motion of the robots.

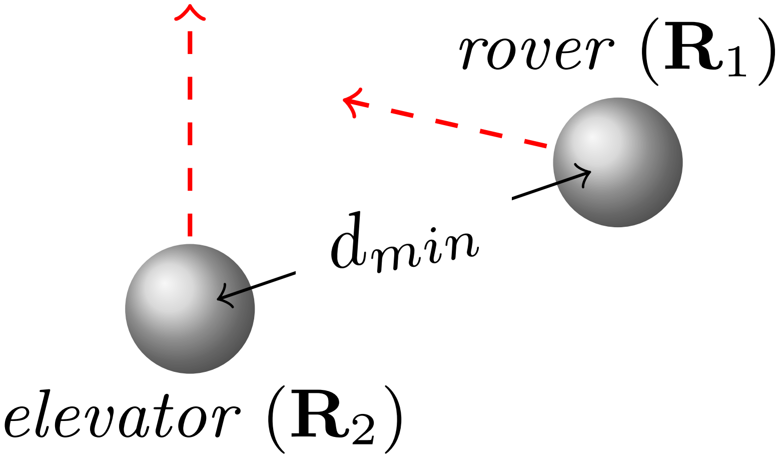


Figure 3.9: Illustration of the initial configuration of moving robots (denoted by spheres) placed d_{min} distance apart in simulation. The elevator robot moves upward, and the rover robot moves horizontally in a direction orthogonal to the line joining the robots' initial locations.

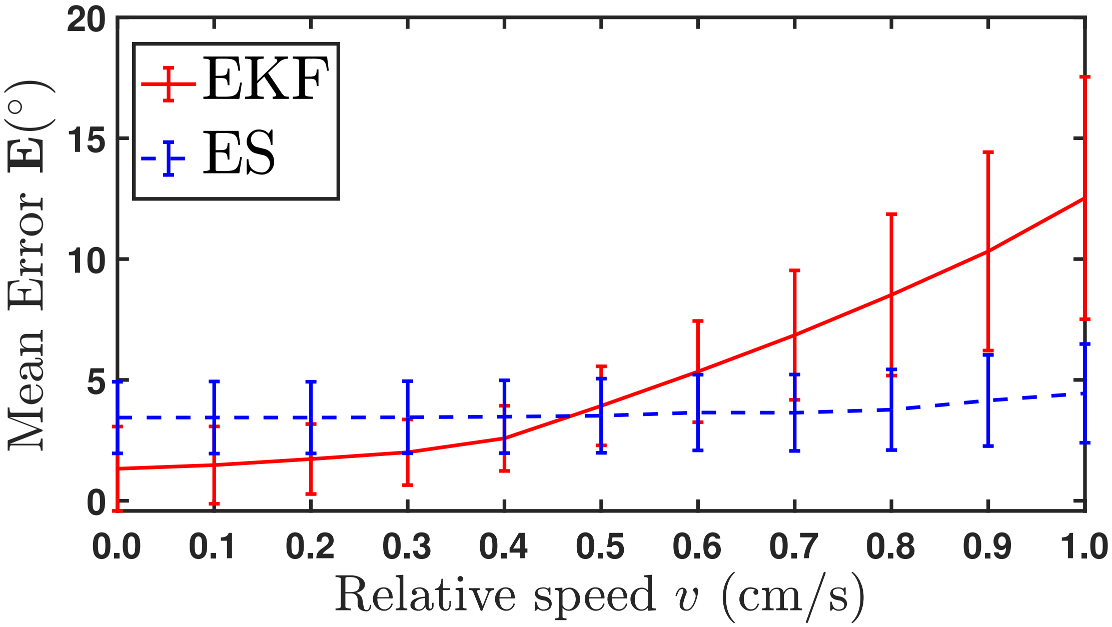


Figure 3.10: Tracking performance in terms of average error **E** in simulation over a range of speeds of the robots.

Table 3.1: Parameters used in the simulation. The values of parameters marked as '♠' are chosen empirically.

Parameter	Value	Description
$\hat{x}_0^f$	$[0.5, 0, 0]^T$	Initial value of the state estimates
P_0^f	diag([10, 900, 900]) ♠	Initial error-covariance matrix
σ_w	0.009	Standard-deviation of system's process noise for angular states x_2 and x_3
Q	diag([10, 900, 900]) ♠	EKF's process noise-covariance matrix
σ_v	5.15e−4	Standard-deviation of system's measurement noise
R	1 ♠	EKF's measurement noise-covariance matrix
$[G_P, G_I]$	$[0.98, 0.2]$	PI controller gains
$\left[\delta_l, \delta_h\right]$	[2, 5] ♠	Scanning amplitude limits
δ_ψ	120° ♠	Scanning angular step-size in degree
K_δ	0.0025 ♠	Proportional scanning gain
T	500 ms	Sampling time
$[\delta_{ES}, K_{ES}]$	$\left[3^\circ, 5\right]$ ♠	Perturbation amplitude and gain for ES control method
$[f_{p,1}, f_{p,2}]$	$\left[0.\overline{66}\ Hz, 0.5\ Hz\right]$ ♠	Perturbation frequencies of the robots for ES control method

3.4 Experimental Results

In this section, we assess the algorithms on an experimental setup. The setup consists of two robots, each equipped with a transceiver and the rotational tracking mechanism, as discussed in Section 3.1. Figure 3.11 shows the actual experimental setup. The elevator robot on the left moves vertically along a pole. A high torque DC motor, along with a pulley and a string, is used to create a controlled vertical motion. The rover robot on the right moves straight on the ground. The distance of the path of the rover is 2.3 m from the pole. A similar setup is modeled for simulation in the

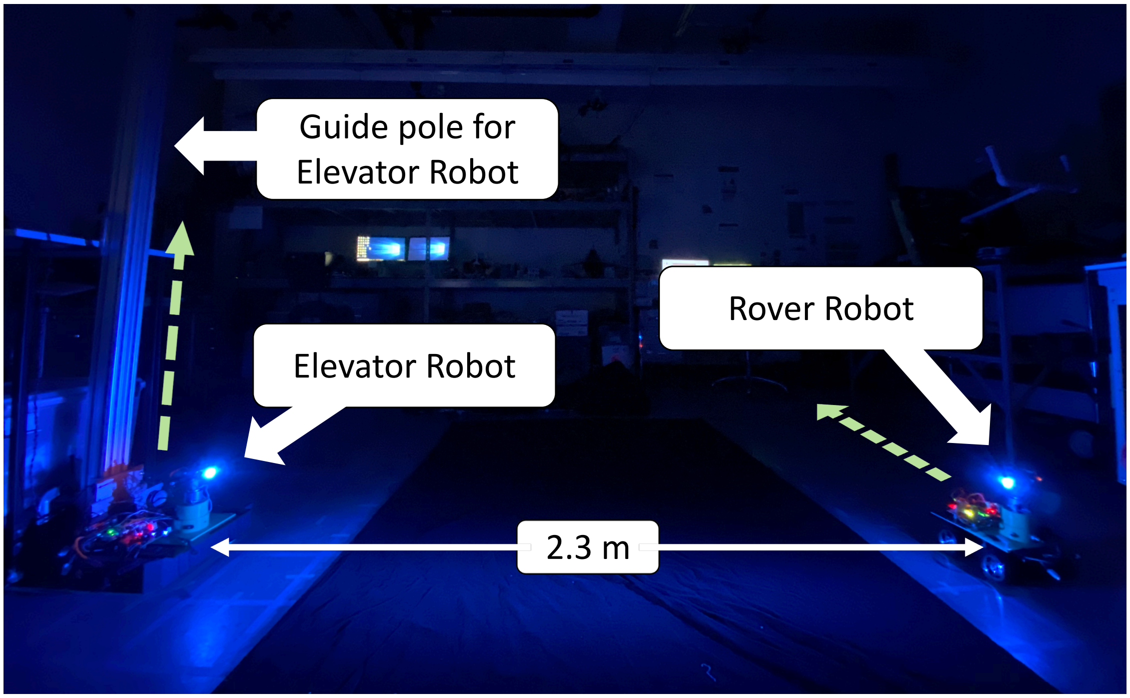


Figure 3.11: Experimental setup with two moving robots in a dark room. The dashed arrows denote the moving direction of the robots.

previous section. The relevant parameters for experiments are also taken from Table 3.1.

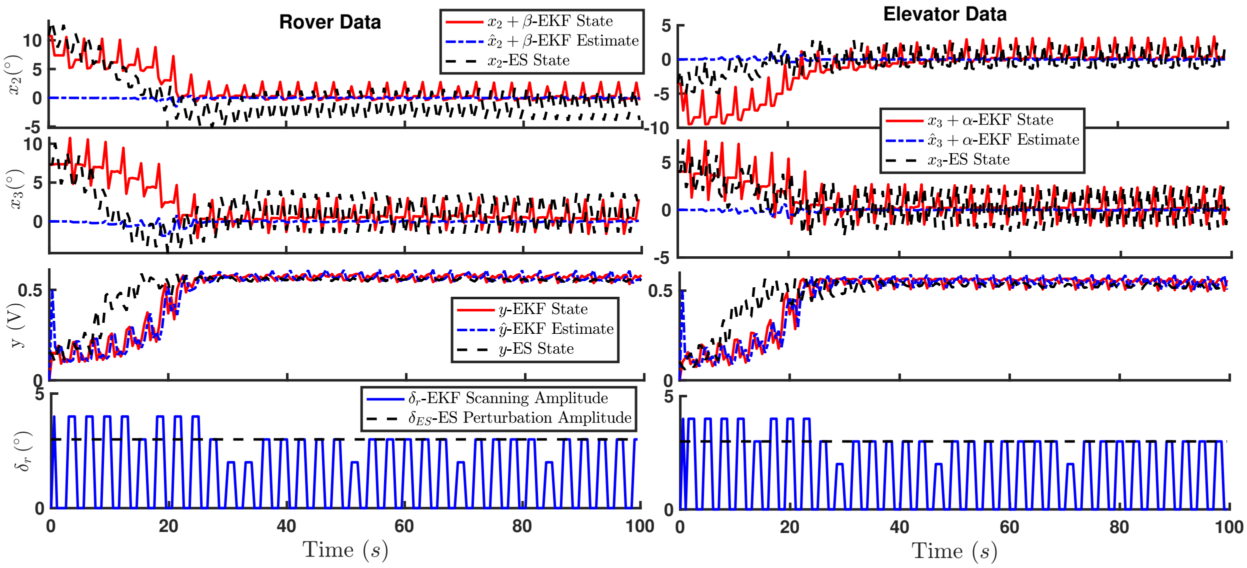


Figure 3.12: Plot of an experiment run illustrating the evolution of the angular states and their estimates (augmented with scanning terms for EKF), intensity measurement and its estimate, and scanning/perturbation amplitude for each robot, when the robots are stationary.

First, we discuss an experiment run with both robots kept stationary. For the EKF approach, a synchronization procedure is implemented, where the command to start the run with the start-time (with sufficient margin to accommodate transmission delays) is sent from an external computer to the on-board Beaglebone boards. The internal clocks of the boards are synchronized over the Wi-Fi network. Additionally, using the start time as the base, the timing of each iteration of both robots is synchronized using their internal clocks without any communication between them. Figure 3.12 shows the results on the evolution of the states, estimated states, intensity measurement, and the scanning amplitude. The actual states x_2 and x_3 are generated from data provided by the motion tracking system. It can be seen that the states x_2 and x_3 for both EKF and ES, for both robots, converge to a neighborhood of zero in about twenty seconds, resulting in the output to attain the steady-state value of around 0.6 V. Overall, the results are similar to what is observed in simulation; however, for EKF, the convergence time is lower than what is observed in simulation.

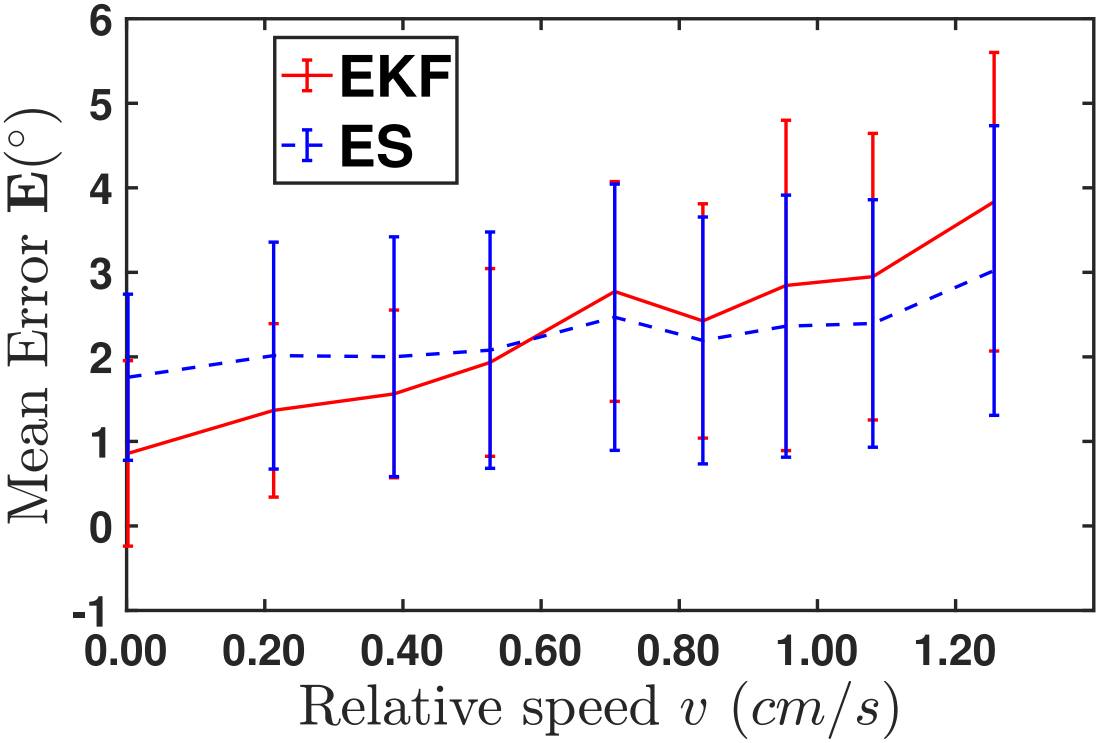


Figure 3.13: Tracking performance of the algorithms in terms of average pointing error **E** over a range of speeds of the robots in experiments.

Next, similar to simulation, we test the algorithm over a range of speeds of the robots. Five runs with the same initial configuration are conducted for each speed. Figure 3.13 shows the tracking performance over different speeds. The characteristics of the plot are similar to what are observed in simulation (Figure 3.10), where at lower speeds, the EKF algorithm shows a lower pointing error than the ES algorithm. Additionally, as the relative speed between the robots increases, due to the violation of the quasi-static assumption, the performance of the EKF algorithm degrades faster than the ES algorithm. However, since the faster convergence is observed in the experiments for EKF, the degradation in the pointing error is significantly slower, and the mean pointing error stays within $4°$.

We acknowledge that the proposed algorithm requires the relative speed between the robots to be relatively small. Faster scanning will mitigate this constraint and allow the system to track and maintain the LOS in the presence of larger relative motions between the communicating parties. The iteration time T in our current experimental system is 500 ms, largely determined by the time it takes the servo motors to go from the current configuration to the next desired configuration. With the availability of better actuators, this iteration time can be reduced by an order of magnitude. With such an upgrade, we expect significant increase of allowable relative motion speed between the communicating robots, which is important for practical applications.

3.5 Chapter Summary

In this work, we propose a bidirectional active alignment control system for LED-based wireless optical communication in the 3D space. With a light signal-strength model, we first formulate a control and estimation problem in the state-space domain. We further proposed a circular scanning technique to take independent successive intensity measurements to satisfy the system's observability criterion for an EKF-based estimation algorithm implementation. The state estimates are then used in a PI controller to achieve the desired alignment. The controller output is later transformed to generate commands for the motors. Additionally, we propose the proportional scanning technique to maximize signal strength at the steady-state and to reduced actuation effort. Moreover, an alternate active switching layer is implemented to achieve bi-directional alignment, where each of the robots takes turns for the scanning. The effectiveness of the proposed approach to achieve LOS is validated in both simulation and experiments involving two mobile robots. Furthermore, the superiority of the approach over an alternative approach is established in terms of efficacy across a range of distances.

CHAPTER 4

SIMULTANEOUS BI-DIRECTIONAL ALIGNMENT CONTROL IN THE FOR 2D SPACE

The EKF-based approaches presented in Chapter 2 and Chapter 3 assumes that the transmitting agent is static and consequently requires the communicating robots to take the scanning in an alternating fashion for the convergence of the estimator in bi-directional alignment control. This approach results in the stopping of the agents for half of their time, and hence the effective tracking efficiency is reduced to half. In a desirable method, the agents do not need to stop and should continuously and simultaneously track each other.

In this chapter, an alternative approach is explored for the 2D setting that allows simultaneous, bi-directional alignment control for both parties. The problem is formulated in a discrete-time dynamical system setting. Each agent seeks to maximize its own output (measurement) function that depends on the states of both agents, and furthermore, the agents simultaneously make their moves. The output functions considered here are non-conflicting; the optimization of one output function helps optimize the other. However, the constraints of no communication between the agents, no access to the states, and parallel actions pose significant challenges. A computationally efficient and model-free output feedback control algorithm meeting all these constraints is proposed. In particular, we establish that, when the output functions satisfy certain conditions, the proposed control procedure guarantees that, in a finite number of steps, the system reaches a limiting set that contains the global optimum of size proportional to the step size. Simulation results demonstrate the efficacy of the approach and establish its superiority over two competing approaches, namely: extremum-seeking control and extended Kalman filter, in terms of convergence speed and robustness to disturbance. Experimental results on a setup involving two robots further validate the efficacy and quantify the proposed approach's performance.

The organization of this chapter is as follows. In Section 4.1, the hardware description of the LED-based communication system is presented, followed by mathematical modeling of the setup. Section 4.2 presents the proposed control scheme along with the key results followed by their mathematical analysis in Section 4.3. The simulation results are presented in Section 4.4, and experimental setup and results are discussed in Section 4.5. Finally, Section 4.6 provides a summary of the chapter.

4.1 System Setup and Problem Formulation

4.1.1 System Setup and Modeling

Consider two agents (e.g., robots) in a planar environment, as illustrated in Figure 4.1. The line joining the robots is the LOS. The distance between the robots is d. The optical axis of the transceiver of robot $\mathbf{R}_i$ makes angle θ_i with the LOS line, where $i \in \{1, 2\}, \theta_i \in (-\pi, \pi]$. In this work, we assume that the positions of the agents are fixed, and they can only change their angle θ_i.

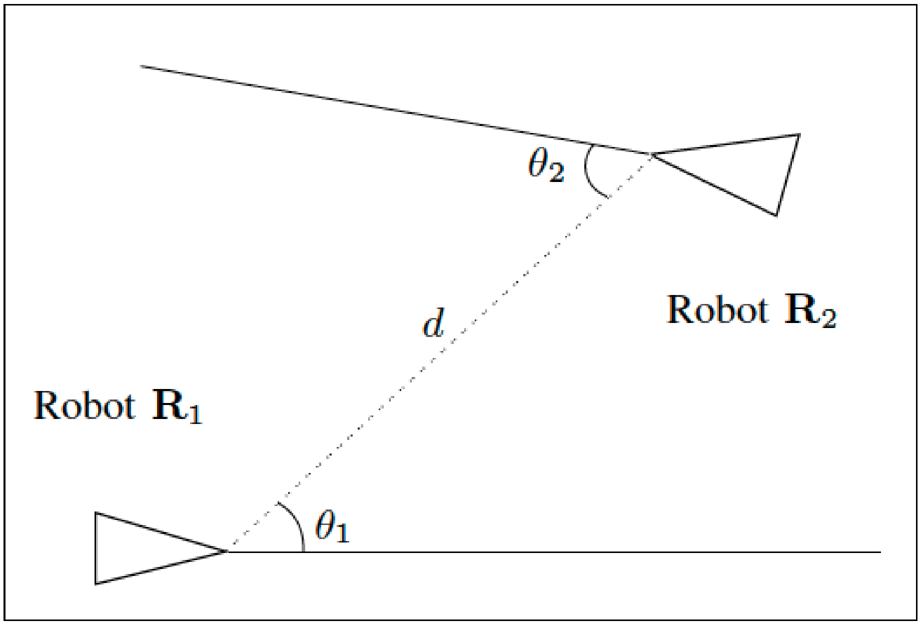


Figure 4.1: Two agents seeking to establish LOS in a 2D scenario.

Figure 4.2 shows the transceiver hardware setup for LED-based free-space optical communication, which has been shown in Figure 3.1, where it was mounted on a mobile robot. Each transceiver has two devices, a photo-diode with a lens and a light-emitting-diode (LED) with a lens, where the lenses are used to achieve desired collection and dispersion of the light, respectively. For the purpose of mathematical modeling, the agents are considered as points, and the optical axes of the LED and the photo-diode are assumed to be aligned for each agent. Moreover, the transceiver is mounted on a rotating platform, which enables the adjustment of transceiver orientation.

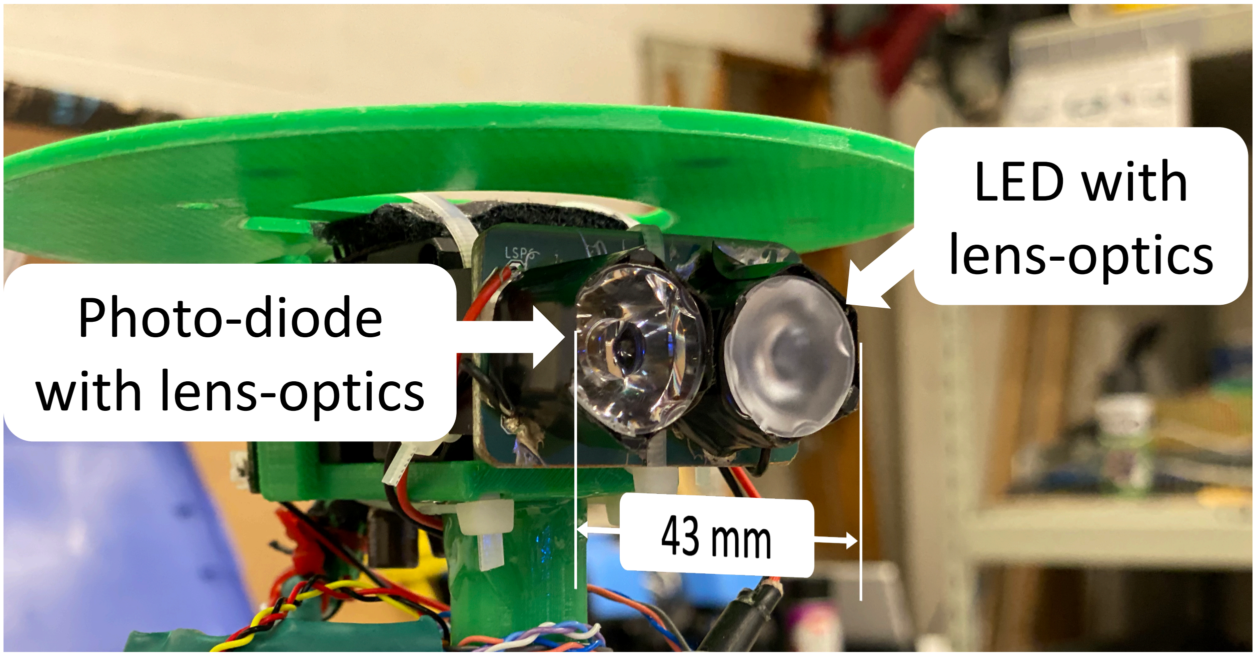


Figure 4.2: Illustration of hardware components of the transceiver for LED communication.

The light signal strength model adopted here, is derived from Section 2.2.1 with minor adjustments to suit the bi-directional experimental setup considered in this chapter. With Eqs. (2.2) and (2.3), the signal strength of the light transmitted from agent $\mathbf{R}_2$, as measured by the photo-diode of agent $\mathbf{R}_1$ can be obtained as

$$V_d = C\frac{e^{-cd}}{d^2}f(\theta_2)g(\theta_1), \tag{4.1}$$

where C is a constant of proportionality, c is the attenuation coefficient of the transmission medium (typically air or water). The function $f(\theta_2)$ represents the angular intensity distribution of $\mathbf{R}_2$'s LED (same as I_γ in Eq. (2.2)), while, $g\ (\theta_1)$ represents the angular sensitivity of the photo-diode of $\mathbf{R}_1$. An expression analogous to (4.1) holds for the light intensity measured by $\mathbf{R}_2$. It is to be noted that $f(\cdot)$ and $g(\cdot)$ depend on multiple factors: optics of the combination of the lens and the LED/photo-diode, the refraction between surrounding medium and the lens interface, and optical characteristics of the medium itself. Therefore, we characterize $f(\cdot)$ and $g(\cdot)$ by collecting intensity measurements (from the experimental setup discussed in Section 4.5) over a range of heading angles and then fitting a Gaussian curve on the collected data. Figure 4.3 shows the data collected on our setup and the corresponding Gaussian fitting functions.

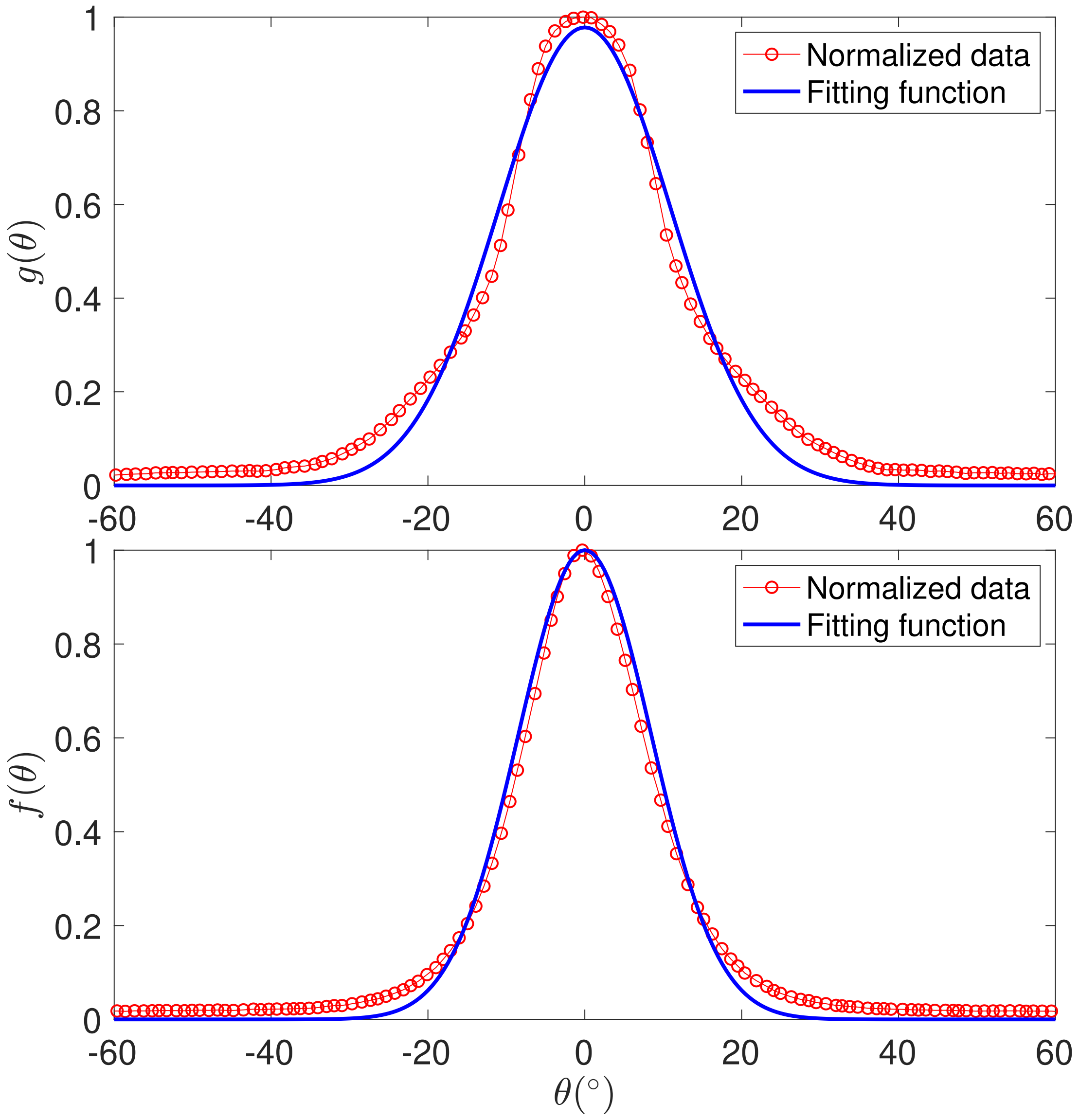


Figure 4.3: Illustration of the Gaussian approximation of the fitting functions of photo-diode sensitivity curve $g(\cdot)$ and LED intensity curve $f(\cdot)$.

4.1.2 State-space problem formulation

Let the state variables x_1 and x_2 be the angles θ_1 and θ_2, respectively. When the agents' positions are fixed, the distance d is constant and can be merged into a new proportional constant C_p, resulting in the measurement functions

$$\mathbf{y}_k \triangleq \begin{bmatrix} y_{1,k} \\ y_{2,k} \end{bmatrix} = \begin{bmatrix} C_p f(x_{2,k}) g(x_{1,k}) \\ C_p f(x_{1,k}) g(x_{2,k}) \end{bmatrix} = \begin{bmatrix} C_p e^{-\left(\frac{x_{1,k}^2}{a^2} + \frac{x_{2,k}^2}{b^2}\right)} \\ C_p e^{-\left(\frac{x_{1,k}^2}{b^2} + \frac{x_{2,k}^2}{a^2}\right)} \end{bmatrix}. \tag{4.2}$$

where $y_{1,k}$ and $y_{2,k}$ denote the light intensities measured by agent $\mathbf{R}_1$ and $\mathbf{R}_2$, respectively, at time instant k. We consider the following dynamics for the states:

$$x_{i,k+1} = x_{i,k} + u_{i,k}, \text{ for } i \in \{1, 2\}, \tag{4.3}$$

where $u_{i,k}$ represents the control of the i-th agent and takes the form,

$$u_{i,k} = U(y_{i,k}, ..., y_{i,0}, u_{i,k-1}, ..., u_{i,0}) \tag{4.4}$$

$U : \mathbb{R}^{2k+1} \longrightarrow \mathbb{R}$, and $|u_{i,k}| \leq \delta$ for some given $\delta > 0$. Eq. (4.4) captures the constraint of no communication between agents since the control term of an agent i can *only* depend on its own history of measurements and control inputs.

4.1.3 Generalized Problem Formulation

In this subsection, we pose a generalized version of the system, where the measurement functions include but are not limited to the class of Gaussian functions. Consider a discrete-time two-agent dynamic system where the agents' measurement functions are *non-conflicting*; i.e., they have a common global maximum, and at every point in the domain (except at the global maximum), there always exists a direction where both of the measurement functions have a positive gradient. The constraints are that 1) there is no communication between the agents, 2) neither agent has access to any of the states, including its own, and 3) the agents move simultaneously at every time step. The assumption of simultaneous actions is justified because the sampling time ΔT is the same for both agents, and each agent makes a move in each time-step.

We denote the state of the i-th agent by $x_i \in (-\pi, \pi]$, for $i \in \{1, 2\}$. Each agent has a smooth measurement function $h_i(\mathbf{x}_k)$, $\mathbf{x}_k = \left[x_{1,k}, x_{2,k}\right]^T$. Therefore $\mathbf{x}_k \in \mathbf{\Pi}$, where $\mathbf{\Pi} = (-\pi, \pi] \times (-\pi, \pi]$. The subscript k represents the k-th time instant and the generalized output now becomes

$$\mathbf{y}_k = [h_1(\mathbf{x}_k),\ h_2(\mathbf{x}_k)]^T. \tag{4.5}$$

Note that in the context of Eq. (4.2), $h_i(\mathbf{x}_k) = C_p f(x_{3-i,k}) g(x_{i,k})$, but the algorithm proposed in this work applies to cases with the measurement functions of a general form h_i that satisfy the following assumptions.

Assumption 4.1.1 For all $\mathbf{x}_k \in \mathbb{R}^2 \setminus \{(0, 0)\}$:

1. $\mathbf{x}_k \cdot \nabla h_i(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_k} < 0$ for $i \in \{1, 2\}$, (4.6)

2. $\nabla h_1(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_k} \cdot \nabla h_2(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_k} > 0$, (4.7)

3. Both h_1 and h_2 have strictly convex super-level sets [1].

Eq. (4.6) implies that both measurement functions h_1 and h_2 are non-conflicting as they both have a unique common global maximum at $(0, 0)$ and the functions smoothly decay away from $(0, 0)$. Eq. (4.7) ensures that at every point in $\mathbb{R}^2 \setminus \{(0, 0)\}$, the gradients of the measurement functions make an acute angle with each other, which results in the existence of a common direction of the positive gradient. The convex super-level sets assumption on h_1 and h_2 results into a corresponding uni-modal function on any line drawn in $\mathbb{R}^2$, which we shall see in Section 4.3. Furthermore, the convex super-level set assumption is weaker than the stronger requirement of functions being convex or concave, which makes the formulation less restrictive and applicable to practical scenarios. The goal is to drive the system state to the origin, using only the knowledge of local measurement.

[1] A set $L_c^+(h, c) = \{(x_1, x_2) | h(x_1, x_2) \geq c\}$ is called a super-level set of the function h.

It can be verified that a set of any two quadratic measurement functions with an optimum at the origin would satisfy the Assumption 4.1.1. There are several physical quantities that satisfy the inverse-square law. The examples include the intensity of sound, light, electrical, and magnetic radiation emitting from a point source. In a two-dimensional setting, this inverse-square law corresponds to a quadratic function. Hence, Assumption 4.1.1 allows a large class of functions that represent physical quantities. Furthermore, a Gaussian function can be mapped to a quadratic function by applying a natural logarithm. The aforementioned Gaussian light-intensity model in Eq. (4.2) can also be transformed into a quadratic function. Next, we propose a control law for the aforementioned system complying with the requirements in Eq. (4.4).

4.2 Main Results

In this section, we first provide a control law satisfying the requirement in Eq. (4.4). Then we define geometric terms which would be essential for upcoming analysis. Afterward, we provide results that the proposed control law can drive the states of the system to an appropriately characterized limiting set in a finite number of steps. Later, we provide stronger results for the model of the physical setup, where the functions h_1 and h_2 are Gaussians.

4.2.1 Proposed Control Law

Consider the following control law:

$$\begin{bmatrix} u_{1,k} \\ u_{2,k} \end{bmatrix} = \begin{bmatrix} \mathrm{sgn}\big(y_{1,k} - y_{1,k-1}\big) u_{1,k-1} \\ \mathrm{sgn}\big(y_{2,k} - y_{2,k-1}\big) u_{2,k-1} \end{bmatrix}, \tag{4.8}$$

$$\text{where, } \mathrm{sgn}(p) = \begin{cases} +1, & \text{if } p \geq 0, \\ -1, & \text{otherwise,} \end{cases} \tag{4.9}$$

which essentially means that an agent takes an action in the direction of increasing measurements, and when it observes a reduction in its measurement, it switches to an action in the opposite direction. The initial direction ($\mathbf{u}_0$) is chosen at random from set $\mathbf{S}_u$:

$\mathbf{S}_u \equiv \{[+\delta, +\delta]^T, [+\delta, -\delta]^T, [-\delta, +\delta]^T, [-\delta, -\delta]^T\}$

The control law performs only one comparison to generate the control for each agent, which makes it highly efficient and ideal for real-time onboard applications. Next, we introduce some geometric terms which would be essential to defining the limiting set and conducting the upcoming analysis.

4.2.2 Definition of Essential Geometric Terms

Consider a line $\rho_{\mathbf{x},\,\mathbf{u}}$ passing through a point $\mathbf{x}$ in a direction $\mathbf{u}$. All points on this line can be parametrized as $\rho_{\mathbf{x},\,\mathbf{u}}(\alpha) \equiv \mathbf{x} + \alpha\hat{\mathbf{u}}$, where, $\alpha \in \mathbf{R}$ and $\hat{\mathbf{u}} = \frac{\mathbf{u}}{\sqrt{2}\delta}$.

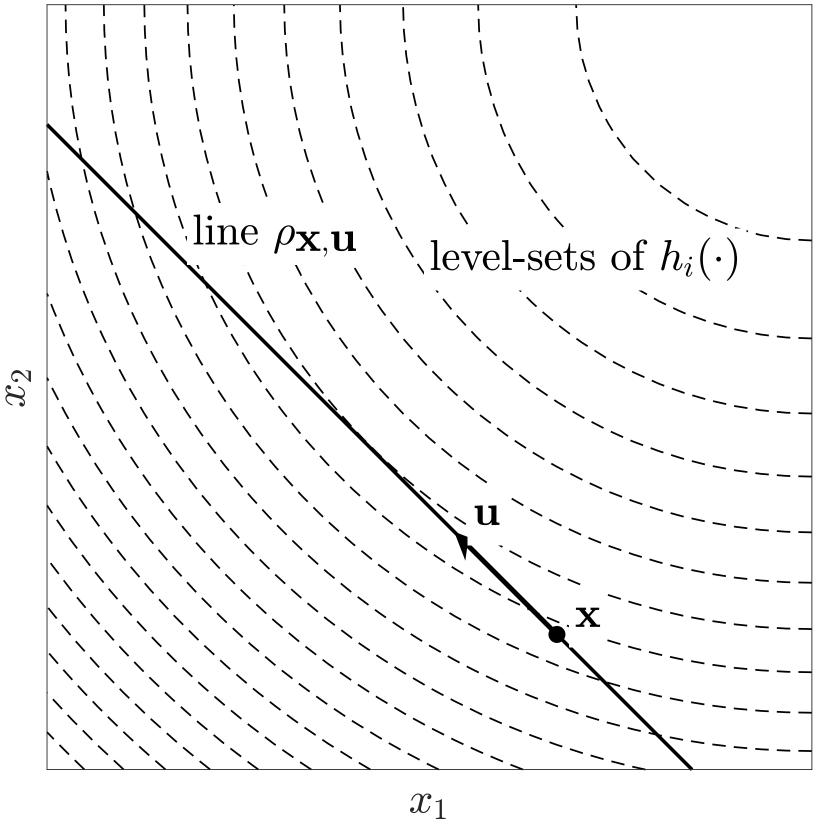

line $\rho_{\mathbf{x},\mathbf{u}}$
level-sets of $h_i(\cdot)$
$\mathbf{u}$
$\mathbf{x}$
x_2
x_1

Figure 4.4: Line $\rho_{\mathbf{x},\,\mathbf{u}}$ with super-level sets of $h_i(\cdot)$.

Definition 4.2.1 (*Transition Interval*) A transition interval on a line $\rho_{\mathbf{x},\,\mathbf{u}}$, corresponding to the agent i, is defined as the set

$$\mathbf{t}_i(\rho_{\mathbf{x},\,\mathbf{u}}) \coloneqq \{\bar{\mathbf{x}} \mid \bar{\mathbf{x}} = \beta \mathbf{x}^l + (1-\beta)\mathbf{x}^m, 0 \le \beta \le 1\}, \tag{4.10}$$

where $\mathbf{x}^l$ and $\mathbf{x}^m$ are the points of intersection of the line $\rho_{\mathbf{x},\,\mathbf{u}}$ and a level set $L_c^+(h, h(\mathbf{x}^l))$, and satisfy:

1. $h_i(\mathbf{x}^l) = h_i(\mathbf{x}^m)$, and (4.11)

2. $\|\mathbf{x}^m - \mathbf{x}^l\| = \sqrt{2}\delta.$ (4.12)

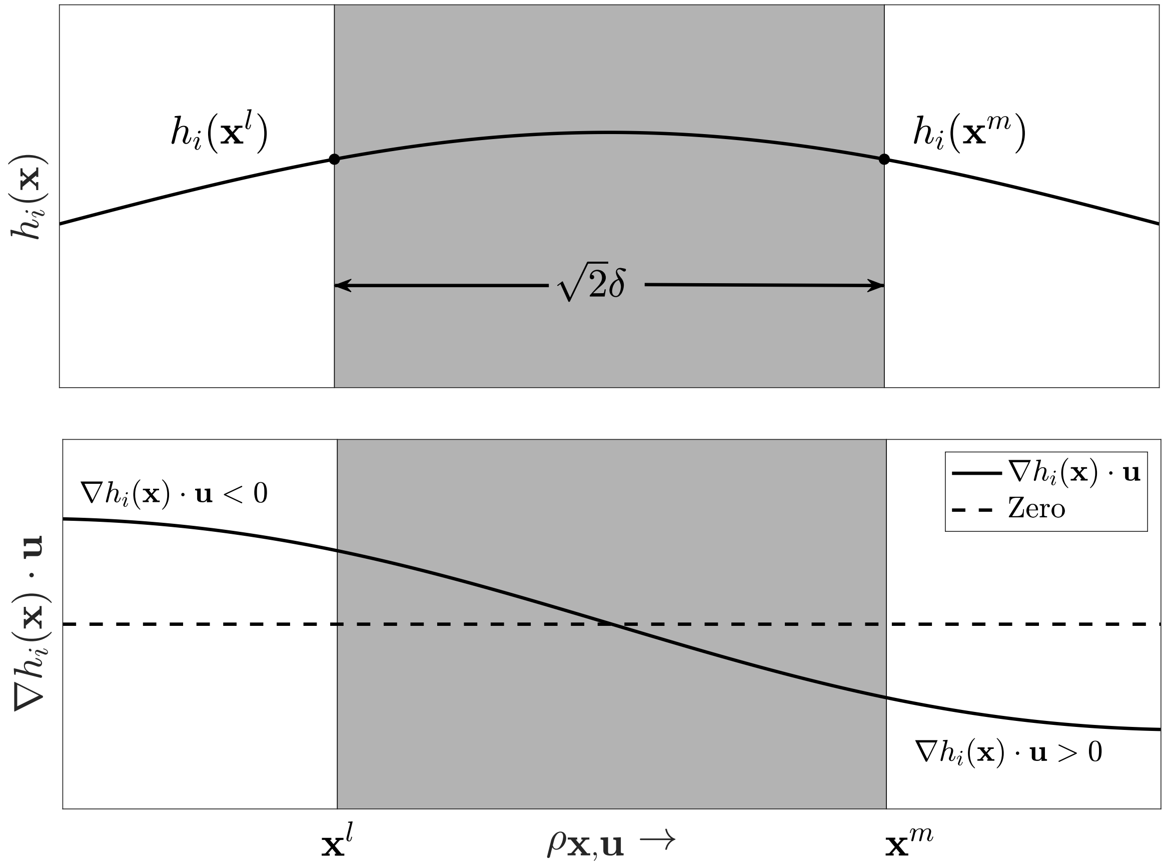


Figure 4.5: Illustration of the transition interval in terms of measurement function h_i and its gradient along the line $\rho_{\mathbf{x},\,\mathbf{u}}$.

Figure 4.4 illustrates a line $\rho_{\mathbf{x},\,\mathbf{u}}$ along with the super-level sets of measurement function h_i and Figure 4.5 illustrates the transition interval on the line $\rho_{\mathbf{x},\,\mathbf{u}}$. From Assumption 4.1.1, because of the strictly convex super-level set assumption on h_i, it can be inferred that the function h_i demonstrates a uni-modal behavior along the line $\rho_{\mathbf{x},\,\mathbf{u}}$ and therefore, the pair ($\mathbf{x}^l$ and $\mathbf{x}^m$) is unique under the assumption.

Definition 4.2.2 (*Transition region*) A transition region of an agent i for a control $\mathbf{u}$ is a continuum of transition intervals $\mathbf{t}_i(\rho_{\mathbf{x},\mathbf{u}})$ and is formally defined as the set

$$\mathbf{T}_i(\mathbf{u}) := \{\bar{\mathbf{x}} \mid \bar{\mathbf{x}} \in \mathbf{t}_i(\rho_{\bar{\mathbf{x}},\mathbf{u}})\} \tag{4.13}$$

Definition 4.2.3 (*Intersecting transition region*) An intersecting transition region $\mathbf{D}(\mathbf{u})$ for a control $\mathbf{u}$ is defined as the intersection of $\mathbf{T}_1(\mathbf{u})$ and $\mathbf{T}_2(\mathbf{u})$:

$$\mathbf{D}(\mathbf{u}) := \mathbf{T}_1(\mathbf{u}) \cap \mathbf{T}_2(\mathbf{u}) \tag{4.14}$$

Remark 1 *The origin is the global optimum, and hence it belongs to both bands of the transition region; consequently, the set* $\mathbf{D}(\mathbf{u})$ *is always non-empty.*

Figure 4.6 illustrates the transition regions $\mathbf{T}_1(\mathbf{u}^2)$ and $\mathbf{T}_2(\mathbf{u}^2)$ for a control direction $\mathbf{u}^2$, and the corresponding intersecting transition region $\mathbf{D}(\mathbf{u}^2)$.

The following subsection summarizes our results.

4.2.3 Key Results

Theorem 4.2.1 (*Main Result*) *From any initial state* $\boldsymbol{x}_0 \in \mathbf{\Pi}$*, the control law defined in Eq. (4.8) drives the system defined by Eqs.* (4.3) *and* (4.5) *to a set* $\mathbf{D}_\cup$ *in a finite number of steps* K *satisfying*

$$K \leq 3\left(\frac{\|\boldsymbol{x}_0\|_2}{\delta} + 2\right) \tag{4.15}$$

The operator $\|\cdot\|_2$ *denotes the (*Euclidean *norm) of a vector and set* $\mathbf{D}_\cup$ *is defined as*

$$\mathbf{D}_\cup := \mathbf{D}_0 \cup \bigcup_{\mathbf{u} \in \mathbf{S}_u} \mathbf{D}(\mathbf{u}) \tag{4.16}$$

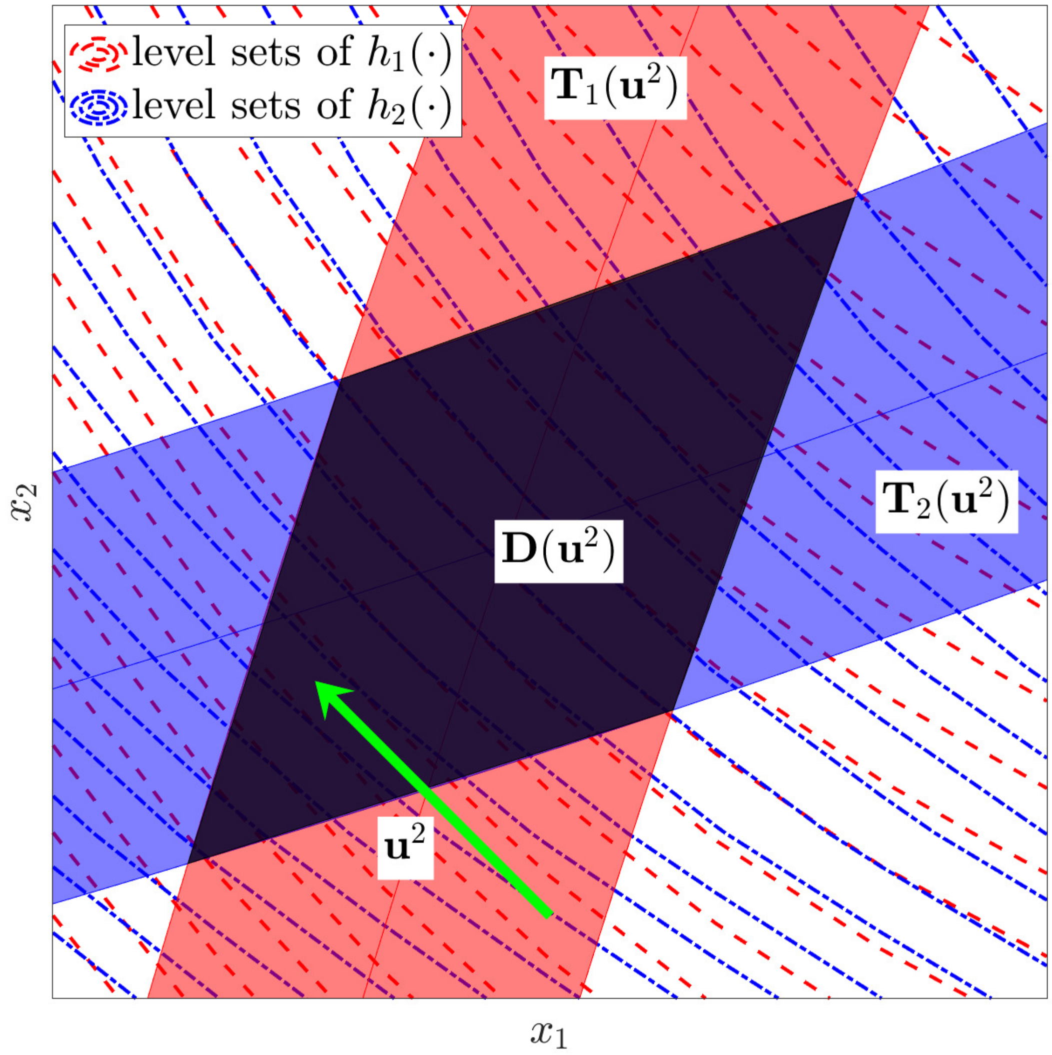

level sets of $h_1(\cdot)$
level sets of $h_2(\cdot)$
$\mathbf{T}_1(\mathbf{u}^2)$
$\mathbf{T}_2(\mathbf{u}^2)$
$\mathbf{D}(\mathbf{u}^2)$
$\mathbf{u}^2$
x_1
x_2

Figure 4.6: Illustration of the intersecting transition region $\mathbf{D}(\mathbf{u}^2)$ for the control direction $\mathbf{u}^2$.

with

$$\mathbf{D}_0 \coloneqq \left\{ \boldsymbol{x} : \|\boldsymbol{x}\|_1 \leq \frac{3\delta}{2} \right\}. \tag{4.17}$$

*The operator $\|\cdot\|_1$ denotes the ℓ_1-norm (*Manhattan *norm) of a vector.*

□

Remark 2 *If the system is already inside the set $\mathbf{D}_\cup$ and if it leaves the set $\mathbf{D}_\cup$, then Theorem 4.2.1 applies to the new state as the initial condition, and the system returns to $\mathbf{D}_\cup$ in a finite number of steps.*

Corollary 1 (*Special Cases*) *For a choice of h_1 and h_2 such that $\mathbf{D}_\cup$ is compact and contiguous, the control law drives the system to an $O(\delta)$-neighborhood of the origin.*

In Section 4.1, the functions h_1 and h_2 corresponding to measurement functions y_1 and y_2 in (4.2) for our setup, are characterized by Gaussians, in which case $\mathbf{D}_\cup$ becomes a compact, contiguous set around the origin (the illustration is provided in Section 4.3, Figure 4.13). This set can be contained inside a neighborhood around the origin with size proportional to δ. The following theorem provides stronger results for the model of the aforementioned physical setup.

Theorem 4.2.2 (*Results for Physical Setup*) *From any permissible initial condition $\boldsymbol{x}_0$, under the assumption that $a > b$, the control law defined in Eq. (4.8) drives the states of the system defined in* (4.3) *and* (4.2) *to the set $\mathbf{D}_\cup$ in a finite number of steps K, where*

$$K \leq \frac{\sqrt{2}\|\boldsymbol{x}_0\|_1}{\delta} + 4(\left\lceil \frac{\ln(\frac{2\|\boldsymbol{x}_0\|_1}{3\delta})}{\ln(\frac{a^2+b^2}{a^2-b^2})} \right\rceil + 1). \tag{4.18}$$

The operator $\|\cdot\|_1$ denotes the ℓ_1-norm of a vector.

□

It is worth noting that the bound on the number of steps K in (4.18) is significantly tighter than the bound provided in the equation (4.15). The proof of the Theorem 4.2.2 is provided in [47], where properties of Gaussians are used to derive the stricter bound. The next section is dedicated to the proof of Theorem 4.2.1.

4.3 Proof of Theorem 4.2.1

In this section, we discuss the proof of the Theorem 4.1. Fig 4.7 illustrates the roadmap of the proof. We first discuss the local behavior of the trajectory of the system, where the next few steps are determined on the basis of the location of the present states. Then we discuss the bounds on the number of steps K. Then we provide a discussion on the nature of limiting configuration set $\mathbf{D}_\cup$.

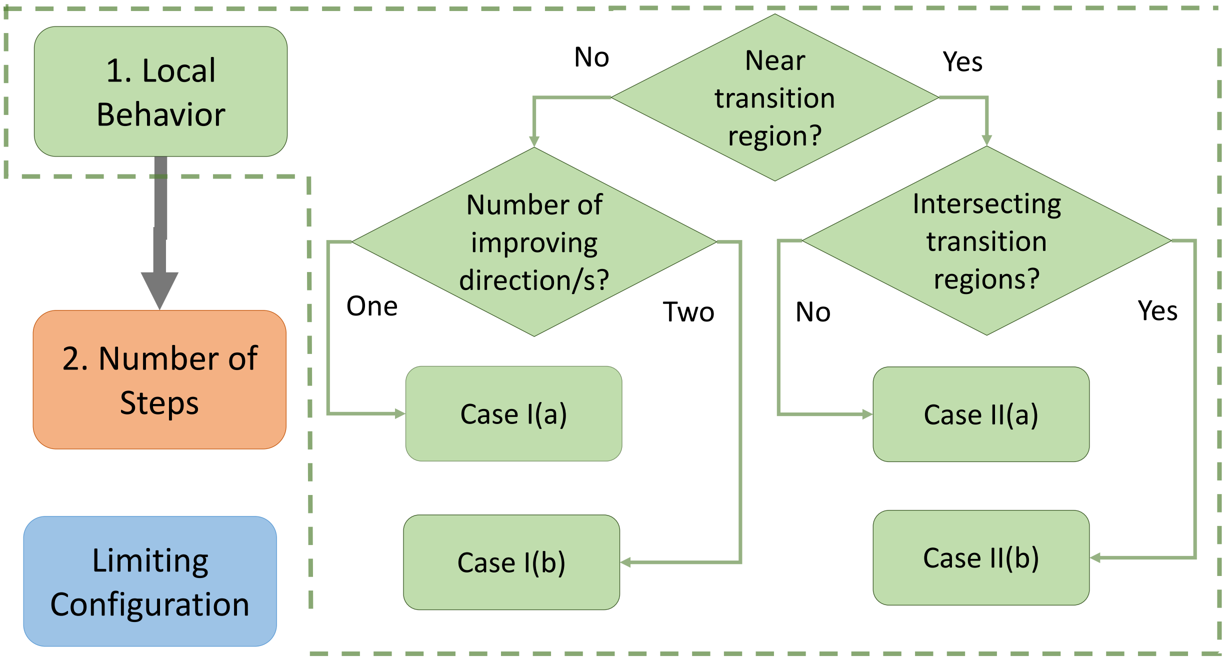


Figure 4.7: Roadmap of the proof of Theorem 4.2.1.

4.3.1 Local Behavior

From the mean value theorem applied to function h_i, for $i \in \{1, 2\}$, for any $k \geq 0$, there exists a point $\mathbf{x}^M$ on the line segment joining $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$ ($\mathbf{x}^M = \alpha \mathbf{x}_k + (1-\alpha)\,\mathbf{x}_{k+1} \mid \alpha \in [0,1]$) such that,

$$h_i\left(\mathbf{x}_{k+1}\right) - h_i\left(\mathbf{x}_k\right) = \nabla h_i \left(\mathbf{x}^M\right) \cdot \mathbf{u}_k. \tag{4.19}$$

Now there arise two cases:

Case I: Both $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$ are outside of any of the transition intervals: ($\mathbf{t}_1(\rho_{\mathbf{x}_k,\mathbf{u}_k})$ or $\mathbf{t}_2(\rho_{\mathbf{x}_k,\mathbf{u}_k})$). This results in the following condition:

$\forall\, \alpha \in [0,1]\,, \nabla h_i\left(\alpha \mathbf{x}_k + (1-\alpha)\,\mathbf{x}_{k+1}\right) \neq 0.$

The condition essentially means that the sign of the dot product of the gradient of any of the measurement functions and the control direction, remains same at any intermediate point $\mathbf{x}^M$ along the control direction. Using (4.19),

$$\begin{aligned}\operatorname{sgn}(h_i(\mathbf{x}_{k+1}) - h_i(\mathbf{x}_k)) &= \operatorname{sgn}\left(\nabla h_i\left(\mathbf{x}^M\right)\cdot \mathbf{u}_k\right)\\ &= \operatorname{sgn}(\nabla h_i(\mathbf{x}_k)\cdot \mathbf{u}_k).\end{aligned}$$

Now the control law defined in (4.8) can be re-written as

$$\begin{bmatrix} u_{1,k}\\ u_{2,k}\end{bmatrix} = \begin{bmatrix}\operatorname{sgn}(\nabla h_1(\mathbf{x}_k)\cdot \mathbf{u}_k)\\ \operatorname{sgn}(\nabla h_2(\mathbf{x}_k)\cdot \mathbf{u}_k)\end{bmatrix} \odot \mathbf{u}_k, \tag{4.20}$$

where the operator ($\odot$) represents the Hadamard product of two vectors.

Figure 4.8 illustrates a possible configuration of the gradients of the two reward functions, where θ_i denotes the angle of gradient of h_i, with respect to the positive x_1 axis. From the figure and (4.7), it can be deduced that out of the four possible control directions, there would be at least one direction, which would make an acute angle with both of the gradients. From this point onward, such control direction(s) is (are) referred to as *improving* direction(s). For example, $\mathbf{u}^1$ is the improving direction in Figure 4.8.

Furthermore, since our problem is two-dimensional, all the variables are represented as complex numbers to simplify the analysis. As the agents have the same control magnitude δ, the overall control term at any time instant k can be represented as:

$$\begin{aligned}\mathbf{u}_k &= \delta e^{\frac{j\pi}{2}\left(n_k - \frac{1}{2}\right)}, \text{ for some } n_k \in \{1,2,3,4\}\\ &= \delta\left(\cos\frac{\pi}{2}\left(n_k - \frac{1}{2}\right) + j\sin\frac{\pi}{2}\left(n_k - \frac{1}{2}\right)\right),\end{aligned} \tag{4.21}$$

where n_k represents the quadrant of the complex representation of $\mathbf{u}_k$. Now with (4.21), Eq. (4.20) results in

$$\begin{aligned}\mathbf{u}_{k+1} = \delta\operatorname{sgn}\big(\cos\left(\Delta\theta_{1,k}\right)\big)\cos\left(\frac{\pi}{2}\left(n_k - \frac{1}{2}\right)\right)\\ + j\delta\operatorname{sgn}\big(\cos\left(\Delta\theta_{2,k}\right)\big)\sin\left(\frac{\pi}{2}\left(n_k - \frac{1}{2}\right)\right)\\ = \operatorname{sgn}\big(\cos\left(\Delta\theta_{1,k}\right)\big)u_{1,k} + j\delta\operatorname{sgn}\big(\cos\left(\Delta\theta_{2,k}\right)\big)u_{2,k},\end{aligned} \tag{4.22}$$

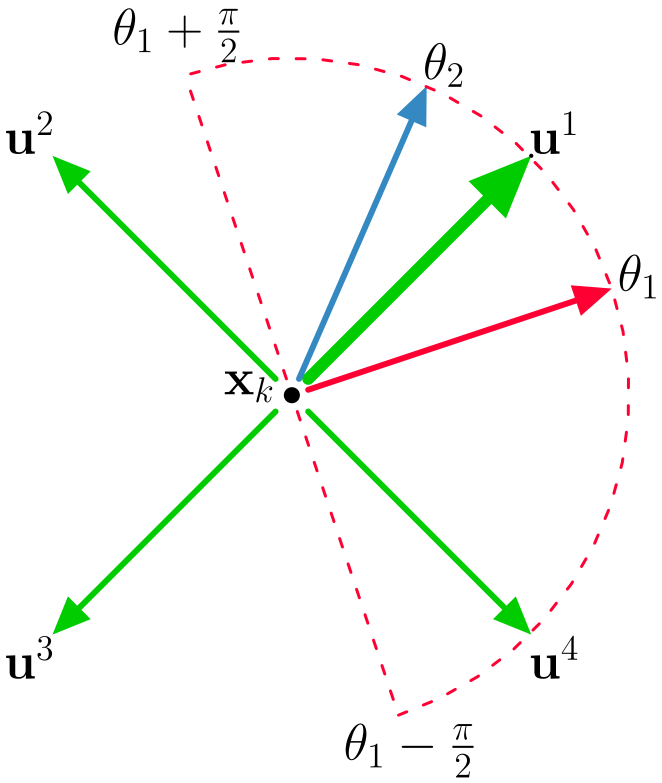


Figure 4.8: Illustration of the gradients of the two measurement functions and the four control directions. θ_1 and θ_2 represent the angles of the gradients of the measurement functions h_1 and h_2, respectively, with respect to the positive x_1 axis, evaluated at point $\mathbf{x}_k$. The improving control direction $\mathbf{u}^1$ is denoted by a thicker arrow.

where $\Delta\theta_{i,k} := \theta_{i,k} - \frac{\pi}{2}(n_k - \frac{1}{2})$ is the angle between the current control direction with the corresponding gradient direction. Here it can be observed that when the gradient of the measurement function of an agent makes an obtuse angle with the improving control direction, the agent switches its control direction at the subsequent time step. Table 4.1 summarizes the relation between the two successive control terms.

All configurations of the gradient directions and the control directions can be captured in the following two subcases:

Case I(a): When there is only one improving direction, and it is in between the two gradient directions. Figure 4.8 illustrates this case when $\mathbf{u}^1$ is the only improving direction while Figure 4.9 shows two configurations of this case. Without loss of generality, all the configurations become equivalent to one out of the two illustrated configurations. For ease of exposition, these

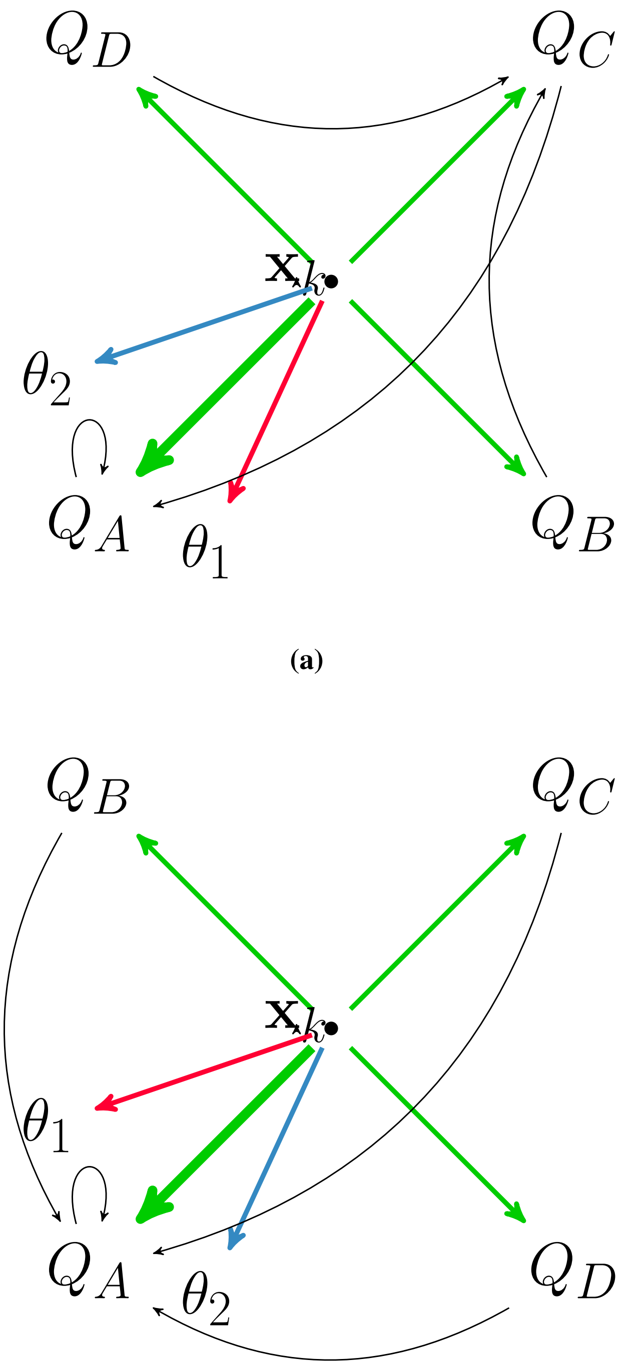
Table 4.1: Summary of the outcome $\mathbf{u}_{k+1}$ based on $\mathbf{u}_k$ and gradient angles. The term $\mathbf{u}_k^*$ denotes the complex conjugate of $\mathbf{u}_k$.

$\lvert\Delta\theta_{2,k}\rvert$ $\lvert\Delta\theta_{1,k}\rvert$	$\leq \frac{\pi}{2}$	$> \frac{\pi}{2}$
$\leq \frac{\pi}{2}$	$\mathbf{u}_k$	$\mathbf{u}_k^*$
$> \frac{\pi}{2}$	$-\mathbf{u}_k^*$	$-\mathbf{u}_k$

configurations have been represented using a finite state machine in the remainder of this proof, where each configuration is represented as a *control state* which is deduced from Table 4.1. The finite state machine representation is also overlaid in Figure 4.9, to illustrate the transition from any control state to the state of improving control direction. The control state Q_A represents the improving direction, Q_C represents the direction opposite to the improving direction, Q_B is the direction closer to θ_1, and Q_D is closer to θ_2.

Consider the first configuration in Figure 4.9(a), where the state Q_A always transitions to itself. For state Q_C, since both of the agents' measurement function gradients make obtuse angles with the control direction, it always transitions to the state Q_A. Next, for state Q_B, θ_2 makes an obtuse angle, and hence agent 2 flips the control direction. This results in transitioning to state Q_C, which transitions to state Q_A in the next step. Similarly, Q_D transitions to Q_C and then to Q_A. In the second configuration, shown in Figure 4.9(b), for state Q_B, the flip in the control direction by agent 2 results in a transition to the state Q_A. Similarly, Q_D also transitions to Q_A in one step. Hence, in a maximum of two steps, the system achieves the improving direction.

Case I(b): There are two improving directions, and both of the gradient directions form acute angles with the two improving directions. Figure 4.10 shows the configuration along with the corresponding finite state machine. The improving directions are labeled as Q_{A1} and Q_{A2}, and the corresponding opposite directions are Q_{C1} and Q_{C2}. Here, if the system is in Q_{C1} or Q_{C2}, both agents make obtuse angles with the current control direction. Hence, they switch simultaneously,



(a)

(b)

Figure 4.9: Two configurations for the case when the improving direction is between the reward function gradient directions. The corresponding state machine diagram is overlaid, where Q_A represents the improving control direction state. Similarly, Q_B, Q_C and Q_D are defined relatively w.r.t. control directions and the gradient directions.

and it takes only one step to get into an improving direction Q_{A1} (or Q_{A2} depending on the previous state).

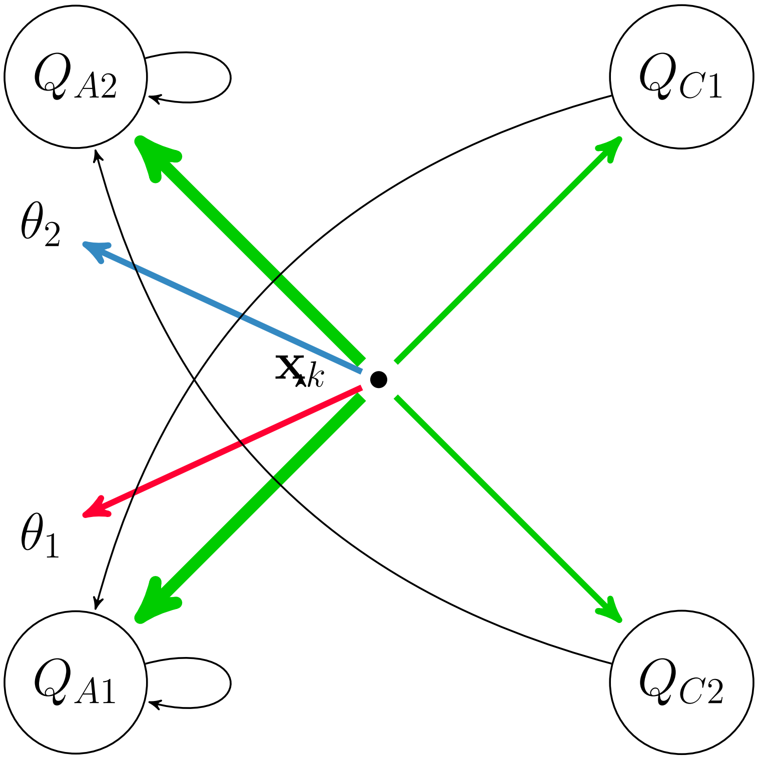
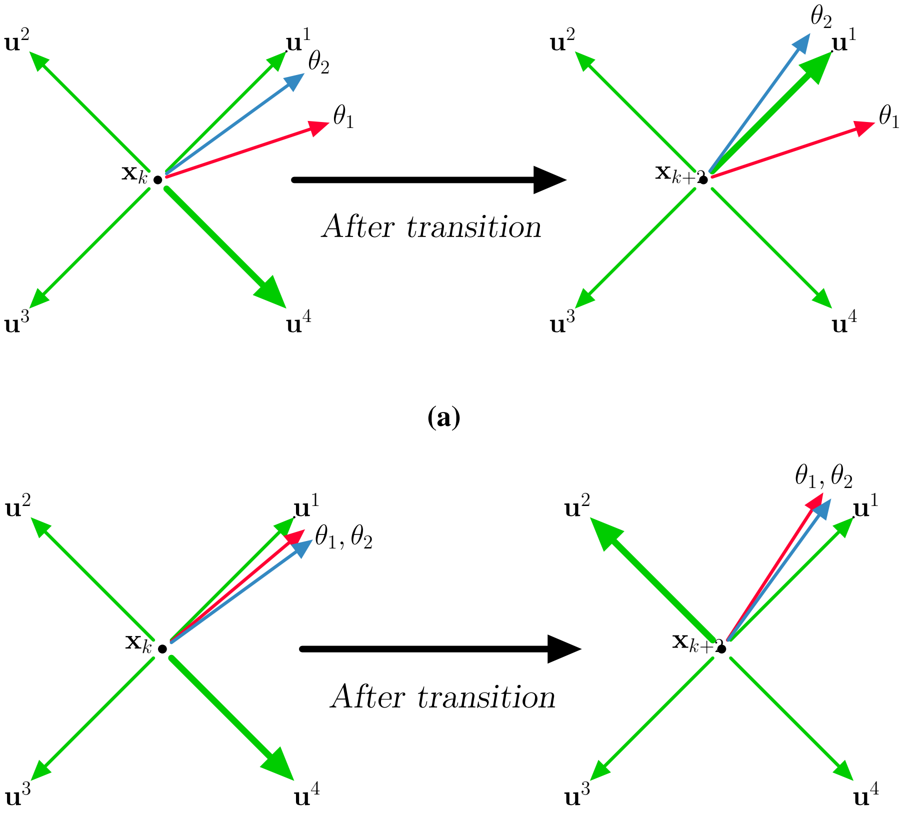


Figure 4.10: Configuration where both of the gradients are in between the improving directions.

Case II: Either of $\mathbf{x}_k$ or $\mathbf{x}_{k+1}$ lies inside any of the transition intervals. Here, without loss of generality, let $\mathbf{x}_{k+1}$ lie inside a transition interval. It is to be noticed that $\mathbf{x}_k$, $\mathbf{x}_{k+1}$ and $\mathbf{x}_{k+2}$ lie on a straight line if the system is inside a transition interval at $\mathbf{x}_{k+1}$. There arise two sub-cases in the scenario. Figure 4.11 illustrates two sub-cases of the effect of transition in the configuration of gradients and the improving direction.

Case II(a): $\mathbf{x}_{k+1} \notin \mathbf{D}(\mathbf{u}), \forall\, \mathbf{u} \in \mathbf{S}_u$. In this scenario the state $\mathbf{x}_{k+1}$ lies inside a transition region but it is outside any intersecting transition region. In Figure 4.11(a), initially the system was moving in the improving direction $\mathbf{u}^4$, at step k, according to the case I(b). It goes into a transition interval (without loss of generality, $\mathbf{t}_2(\rho_{\mathbf{x}_k,\mathbf{u}_k})$ is considered in the illustration) in the step $k+1$ and goes out in the step $k+2$. During the transition from $\mathbf{x}_k$ to $\mathbf{x}_{k+2}$, the gradient direction of agent 2 crosses a control direction $\mathbf{u}^1$ and the overall gradient configuration changes to case I(a). Therefore, $\mathbf{u}^2$ becomes the new improving direction. Additionally, depending on which of the configurations described in case I(a) is applicable, the system takes at most two steps to achieve the new improving control direction.



(a)

(b)

Figure 4.11: Illustration of two sub-cases of the change in gradient configuration when a transition occurs between step k and $k+1$.

Case II(b): $\exists\, \mathbf{u} \in \mathbf{S}_u : \mathbf{x}_{k+1} \in \mathbf{D}(\mathbf{u})$. Here, the state $\mathbf{x}_{k+1}$ lies inside an intersecting transition region. Figure 4.11(b) illustrates that while passing through an intersecting transition region $\mathbf{D}(\mathbf{u}^1)$, both of the function gradients cross the control direction $\mathbf{u}^1$ together. This results in case I(b) after the transition, where the system switches its control direction to $\mathbf{u}^2$, which is opposite to its initial direction $\mathbf{u}^4$. In this situation, the proposed control law leads to a perpetual oscillatory movement around the intersecting transition region.

4.3.2 Calculation of path length and the number of steps

In the previous discussion, we have shown that after corrections at the initial instant and at the transition regions, the system continues to move in an improving direction which is one of the four control directions. Finally, the system reaches $\mathbf{D}_{\cup}$ (it either reaches an intersecting transition region or it reaches the set $\mathbf{D}_0$, which is defined in Eq. (4.17)). $\mathbf{D}_0$ accounts for the scenario when the system reaches the vicinity of origin and oscillates in a neighborhood around the origin without ever entering any intersecting transition region. Now we discuss the path-length and the number of steps the system takes from an arbitrary initial condition $\mathbf{x}_0$ to $\mathbf{D}_{\cup}$.

It can be deduced that the system always moves in two orthogonal directions and hence, the total path length l_{total} from any initial point $\mathbf{x}_0$ to $\mathbf{D}$ would be smaller than the *Manhattan distance* of the initial point from the origin along the control directions, i.e.,

$$l_{total} \leq \left(\frac{|x_{1,0} + x_{2,0}|}{\sqrt{2}} + \frac{|x_{1,0} - x_{2,0}|}{\sqrt{2}} \right). \tag{4.23}$$

A bound on the number of steps K_s can be calculated as:

$$\begin{aligned} K_s &\leq \left\lceil \frac{|x_{1,0} + x_{2,0}|}{2\delta} \right\rceil + \left\lceil \frac{|x_{1,0} - x_{2,0}|}{2\delta} \right\rceil \\ &\leq \frac{|x_{1,0} + x_{2,0}|}{2\delta} + \frac{|x_{1,0} - x_{2,0}|}{2\delta} + 2, \end{aligned}$$

where $\lceil \cdot \rceil$ represents the greatest integer function. Now using the right triangle inequality, where each side is smaller than the hypotenuse, we get

$$K_s \leq \left(\frac{|x_{1,0} + x_{2,0}|}{2\delta} + \frac{|x_{1,0} - x_{2,0}|}{2\delta} + 2 \right) \leq \frac{\|\mathbf{x}_0\|_2}{\delta} + 2.$$

Additionally, at each transition, there can be at most two additional steps for correction (as per the first configuration of Case I(a)). The total number of transitions depends on the functions h_1

and h_2; however, a loose upper bound on the number of transitions can be the total number of steps on the path. Hence, an upper bound on the total number of steps K is:

$$K \leq 3\left(\frac{\|\mathbf{x}_0\|_2}{\delta} + 2\right), \tag{4.24}$$

where the factor 3 accounts for the worst case that every step requires a transition. ■

4.3.3 Limiting Configuration

It is desirable that the set $\mathbf{D}_\cup$ be compact and contiguous. However, that is not true in general. Figure 4.12 illustrate a separate portion $\mathbf{D}(\mathbf{u}^2)$, which is a part of $\mathbf{D}_\cup$ but it is away from origin. A sample path p is shown that ends in oscillation between three points. The central point lie in set $\mathbf{D}(\mathbf{u}^2)$. The two green shaded diamond regions on the side of $\mathbf{D}(\mathbf{u}^2)$ illustrate the limiting configuration such that once the trajectory reaches $\mathbf{D}(\mathbf{u}^2)$, it stays within a region consisting of $\mathbf{D}(\mathbf{u}^2)$ and the two diamonds.

Figure 4.13 illustrates the set $\mathbf{D}_\cup$ as dark shaded region, for the case when the measurement functions h_1 and h_2 are Gaussians and characterize the physical setup discussed in Section 4.1. It can be seen that $\mathbf{D}_\cup$ is a compact and contiguous set around the origin. Sample paths $p1$ and $p2$ are shown to illustrate that once a trajectory reaches $\mathbf{D}_\cup$ it either stays in $\mathbf{D}_\cup$ or when it comes out of $\mathbf{D}_\cup$, it comes back in one control step.

A set $\mathbf{E}$ is also shown, which consists of $\mathbf{D}_\cup$ and light-green shaded region around its periphery. $\mathbf{E}$ can be formally defined as follows:

$$\mathbf{x} \in \mathbf{E} \implies \begin{cases} \mathbf{x} \in \mathbf{D}_\cup & \text{or} \\ \exists\ \mathbf{x_D} \in \mathbf{D}_\cup \text{ and } \exists\ \mathbf{u_D} \in \mathbf{S}_u \text{ s.t. } \mathbf{x_D} + \mathbf{u_D} = \mathbf{x} \end{cases} \tag{4.25}$$

Now it can be easily shown that for any $\mathbf{x}_k \in \mathbf{D}_\cup$, $\mathbf{x}_{k+n} \in \mathbf{E}, \forall\ n > 0$, when the states are governed by Eq. (4.8).

Figure 4.14 shows the set $\mathbf{D}_\cup$ and $\mathbf{E}$ for Gaussian functions with $a = 15^\circ$ and $b = 14^\circ$. It is observed that as a and b becomes closer in value, the length of set $\mathbf{D}(\cdot)$ increases, that results in

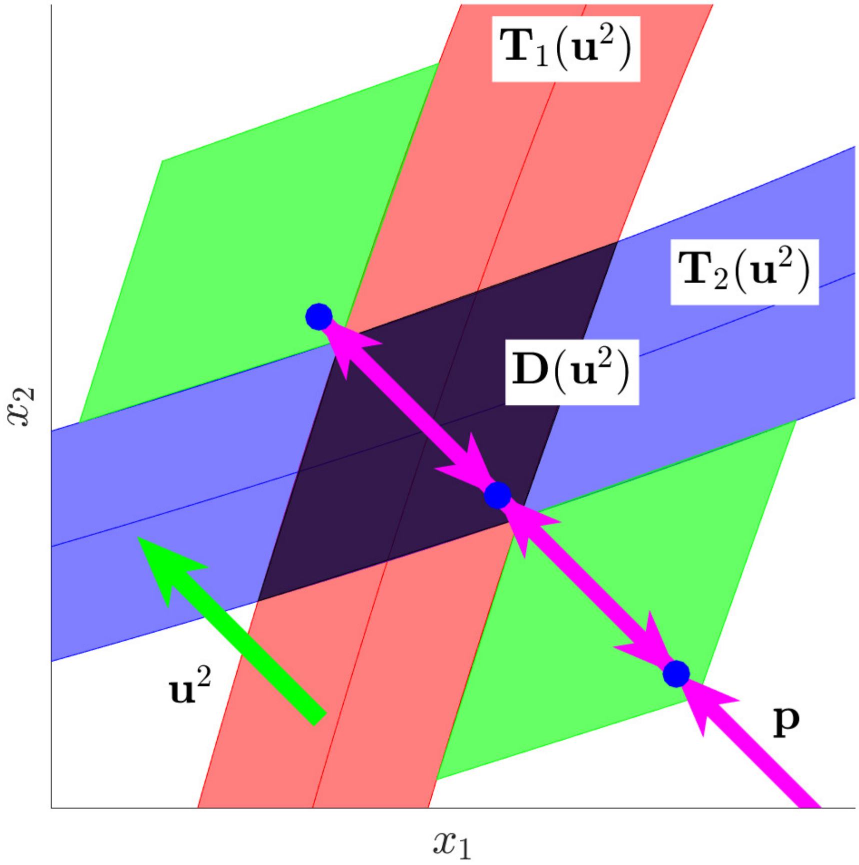

$\mathbf{T}_1(\mathbf{u}^2)$
$\mathbf{T}_2(\mathbf{u}^2)$
$\mathbf{D}(\mathbf{u}^2)$
$\mathbf{u}^2$
$\mathbf{p}$
x_2
x_1

Figure 4.12: Illustration of $\mathbf{D}(\mathbf{u}^2)$ as a separate portion of $\mathbf{D}_\cup$, and a path p that ends in oscillation around $\mathbf{D}(\mathbf{u}^2)$.

increase in size of sets $\mathbf{D}_\cup$ and $\mathbf{E}$. For $a = b$, the transition regions $\mathbf{T}_1(\cdot)$ and $\mathbf{T}_2(\cdot)$ then become identical, which results in the set $\mathbf{D}_\cup$ and $\mathbf{E}$ to be unbounded.

4.4 Simulation results

In this section, we study the performance of the proposed algorithm in simulation. First, we study the approach on a generalized system where the set $\mathbf{D}_\cup$ is non-contiguous. Then we implement the approach on the system corresponding to the physical setup used in this work, where the measurement function is the product of Gaussians. We also compare the performance of the approach with two other approaches, extremum seeking control and extended Kalman filter-based

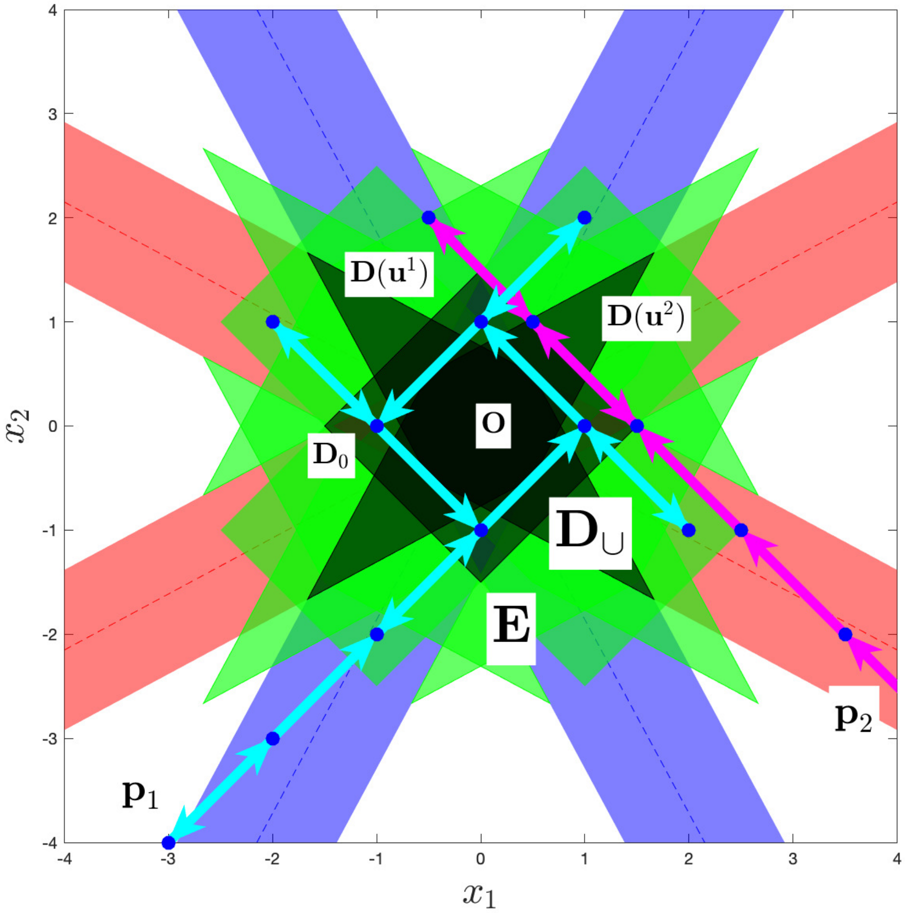


Figure 4.13: Illustration of an isolated portion of set $\mathbf{D}_\cup$ and a sample path p.

algorithms.

4.4.1 Case with Non-contiguous $\mathbf{D}_\cup$

To illustrate the applicability of Theorem 4.2.1 on a system with non-contiguous $\mathbf{D}_\cup$, the following measurement model is considered:

$$\mathbf{y}_k \triangleq \begin{bmatrix} h_1(\mathbf{x}_k) \\ h_2(\mathbf{x}_k) \end{bmatrix} = \begin{bmatrix} C_p e^{-\left(\frac{x_{1,k}^4}{2a} + \frac{x_{2,k}^2}{2b}\right)} + v_{1,k} \\ C_p e^{-\left(\frac{x_{2,k}^4}{2a} + \frac{x_{1,k}^2}{2b}\right)} + v_{2,k} \end{bmatrix}, \tag{4.26}$$

with $C_p = 5$, $a = 10000°$, $b = 15°$ and $\delta = 2°$. The terms $v_{1,k}$ and $v_{2,k}$ represent measurement

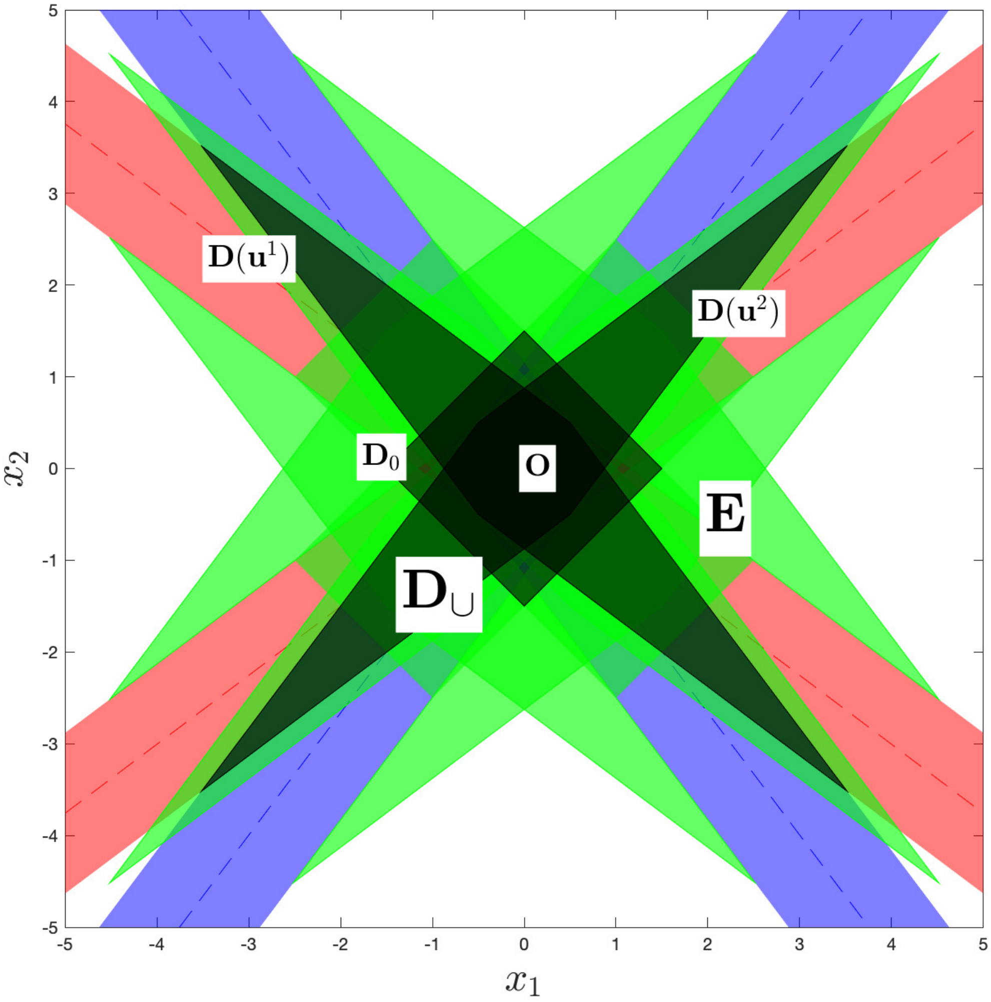


Figure 4.14: Illustration of set $\mathbf{D}_\cup$ for the scenario when the measurement functions are Gaussians and corresponds to the physical setup with $a = 15°$ and $b = 14°$.

noise that follows normal distribution ($\sim \mathcal{N}(0, \sigma_v^2)$). Figure 4.15 illustrates the convex super-level sets of measurement functions of the system along with the transition region bands. It can be observed that the intersection of the transition regions occurs at multiple places, which results in a non-contiguous $\mathbf{D}_\cup$. Additionally, three sample trajectory paths under the proposed control law are illustrated in Figure 4.15. Initially, σ_v is set to zero for all of the simulation runs. Path p_1, which starts at S_1, reaches a portion $\mathbf{D}_1$ of the non-contiguous set $\mathbf{D}_\cup$, which is away from the origin. The path p_2 starting at S_2 reaches the subset $\mathbf{O}$ of $\mathbf{D}$, which contains the origin. The path p_3 starting at

S_3 reaches a subset $\mathbf{D}_3$. Figure 4.16 shows the evolution of the states corresponding to the paths in Figure 4.15. It can be observed that the trajectory of p_3 oscillates around $\mathbf{D}_3$ until around 23 s. The noise term v_k is made non-zero ($\sigma_v = 9.1 \times 10^{-8}$) after the 23 s mark, that helps in bringing the state out of $\mathbf{D}_3$ and then the state reaches the neighborhood $\mathbf{O}$ of the origin and then oscillates in and one-step out for the rest of the run. This illustrates the advantage of the small measurement noise, which brings stochasticity in the otherwise deterministic system, and is helpful in bringing the system to the optimum point. Furthermore, the upper bound for convergence time, computed using (4.24), is around 29 s (using the sampling time $\Delta T = 0.45$ s) for the paths, and the algorithm takes less than 15 s to reach $\mathbf{D}_\cup$ in all of the scenarios, which supports the validity of the bound.

4.4.2 Case Corresponding to Optical Tracking Setup

Here, the system's model is simulated with $C_p = 0.6,\ a = 15°,\ b = 11°$ and $\delta = 2°$. Figure 4.17 illustrates the elliptical super-level sets of the measurement functions. A sample trajectory path under the proposed control law, starting from an initial point S, is also shown in Figure 4.17. It can be observed that the system moves in a straight line path until it passes a transition region, where it corrects itself. This motion transitions to a perpetual oscillatory motion when it reaches the set $\mathbf{D}_\cup$ around the origin.

To evaluate the performance of the algorithm in terms of convergence speed, two contending algorithms are considered: extended Kalman filtering (EKF)-based alignment control and extremum-seeking (ES) control. The EKF approach that is discussed in Chapter 2 is implemented on each robot separately. In the ES control algorithm [25], sinusoidal perturbation signals are applied to the inputs (states in this setting), and the corresponding changes in outputs are then used to drive the output towards the extrema (maximum in this work). We briefly discuss a two-variable discrete-time version of the ES algorithm that is implemented in this work for comparison. The block diagram in Figure 4.18 illustrates the details of the implementation. Starting from **step 1**, the perturbation signals $A \sin(2\pi f_p \Delta T k)$ and $A \cos(2\pi f_p \Delta T k)$ are added to each of the current mean (state without perturbation) of the states: $\hat{x}_{1,k}$ and $\hat{x}_{2,k}$. The mean is initialized with the given initial condition $[x_{1,0}, x_{2,0}]^T$.

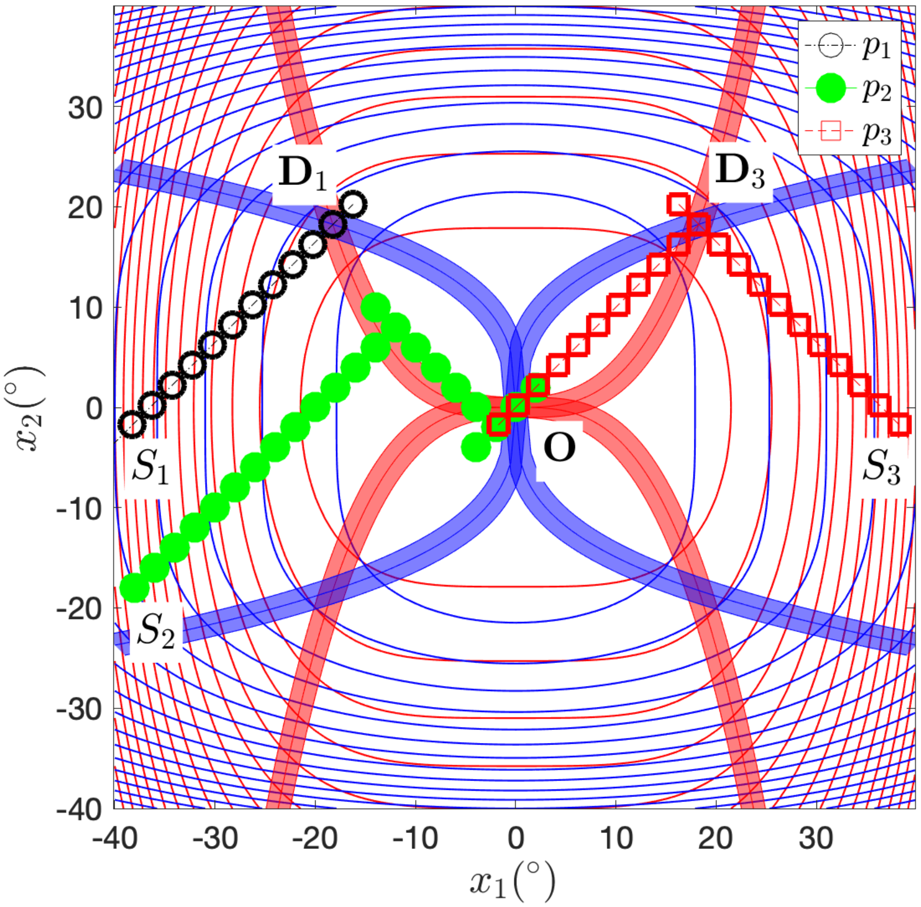


Figure 4.15: Illustration of the paths of sample trajectories for the system with non-contiguous $\mathbf{D}$. The trajectory starting at S_1 reaches a subset $\mathbf{D}_1$ which is not around the origin. The trajectory starting at S_2 reaches a subset $\mathbf{O}$ around the origin. The trajectory starting at S_3 reaches the subset $\mathbf{D}_3$ and oscillates in the subset until the non-zero measurement noise terms are introduced.

This mean corresponds to the present configuration of the agents, which is not known to them. In **step 2** the resulting system outputs, generated by the measurement model applied on the perturbed states, are passed through high-pass filters (HPF). In **step 3** each of the filtered output is multiplied by the corresponding perturbation signals to generate the biases $\xi_{1,k}$ and $\xi_{2,k}$, which are then used to get control terms $u_{1,k}$ and $u_{1,k}$ to move the mean of the states, completing a feedback loop of the system.

For a fair comparison between the algorithms, the perturbation amplitude of ES and the scanning amplitude of EKF are made the same as the step-size of the proposed approach ($\delta = 2^\circ$). We consider ideal and practical scenarios for the simulation of the physical setup that are discussed as follows.

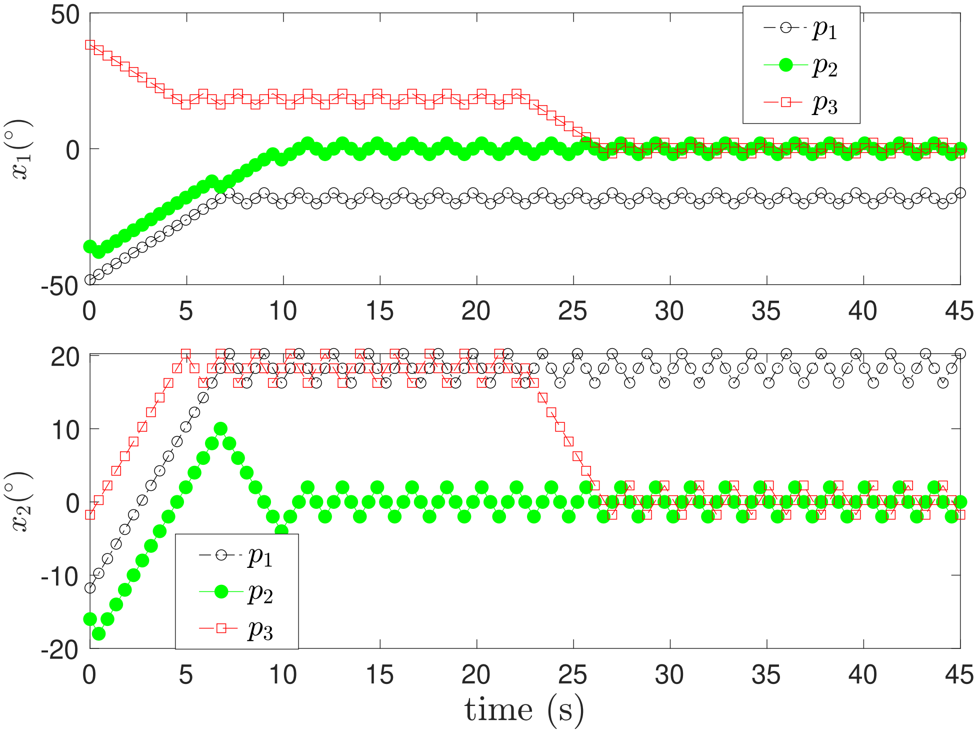


Figure 4.16: Evolution of the states of the trajectories corresponding to the paths illustrated in Figure 4.15.

4.4.2.1 Ideal scenario

In this case we consider the physical system as described by equations (4.3) and (4.2) without any non-idealities like noise, system uncertainties or disturbances. Figure 4.19 illustrates the paths of the three algorithms with the same initial condition **S** ($\mathbf{x}_0 = [-15, 10]^T$) and converging to the neighborhood $\mathbf{D}_\cup$ of the origin **O**. The corresponding evolution of the states with time is shown in Figure 4.20, and the corresponding outputs are shown in Figure 4.21. The proposed algorithm drives the states to $\mathbf{D}_\cup$ in less than 10 s, which is within the theoretical upper bound of 13.5 s, computed from (4.18). It is to be noted that all of these algorithms have steady-state oscillations. These oscillations are necessary for exploration and help in stabilizing the system in the presence of disturbance and slow relative motion. It is observed that the steady-state region of oscillations for the proposed algorithm is twice as bigger as that of the EKF and extremum-seeking, which is not desirable. However, the proposed algorithm converges faster than both of the contending

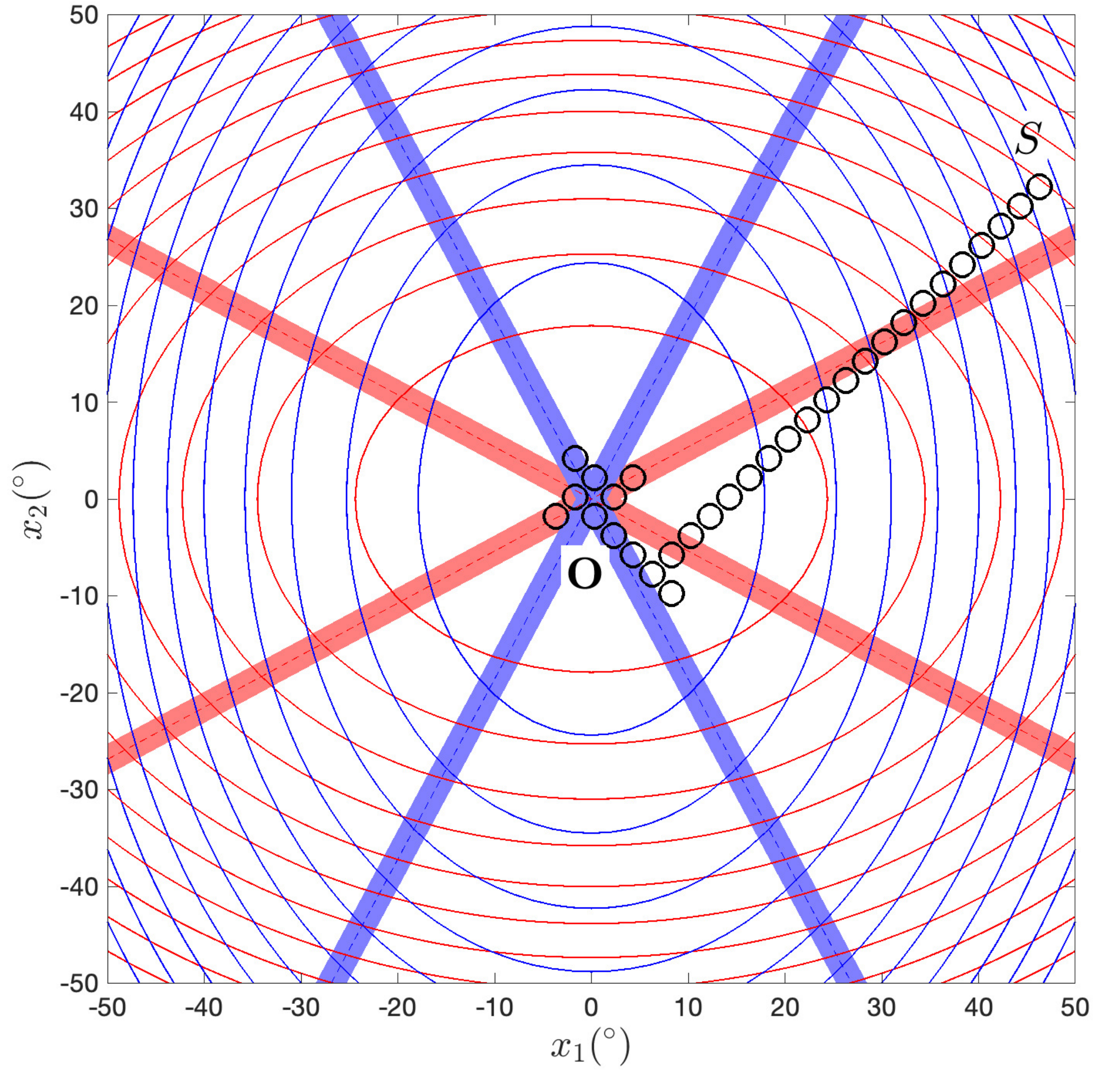


Figure 4.17: Illustration of the level sets, transition regions, and of a sample trajectory for the physical system.

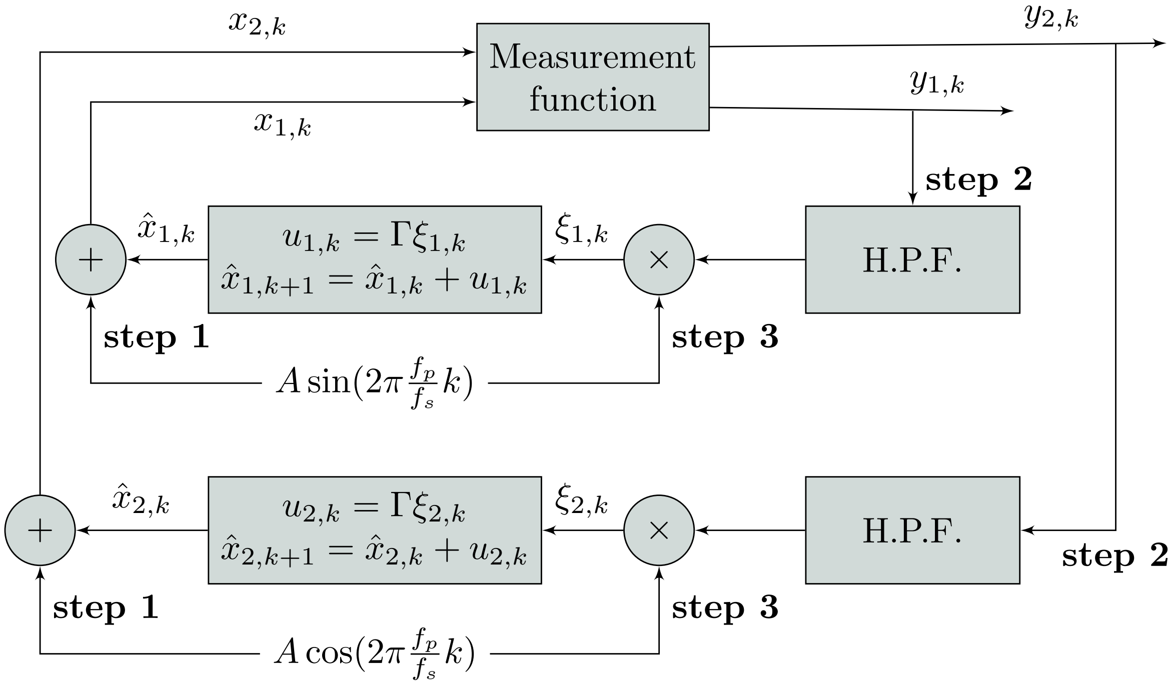

$x_{2,k}$
Measurement function
$y_{2,k}$
$y_{1,k}$
$x_{1,k}$
step 2
$\hat{x}_{1,k}$
$u_{1,k} = \Gamma\xi_{1,k}$
$\hat{x}_{1,k+1} = \hat{x}_{1,k} + u_{1,k}$
$\xi_{1,k}$
H.P.F.
step 1
step 3
$A\sin(2\pi\frac{f_p}{f_s}k)$
$\hat{x}_{2,k}$
$u_{2,k} = \Gamma\xi_{2,k}$
$\hat{x}_{2,k+1} = \hat{x}_{2,k} + u_{2,k}$
$\xi_{2,k}$
H.P.F.
step 2
step 1
step 3
$A\cos(2\pi\frac{f_p}{f_s}k)$

Figure 4.18: Block diagram for extremum seeking control.

algorithms. The convergence speed of EKF is slightly slower than the proposed algorithm. The ES algorithm is significantly slower than the other two algorithms. This delay is attributed to the large time spent by the ES algorithm in exploration. In a trade-off between convergence speed and the size of the region of oscillations, the high convergence speed is more desirable than the low-size region of oscillations for a real scenario with moving robots.

4.4.2.2 Practical scenario

In this case we consider the following modified version of the dynamics of the physical system which resembles the practical scenario:

$$\begin{bmatrix} x_{1,k+1} \\ x_{2,k+1} \end{bmatrix} = \begin{bmatrix} x_{1,k} + u_{1,k} + w_{1,k} \\ x_{2,k} + u_{2,k} + w_{2,k} \end{bmatrix},$$

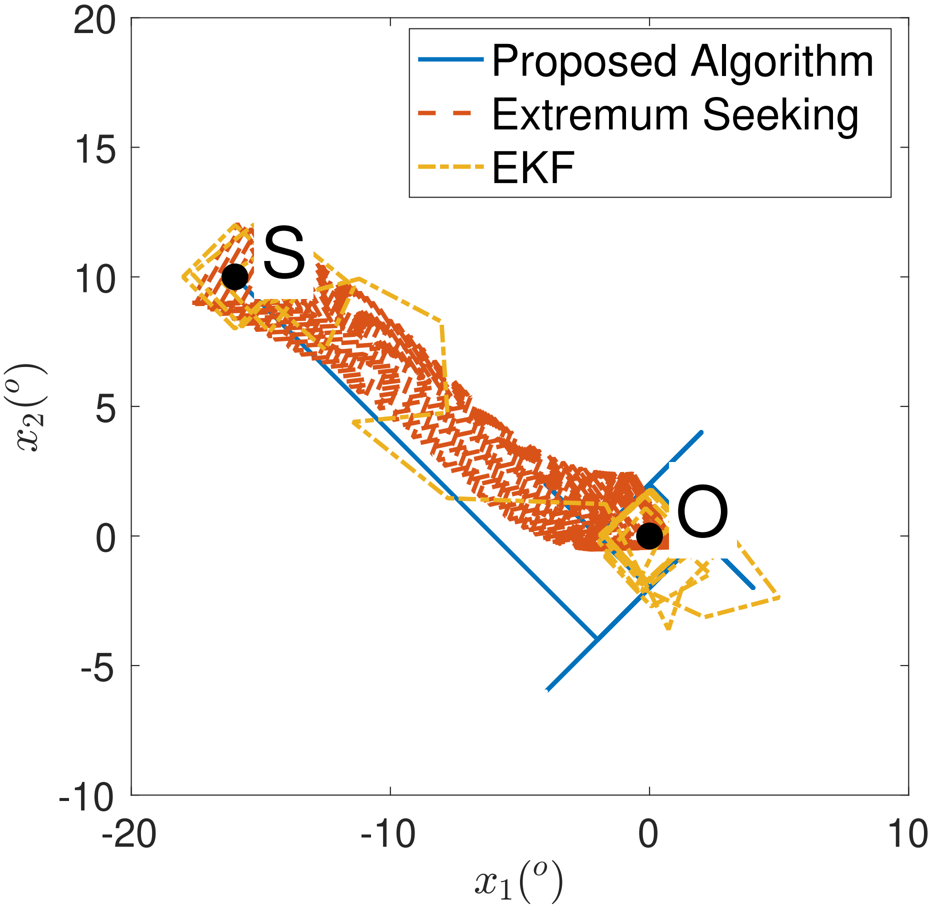


Figure 4.19: Comparison of the trajectories of a sample simulation run for the three algorithms.

where $w_{1,k}$ and $w_{2,k}$ represents the uncertainty in the motion of the motors that follows normal distribution ($\sim \mathcal{N}(0, \sigma_w^2)$). The output now becomes

$$\mathbf{y}_k \triangleq \begin{bmatrix} h_1(\mathbf{x}_k) \\ h_2(\mathbf{x}_k) \end{bmatrix} = \begin{bmatrix} C_p e^{-\left(\frac{x_{1,k}^2}{a^2} + \frac{(x_{2,k}+\kappa_2)^2}{b^2}\right)} + v_{1,k} \\ C_p e^{-\left(\frac{x_{2,k}^2}{a^2} + \frac{(x_{1,k}+\kappa_1)^2}{b^2}\right)} + v_{2,k} \end{bmatrix},$$

where the new terms κ_1 and κ_2 are unknown constants that represent the misalignment angles between the axes of the photo-diode and the LED of robots $\mathbf{R}_1$ and $\mathbf{R}_2$, respectively. It is worthy to be noted that the problem formulation assumes the LED, the photo-diode and the anchor point of the rotation as concurrent. This assumption is still viable as the distance d considered in this work is significantly larger than the offsets between the location of the three aforementioned points

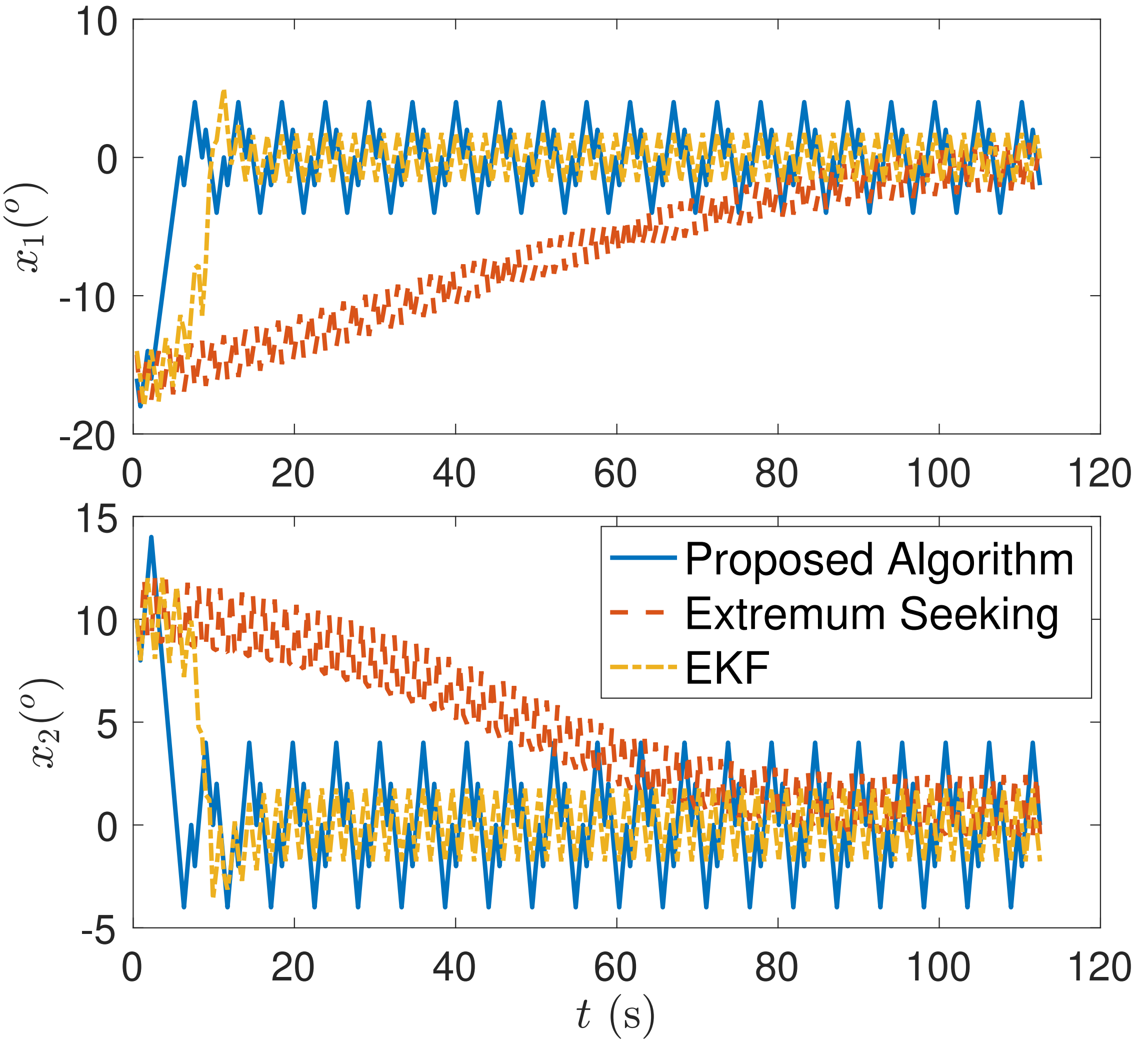


Figure 4.20: Illustration of the evolution of the states for the three algorithms corresponding to paths in Figure 4.19.

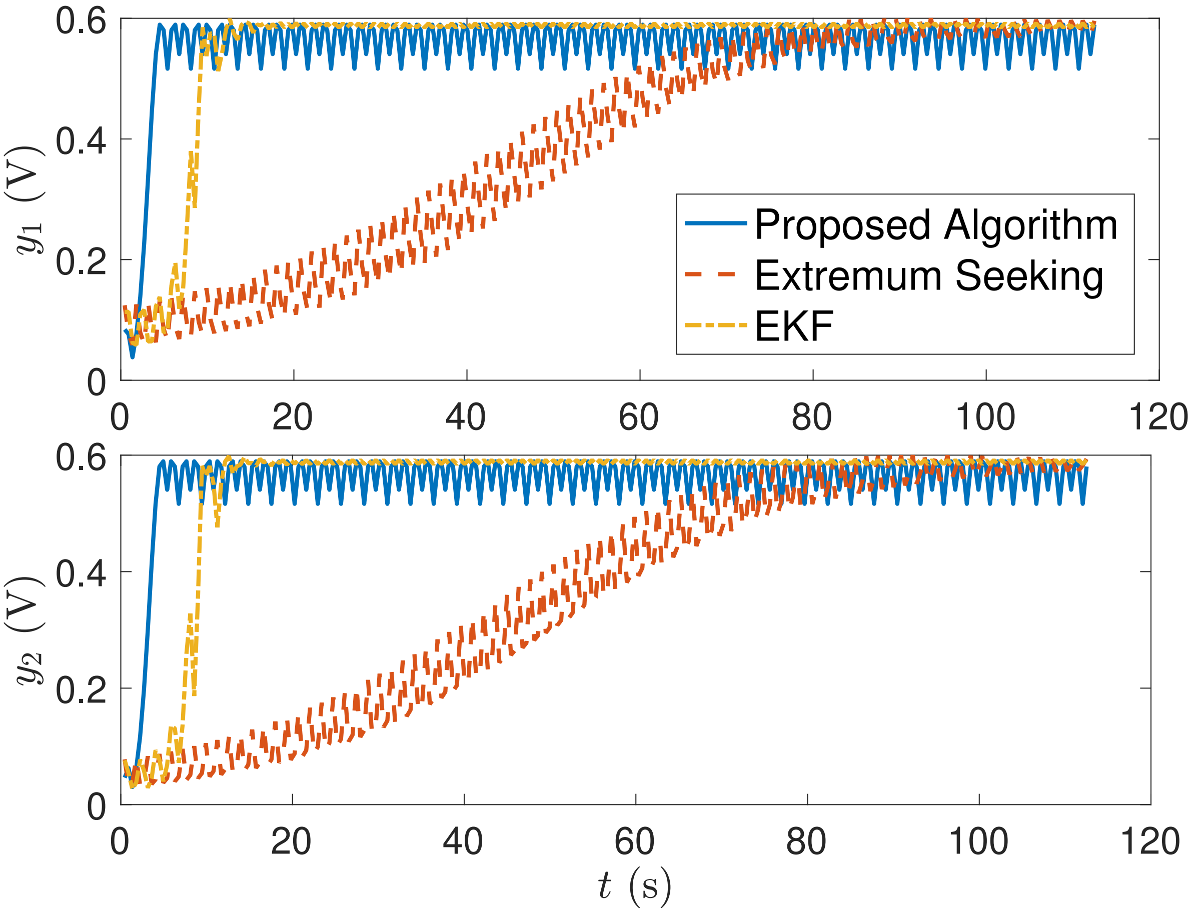


Figure 4.21: Illustration of the outputs for the three algorithms corresponding to the paths in Figure 4.19 and states in Figure 4.20.

of interest. Table 4.2 lists all the parameters of interest which are used in the simulation. The parameters are obtained empirically from the experimental setup used in our work.

Figure 4.22 illustrates the paths of the three algorithms with the same initial condition **S** and converging to origin **O**. The corresponding evolution of the states with time is shown in Figure 4.23, and the corresponding outputs are shown in Figure 4.24. While comparing the algorithms' performance in the ideal scenario and the practical scenario, one can observe that there is no significant deterioration in the performance in terms of the convergence time. However, for the proposed algorithm, the region of steady-state oscillations becomes larger than what is observed in the ideal scenario. This is attributed to the nature of the proposed algorithm that the

Table 4.2: Parameters used in simulation.

Parameter	**Value**	**Description**
C_p	0.6 V	Proportionality constant for measurement functions
a	15°	Gaussian width for function g
b	11°	Gaussian width for function f
σ_v	5.151×10^{-4} V	Standard deviation of measurement noise
σ_w	0.3095°	Standard deviation of actuator uncertainty
κ_1	−0.41°	Misalignment angle of $\mathbf{R}_1$
κ_2	0.14°	Misalignment angle of $\mathbf{R}_2$
ΔT	0.45 s	Sampling time
δ	2°	Step size
f_p	6.67 Hz	Perturbation frequency for ES
Γ	500	Controller gain for ES

feedback control law depends only on the present measurements. Hence, any stochasticity in the measurements directly affects the states of the system. In contrast, the EKF and the ES algorithms have inherent filtering of high-frequency noise; therefore, the stochasticity in the measurements is not reflected directly in the states of the system.

To test the efficacy of the algorithms in the presence of unknown and unwanted relative motion, we introduce a constant disturbance ω in the state dynamics of one agent:

$$x_{1,k+1} = x_{1,k} + u_{1,k} + w_{1,k} + \omega \Delta \mathrm{T}$$

This disturbance emulates a scenario where the agent $\mathbf{R}_1$ is rotating with an unknown angular speed ω and ΔT is the sampling time. This is also equivalent to a scenario where the agent $\mathbf{R}_2$ revolves in a circle around agent $\mathbf{R}_1$ with an angular speed of $-\omega$. Furthermore, to test repeatability, and to characterize the performance and limitations of the algorithms, we simulate the algorithms over a range of ω; $\omega \in [0.15, 3]\ {}^\circ/s$.

Next, we generate 1000 initial points from the set ($\mathbf{x}_0 \in (0^o, 30^o) \times (-30^o, 30^o)$) using the Latin Hyper-cube Sampling (LHS) technique in MATLAB, and perform a simulation run for each of these initial conditions, for every value of angular speed. It is to be noted that we considered only the positive values of the state x_1, as the disturbance is positive and therefore, it acts to move

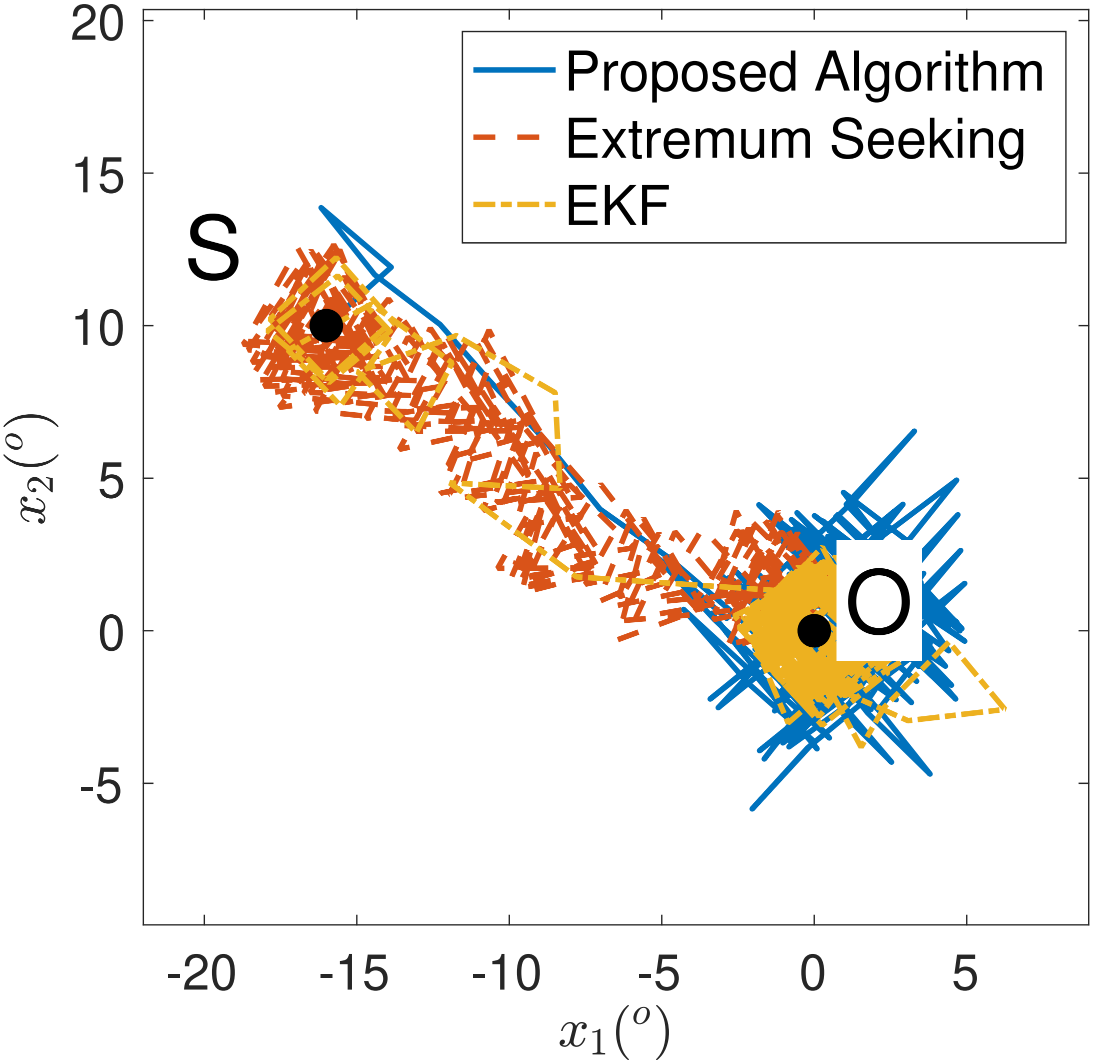


Figure 4.22: Comparison of the trajectories of a sample simulation run for the three algorithms in the practical scenario.

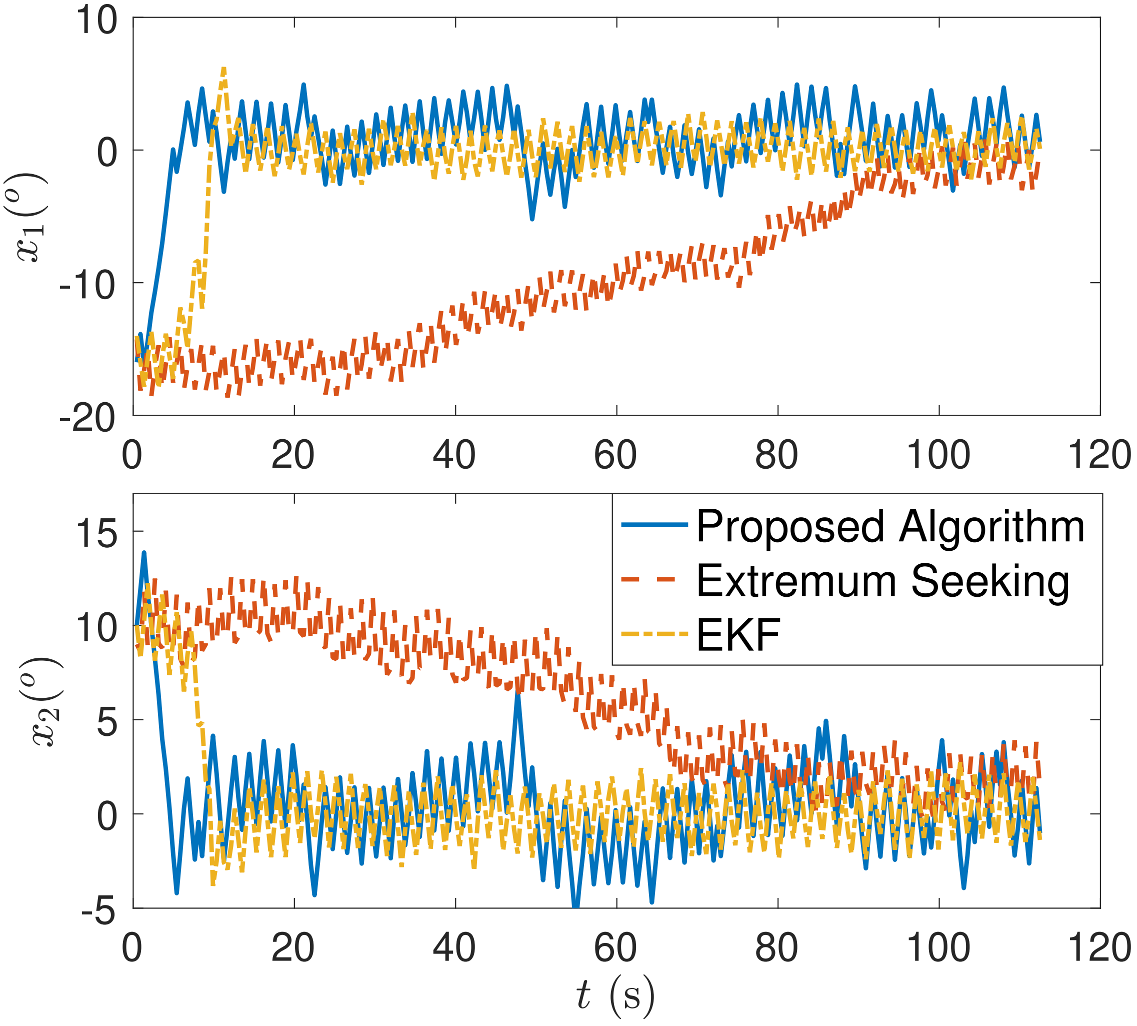


Figure 4.23: Illustration of the evolution of the states for the three algorithms in the practical scenario.

the states away from the origin. A simulation run is labeled as converged if, at any iteration k, the states lie inside the set $\mathbf{D}_{\cup}$. The performance at each speed is measured by counting the number of converged simulation runs ($\mathbf{N}_C$) and the average tracking error of the converging runs ($\mathbf{E}_C$), which is defined for a simulation run as follows:

$$\mathbf{E}_C = \frac{10}{n_f}\sum_{k=n_s}^{n_f}\|\mathbf{x}_k\|_2, \text{ with } n_s = n_f - n_f/10 + 1$$

where n_f is the total number of time-steps in one simulation run of an algorithm (the value of

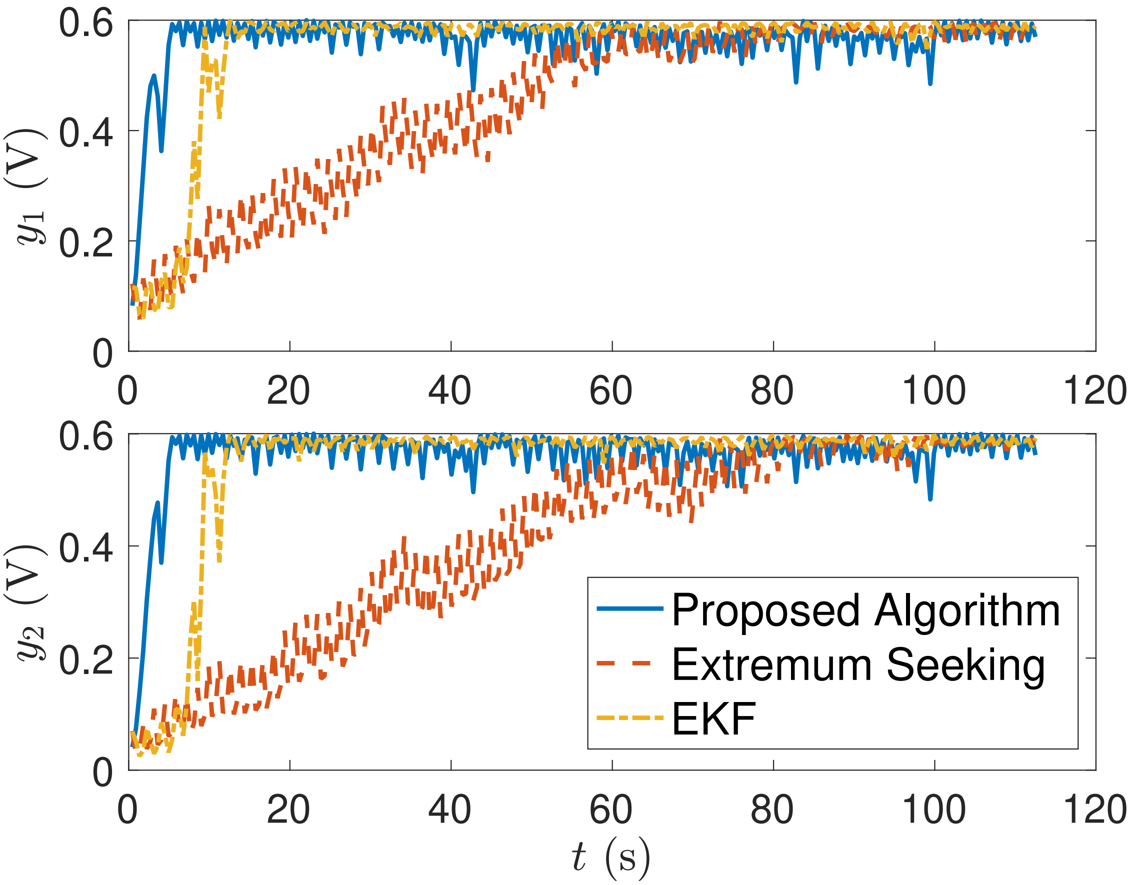


Figure 4.24: Illustration of the outputs for the three algorithms in the practical scenario.

5000 is used in the simulation runs). The final 10% iterations ($n_f/10$) are considered to effectively capture the average of steady-state oscillating points. A new term called *disturbance fraction* denoted by χ is introduced, that equals $\omega\Delta T/\delta$ to represent the relative strength of the disturbance in comparison with the speed of robots' actuation.

Figure 4.25 shows the cumulative performance of the three algorithms in terms of $\mathbf{N}_C$ and $\mathbf{E}_C$ over a range of χ. The mean tracking error $\mathbf{E}_C$ is plotted for an algorithm for a value of disturbance strength only when more than 10% of the runs show convergence. All the algorithms show a decreasing trend of convergence count as the disturbance fraction χ increases. The convergence count of the proposed algorithm remains highest throughout the range, with close to 100% runs converging for the low range of the disturbances. Furthermore, the mean and standard deviation of the error $\mathbf{E}_C$ stays lowest for the proposed algorithm in the low range of disturbances. The convergence count of the EKF algorithm starts with about 600 that gradually decays to about 100. From χ = 0.15 onward, the mean tracking error of EKF stays close to zero with consistent values of the standard deviation. It outlines that the system states for the converging runs stay moderately close to the origin throughout the run. The standard-deviation of $\mathbf{E}_C$ values stays significantly high for most of the χ values. The performance of the ES algorithm is poor; its convergence count starts with about 400 and drops below 10% at a low χ value of 0.10. Therefore only two data points of $\mathbf{E}_C$ are plotted. As discussed earlier, the ES algorithm is significantly slower than the other two algorithms, and hence it fails to converge in the presence of disturbance.

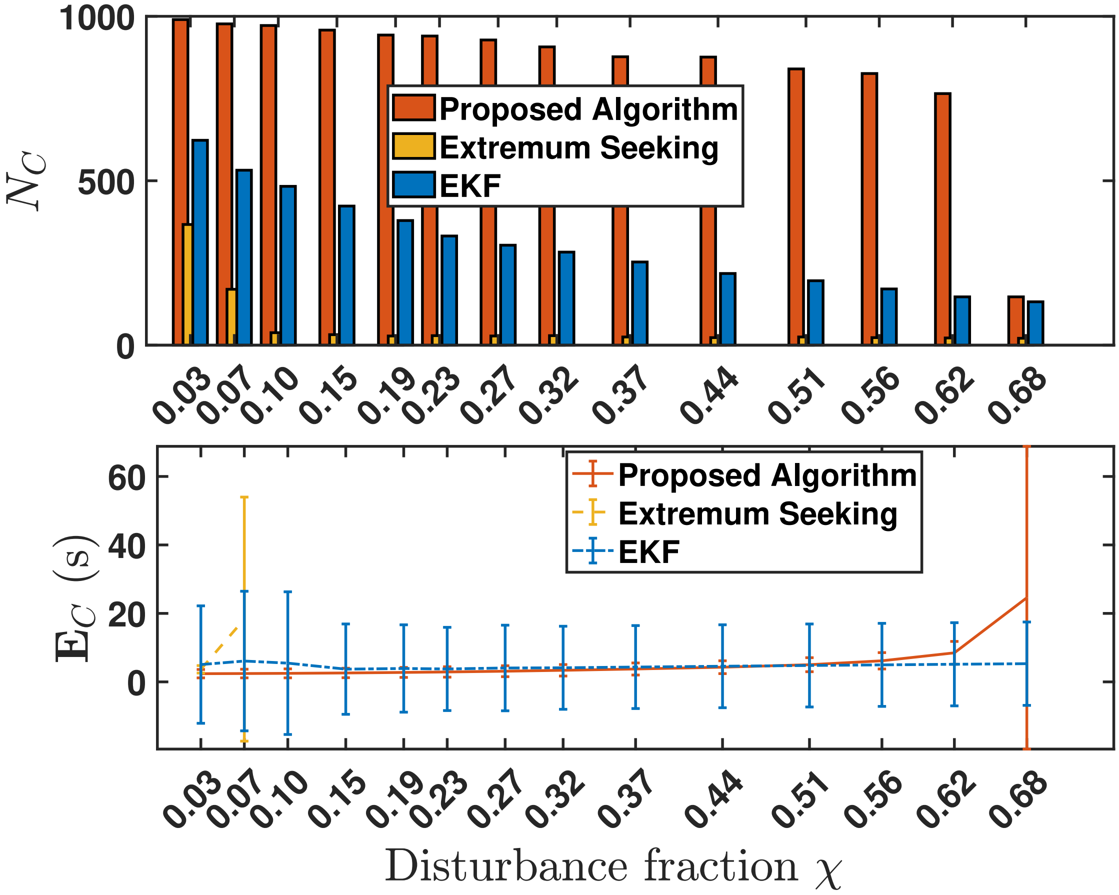


Figure 4.25: Illustration of performance of the algorithms in terms of convergence count $\mathbf{N}_C$ and tracking error $\mathbf{E}_C$ over a range of disturbance fraction χ. The error bars for $\mathbf{E}_C$ denote the standard-deviation.

To study the effect of the initial conditions on the convergence performance, we considered a range of magnitude of initial conditions. For a given magnitude β, 1000 points are generated using the LHS technique on a circle of radius β centered around the origin. It is to be noted that our domain $\mathbf{\Pi}$ represents angles, and hence the radius β is expressed in degrees. A new metric called convergence time $\mathbf{T}_C$, defined as the time taken to converge in a simulation run, is considered to evaluate the performance.

Figure 4.26 shows the convergence performance of the algorithms over a range of magnitude of initial conditions. For low initial conditions until 20°, all of the simulation runs result in convergence for each of the algorithms. At $\beta = 30^\circ$, the performance of the EKF and that of the ES algorithms start degrading where the ES algorithm has a higher convergence count. At $\beta = 60^\circ$, the convergence counts of both the algorithms diminish to less than 10%. The convergence count of the proposed algorithm also degrades but significantly slower than the two algorithms, and despite the high initial condition of $\beta = 90^\circ$, more than half of the simulation runs result in convergence. Furthermore, it is observed that the convergence time ($\mathbf{T}_C$) increases with the magnitude of the initial conditions β. Additionally, the standard deviation of $\mathbf{T}_C$ also increases with β, which is attributed to the increase in the stochasticity of the measurements; at high initial conditions, the measurement values become small, and the magnitude of noise becomes comparable to the measurements. This decrease in signal-to-noise ratio (SNR) contributes to the stochasticity in the dynamics and hence results in the poor performance of the algorithms. Moreover, for high initial conditions, the control steps for each agent for the EKF algorithm are large, which violates its quasi-static assumption on the system dynamics. This violation further affects the failure rate of convergence of the simulation runs, even for the relatively smaller initial conditions. On a side note, it can be inferred that the converging runs of the EKF algorithm observed earlier in Figure 4.25 are the ones that started with smaller initial conditions. The mean tracking error $\mathbf{E}_C$ remains small for the ES and the proposed algorithm; however, for the EKF, the error increases with the β for the reason that a significant fraction of the converging runs diverges away from the origin despite coming close to the origin.

Henceforth, we have seen from the numerical results that the proposed approach significantly outperforms the EKF and the ES algorithm in terms of convergence speed and robustness to noise and unknown disturbances and large initial conditions. Moreover, it is to be noted that the proposed algorithm is computationally economical than the other two algorithms, which becomes advantageous in real-time onboard implementation as it plays a major role in deciding the sampling time ΔT.

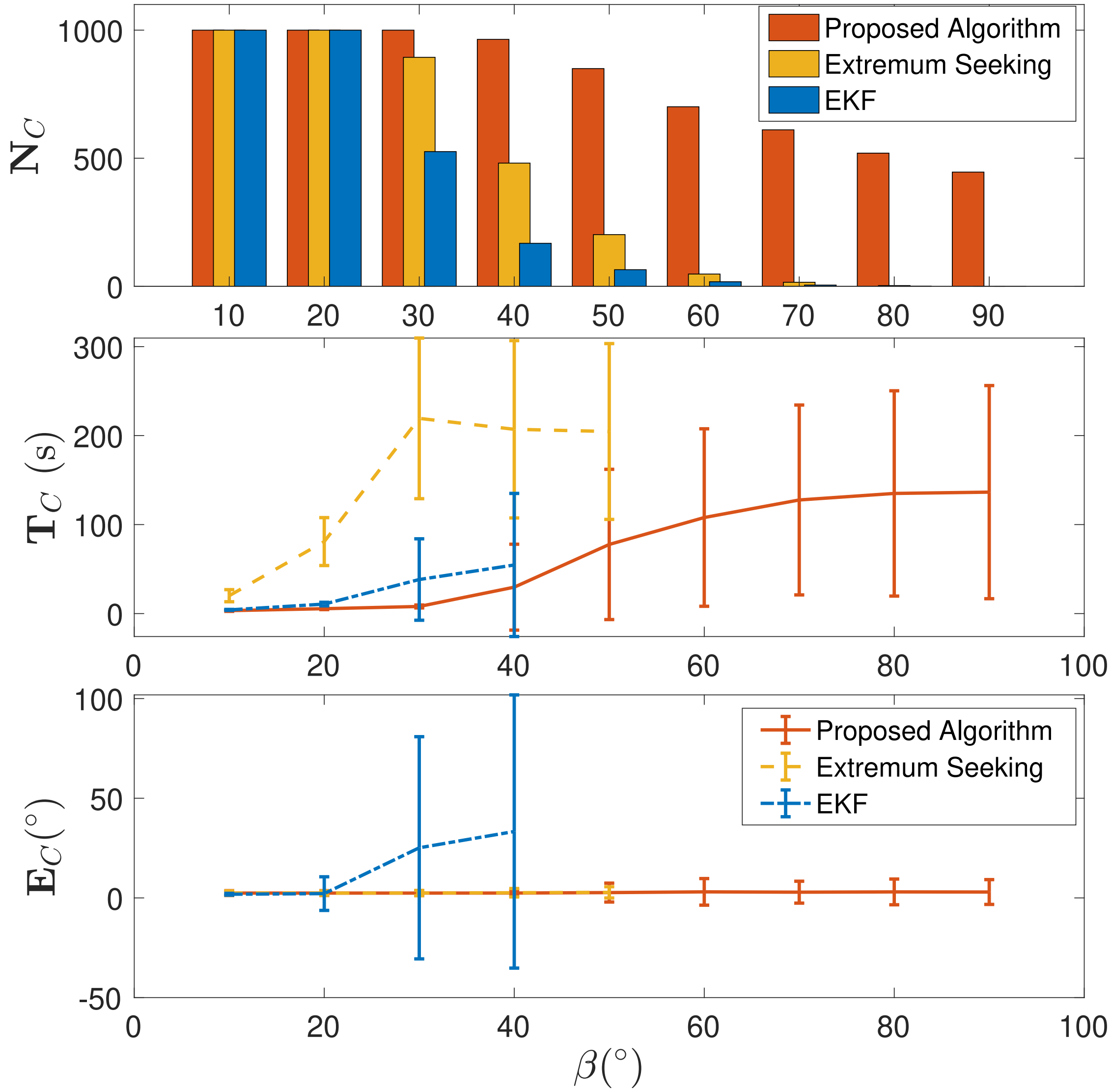


Figure 4.26: Illustration of performance of the algorithms in terms of convergence count $\mathbf{N}_C$, convergence time $\mathbf{T}_C$, and tracking error $\mathbf{E}_C$ over a range of magnitudes of initial condition β. The error bars for $\mathbf{T}_C$ and $\mathbf{E}_C$ denotes the standard-deviation.

4.5 Experimental Results

In this section, we test the efficacy of the proposed algorithm in experiments. We first describe the details of the experimental setup used in this work and then discuss the results.

Figure 4.27 illustrates the hardware components of one of the two robots. For the transmitter, a Cree XR-E Series Blue LED with a principal wavelength of 480 nm is used, and for the receiver, a blue-enhanced photo-diode from Advanced Photonix (part number PDB-V107) is used. A DYNAMIXEL® servo motor (model number XL430-W250-T) is used to control the pointing angle of the transceiver. A Beaglebone Blue® board is used as an on-board computer for real-time processing and computation. The robot in Figure 4.27 is placed on a metal disc that serves as a rotating base to emulate relative angular motion between the two robots. Another robot with the same configuration on a static platform is used as the second agent. Furthermore, we used OptiTrack® motion capture system to access the ground truth values of the system's states. Figure 4.27 shows the reflective markers placed on the top of the robot and on the base disc, which are used by the motion capture system. The illustrated robot is labeled as $\mathbf{R}_1$, and the copy of this robot is labeled as $\mathbf{R}_2$, which is placed on a static base. Figure 4.28, shows the experimental setup consisting of robots $\mathbf{R}_1$ and $\mathbf{R}_2$, that are placed 2.8 meters apart. The experiments are performed in a dark room to minimize ambient optical noise.

The code to implement the proposed algorithm setup is written in Python, which is then executed on the on-board computer of each robot. Figure 4.29 and Figure 4.30 show the results of a sample experiment run where the disc is rotating with an angular speed of $\omega = 1^\circ/\text{s}$ ($\chi = 0.23$). The path of the trajectory of the states of the system is shown in Figure 4.29, and the evolution of the states and the intensity measurements is shown in Figure 4.30. The starting point $\mathbf{S}$ denotes the initial condition, where the transceivers of both the robots were pointing away from the LOS. During the course of the algorithm, the states reach the neighborhood $\mathbf{D}_\cup$ of the origin $\mathbf{O}$, in about 10 s, and then they oscillate around the origin for the rest of the experiment run. The convergence of the states to the set $\mathbf{D}_\cup$ indicates the achievement of the near LOS by the transceivers of the robots. The robots' signal strength measurements start from low initial values, and then they reach steady-state

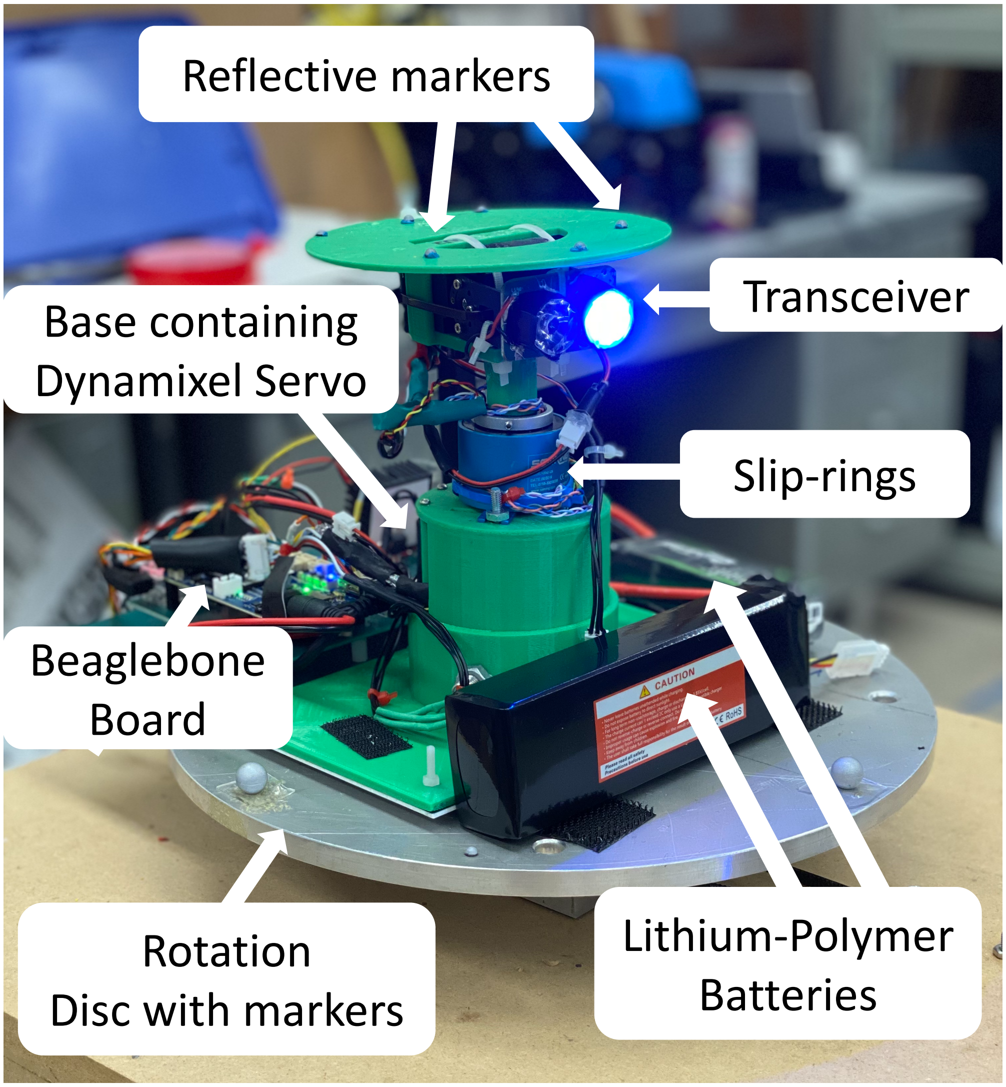

Reflective markers
Transceiver
Base containing Dynamixel Servo
Slip-rings
Beaglebone Board
Rotation Disc with markers
Lithium-Polymer Batteries

Figure 4.27: Illustration of a robot with the optical transceiver system. The robot stands on a rotating disc to emulate the relative motion between two robots.

oscillations at around 0.4 V when the system reaches the neighborhood of LOS.

Next, to test the repeatability of the experiments and limitations of the proposed algorithm on the experimental setup, we consider a range of angular speeds for the disc similar to the range used in the simulation. We first perform 10 experimental runs with practically the same initial condition for each of the angular speeds. We perform additional 20 experimental runs if the initial 10 runs exhibit both success and failure in convergence in order to obtain better statistical measures. The sign of the initial angle $x_{1,0}$ of robot $\mathbf{R}_1$ is chosen to be the same as the sign of the ω so that the rotation of the disc moves the pointing direction of $\mathbf{R}_1$ away from the LOS. Henceforth, all of the

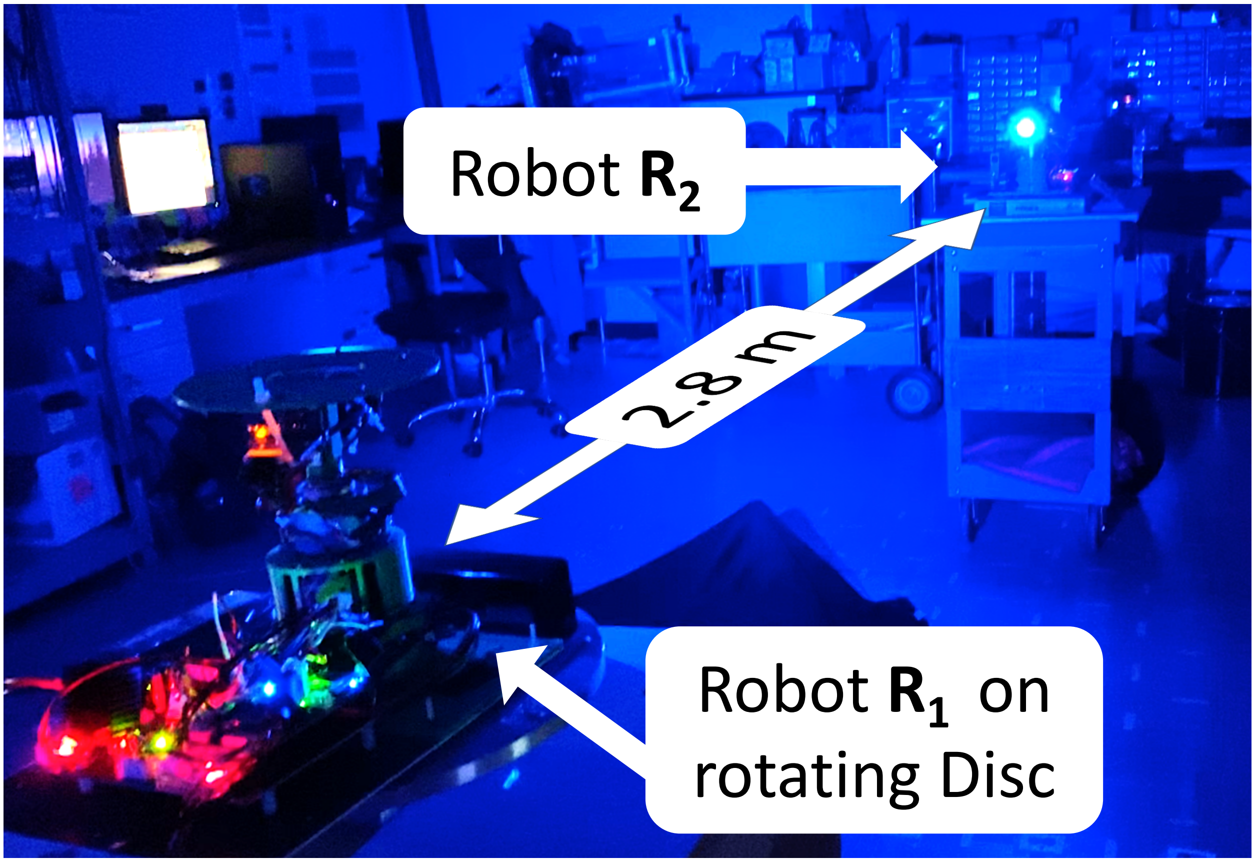


Figure 4.28: Illustration of experimental setup of the two-robots scenario. The overhead lights of the room are turned off to minimize the ambient optical noise.

experiment runs start from the initial condition $\mathbf{x}_0 = [16°, -23°]^T$.

Figure 4.31 shows the results of the experiment runs, performed over a range of rotational speeds of the disc in terms of percentage of converging runs along with the metrics $\mathbf{T}_C$ and $\mathbf{E}_C$ which were defined in Section 4.4. The x-axis of the sub-plots represents the disc speed in terms of the disturbance fraction χ. The number above each of the bars represents the total count of experiment runs performed at that value of χ. Similar to the simulation, the $\mathbf{T}_C$ and $\mathbf{E}_C$ are only plotted at a particular value of χ, if the convergence count is greater than 10% of the total count. It is observed that at low disturbance values until $\chi = 0.19$, the success rate of convergence is 100%, and then it decreases gradually. The average of convergence time $\mathbf{T}_C$ remains between 10 s and 20 s till $\chi = 0.44$, as the initial condition is the same for all the runs. Based on consistent values, it is inferred that when an experiment run converges, it is likely to reach the LOS-neighborhood

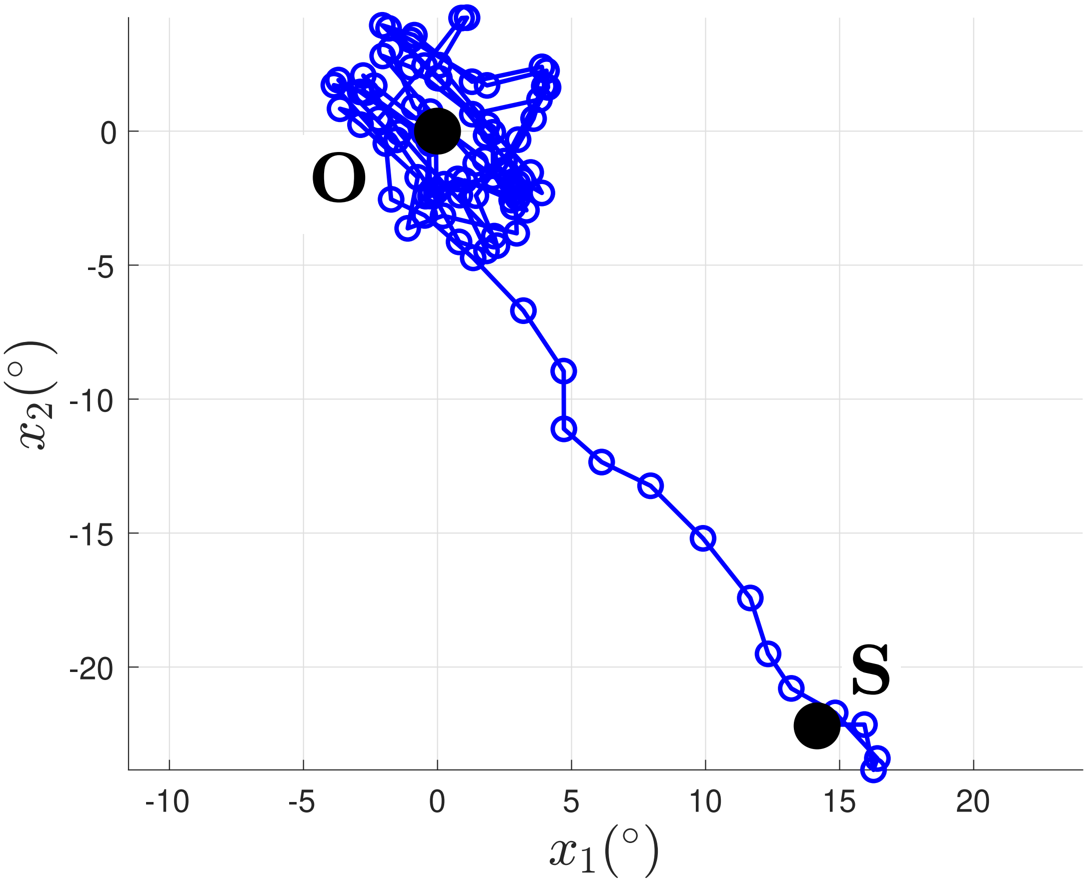


Figure 4.29: Path of the trajectory of a sample experiment run when the base disc rotates with the angular speed of $\omega = 1^\circ$/s.

in a certain time limit. Beyond that, it is extremely unlikely to converge as due to the platform's rotation, the pointing direction of the transceiver of robot $\mathbf{R}_1$ moves far away from the LOS where the effect of the measurement noise and the motor uncertainty becomes prominent. The average tracking error $\mathbf{E}_C$ stays at consistently low value of about 4 degrees until $\chi = 0.31$. Beyond that value, it starts increasing gradually, such that at $\chi = 0.56$, its value averaging at 35 degrees with significantly high standard-deviation. These high values at high disturbances depict that the system leaves the LOS-neighborhood after convergence most of the time.

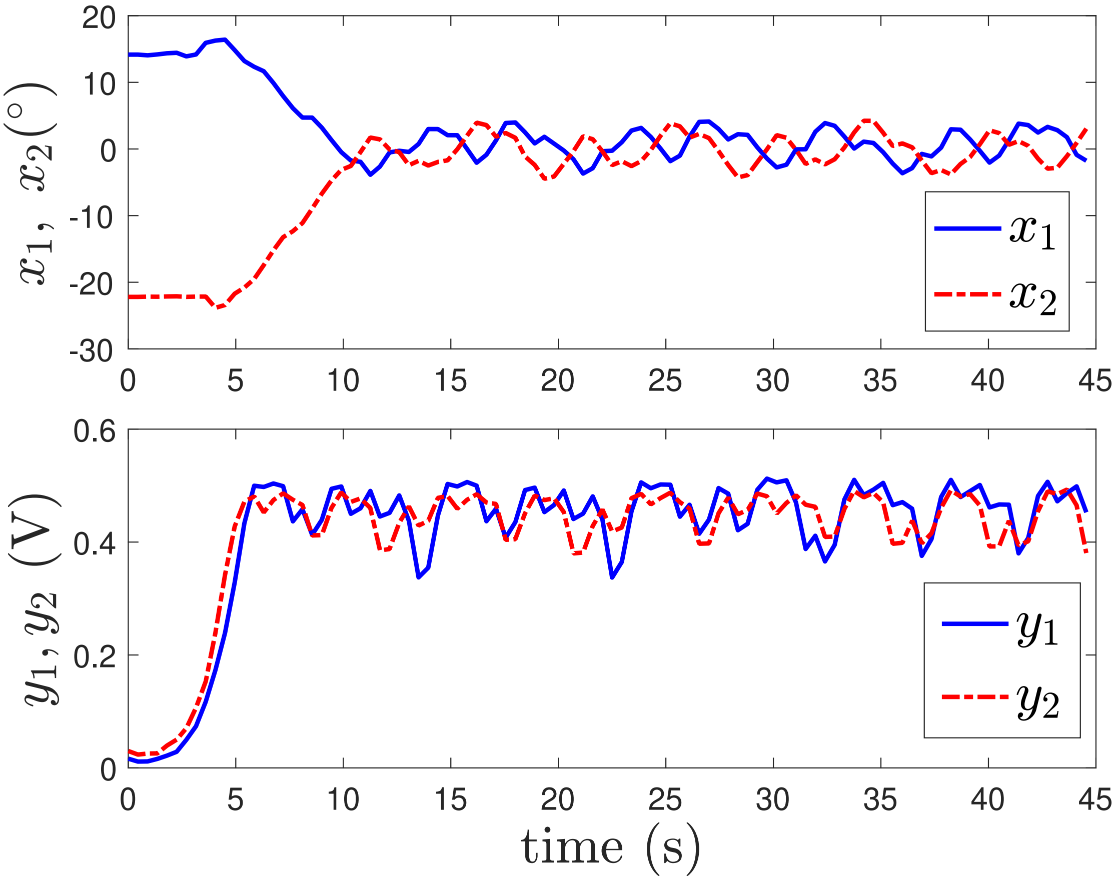


Figure 4.30: Evolution of the system's states and the output corresponding to the path of the sample experiment run shown in Figure 4.29.

4.6 Chapter Summary

In this work, we formulate a bidirectional optical beam tracking problem as a discrete-time dynamical system. We propose a model-free output feedback control law for a class of systems that follow certain assumptions with the constraint that the control command of an agent can depend only on the information accessible to that agent. Through rigorous analysis, we show that for any initial condition, the proposed control law drives the system to a defined set $\mathbf{D}_\cup$ in a finite number of steps. For our physical setup with Gaussian measurement function choices, we get stronger results regarding the convergence towards $O(\delta)$ neighborhood of the origin that corresponds to LOS. The proposed algorithm is computationally economical than the two contending approaches:

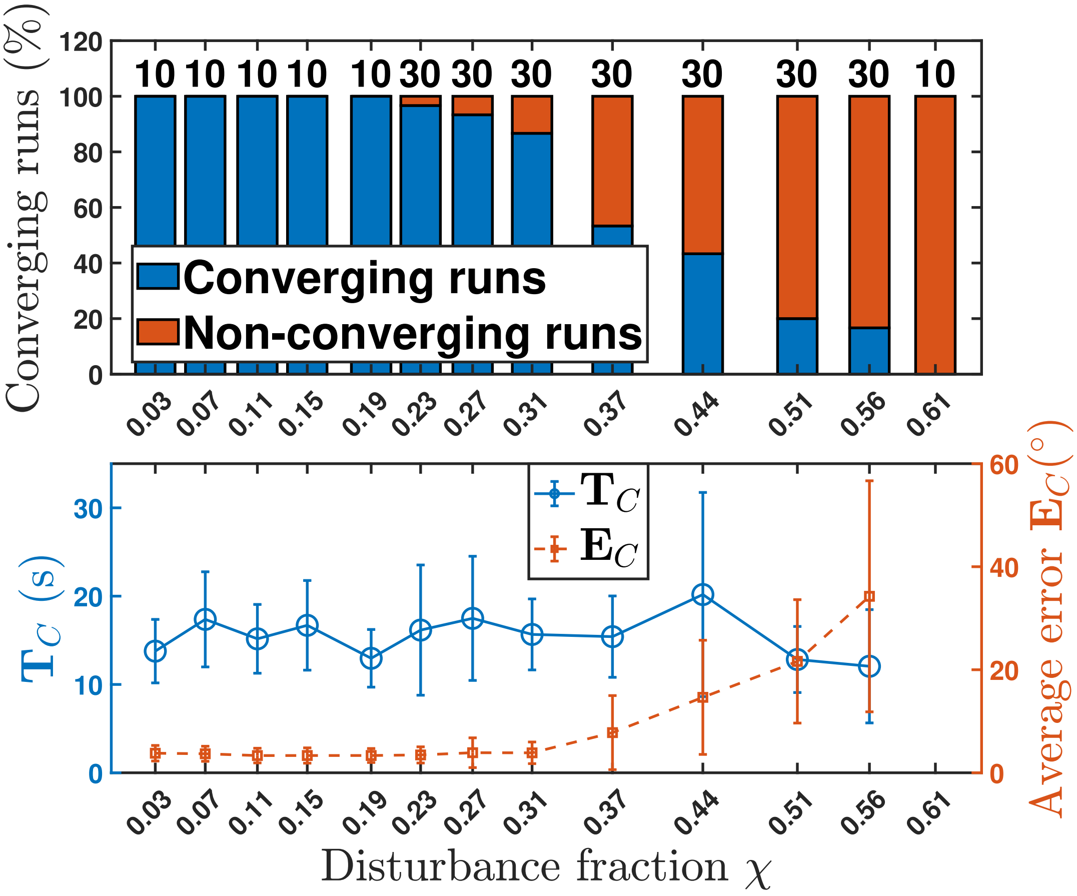


Figure 4.31: Illustration of performance of the algorithms in terms of convergence count $\mathbf{N}_C$ and tracking error $\mathbf{E}_C$ over a range of disturbance speeds. The error bars for $\mathbf{E}_C$ denotes the standard-deviation. The number above each of the bar represents the total count of experiment runs performed at that angular speed.

ES and EKF, and superior in terms of convergence speed, robustness to unknown disturbances, and handling large initial conditions. The proposed approach is also validated on an experimental setup consisting of two robots in the presence of a constant external disturbance.

CHAPTER 5

SIMULTANEOUS BI-DIRECTIONAL ALIGNMENT CONTROL IN THE 3D SPACE

Motivated by the model-free approach for the bi-directional 2D setting that is proposed in Chapter 4, in this work, we explore a model-free bi-directional active alignment control-based approach for the 3D setting. Utilizing the uni-modal nature of the dependence of the light signal strength on local angles, we propose a novel triangular exploration algorithm, that does not require the knowledge of the underlying light intensity model, to maximize the signal strength that leads to achieving and maintaining LOS. The method maintains an equilateral triangle shape in the angle space for any three consecutive exploration points, while ensuring the consistency of exploration direction with the local gradient of signal strength. The approach can be directly implemented on two robots for bi-directional setting, without the need of any synchronization between the robots.

The effectiveness of the approach is first evaluated in the simulation setup of two robots, which was presented in Section 3.3, by comparison with the EKF-based approach and the ES approach. Simulation results show that the proposed approach is optimal and effective in terms of its convergence speed for a wide range of relative speed and distance between the robots. The performance of the approach is further assessed against the EKF approach and the ES approach on the experimental setup, which was presented in Section 3.4. The experimental results further support the superiority of the approach with the other two contending approaches.

The efficacy of the approach and the overall communication system is further demonstrated in underwater setting where communication is performed simultaneously with the alignment control. One way communication and alignment are first demonstrated on a setup where a human operator wirelessly controls a robot in the underwater scenario using an LED communication-based joystick, followed by demonstration of bi-directional alignment control and communication between two underwater robots.

The organization of the rest of the chapter is as follows. Section 5.1 reviews the basic problem set up. The details of the alignment algorithm is discussed in Section 5.2. Simulation results are presented in Section 5.3. In-air experiment results are provided in Section 5.4, followed by the discussion on underwater experimental setup and results in Section 5.5. Finally, concluding remarks are provided in Section 5.6.

5.1 Review of System Setup and Modeling

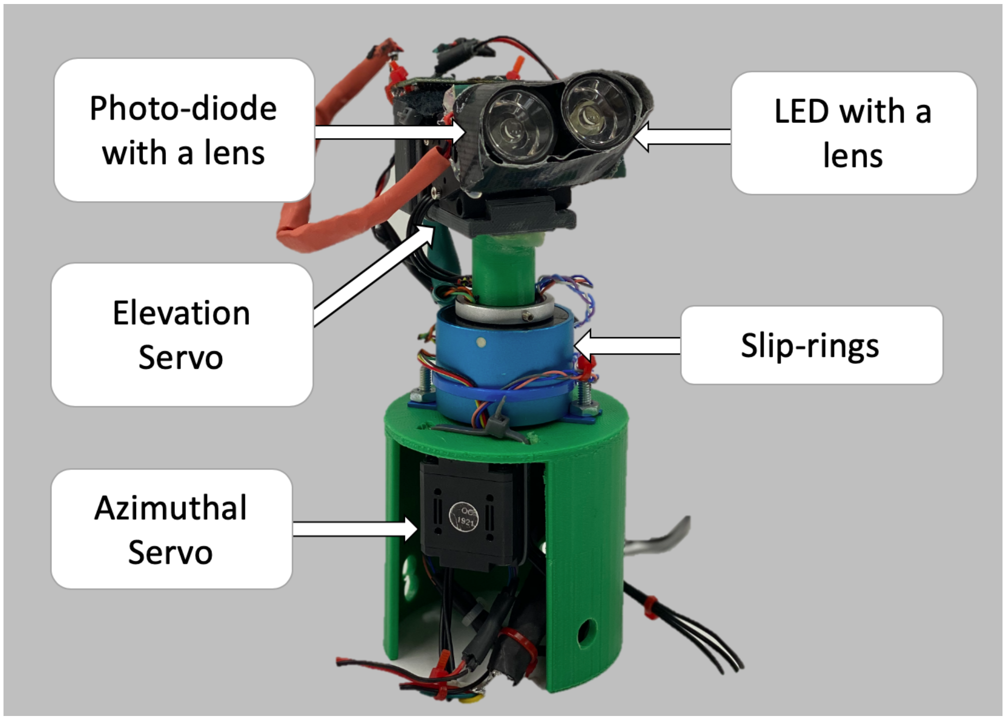


Figure 5.1: Hardware description of the active transceiver module.

Here, we first briefly review the hardware setup and system model from Chapter 3 and describe the relevant mathematical representation of the system behavior. For simplicity in formulation, we consider the model of each robot separately in this work. Figure 5.1 shows the hardware setup consisting of the LED-photodiode pair in the transceiver module that is mounted on a 2DOF active pointing mechanism. The same setup was used in the experiments which were presented in Chapter 3. Next, we derive the states of the system for one robot from Eq. (3.4):

$$\mathbf{x} = \begin{bmatrix} x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \phi \\ \theta \end{bmatrix}, \tag{5.1}$$

where ϕ and θ are the azimuth and elevation component of relative angle between LOS and heading direction of the transceiver. The state x_1 from Eq. (3.4) is omitted from the state vector and it is considered as a constant $\mathbb{C}_q$ in this work. Now the system output, which is derived from Eq. (3.6), can be expressed in terms of the state variables as

$$\mathbf{y}_k = \mathbb{C}_q g\left(\arccos\left(\cos x_{2,k} \cos x_{3,k}\right)\right). \tag{5.2}$$

The evolution of system can be expressed in the discrete time as:

$$\begin{bmatrix} x_{2,k+1} \\ x_{3,k+1} \end{bmatrix} = \begin{bmatrix} x_{2,k} + u_{2,k} \\ x_{3,k} + u_{3,k}, \end{bmatrix} \tag{5.3}$$

where $u_{2,k}$ and $u_{3,k}$ denote the control terms. The noise terms are omitted from the dynamics and the output equations for simplicity in the formulation.

5.2 A Triangular Exploration Algorithm

In this section, we discuss the details of the proposed alignment control algorithm that is termed as a *Triangular Exploration* algorithm. Given any initial condition $[x_{2,0}, x_{3,0}]^T$, we choose

$$\begin{bmatrix} u_{2,0} \\ u_{3,0} \end{bmatrix} = \begin{bmatrix} \delta\cos(\psi_0) \\ \delta\sin(\psi_0) \end{bmatrix}, \quad \begin{bmatrix} u_{2,1} \\ u_{3,1} \end{bmatrix} = \begin{bmatrix} \delta\cos(\psi_0 + \Delta\psi_0) \\ \delta\sin(\psi_0 + \Delta\psi_0) \end{bmatrix}$$

where $\delta > 0$ is the step-size, with ψ_0 chosen randomly from $(-\pi, \pi]$ and $\Delta\psi_0 = \pm\frac{2\pi}{3}$, where the sign is chosen randomly. The above initialization places the first three values of the states ($\mathbf{x}_0, \mathbf{x}_1$, and $\mathbf{x}_2$) in an equilateral triangle pattern. Now, define the control law $\mathbf{u}_k = [u_{2,k}, u_{3,k}]^T$ as follows:

$$\mathbf{u}_k = \begin{cases} \mathbf{x}_{k-1} - \mathbf{x}_{k-2}, & \text{if } \Delta\mathbf{y}_k \geq 0 \\ \mathbf{x}_{k-2} - \mathbf{x}_k, & \text{if } \Delta\mathbf{y}_k < 0, \end{cases} \tag{5.4}$$

where $\Delta\mathbf{y}_k = \frac{\mathbf{y}_k + \mathbf{y}_{k-1}}{2} - \mathbf{y}_{k-2}$. The control algorithm with the initialization ensures that the next, present and the previous states form vertices of on an equilateral triangle, as illustrated in Figure 5.2. Consider three points, illustrated in Figure 5.2 as, $\mathbf{x}_k$, $\mathbf{x}_{k-1}$ and $\mathbf{x}_{k-2}$ forming an equilateral triangle with side δ. There are only two possibilities for the next point $\mathbf{x}_{k+1}$; when $\Delta\mathbf{y}_k < 0$, the next point x^b_{k+1} comes back to the second previous point $\mathbf{x}_{k-2}$, and when $\Delta\mathbf{y}_k \geq 0$ the next point x^a_{k+1} completes the rhombus with the last three points. Further, the approximate gradient at point

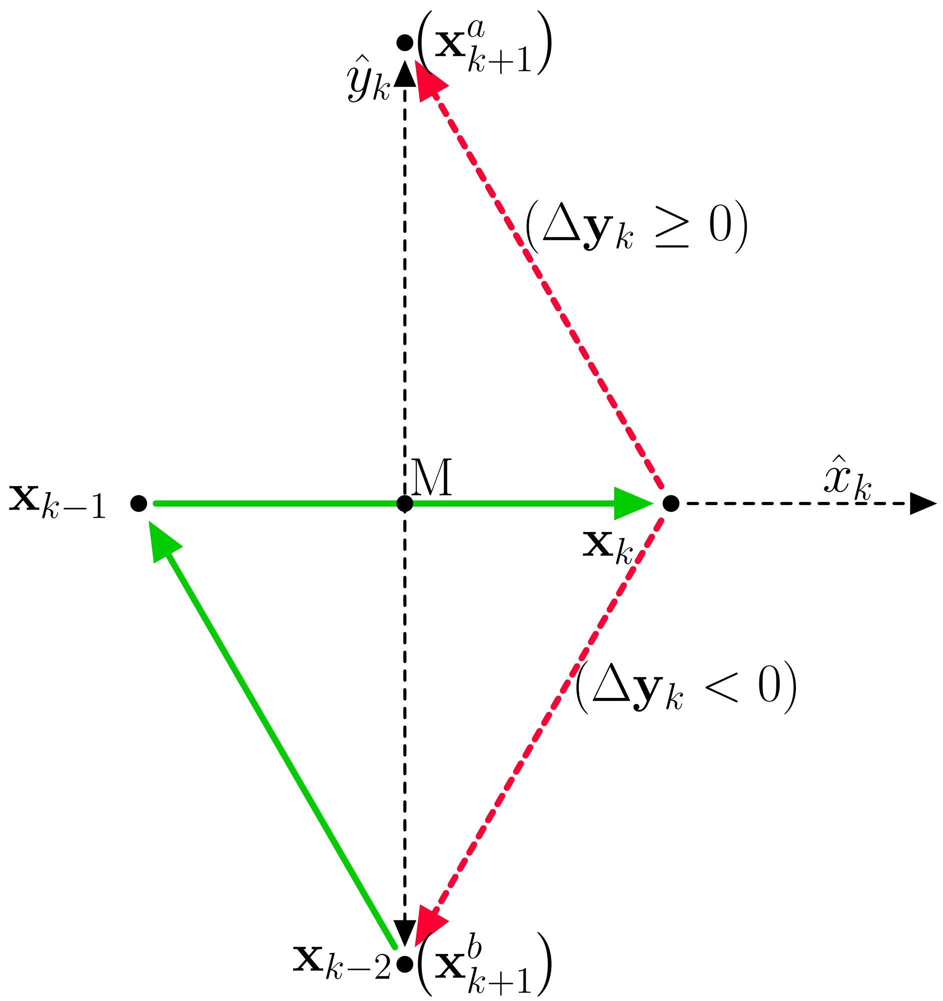


Figure 5.2: Illustration of the triangular-exploration method.

M_k (midpoint of segment joining $\mathbf{x}_{k-1}$ and $\mathbf{x}_k$) along the local coordinate axes $\hat{x}_k$ (direction of $\mathbf{x}_k - \mathbf{x}_{k-1}$) and $\hat{y}_k$ (direction orthogonal to $\hat{x}_k$), computed by finite difference is

$$\nabla\mathbf{y}_k = \begin{bmatrix} \frac{\mathbf{y}_k - \mathbf{y}_{k-1}}{\delta} \\ \frac{\mathbf{y}_k + \mathbf{y}_{k-1} - 2\mathbf{y}_{k-2}}{\sqrt{3}\delta} \end{bmatrix}.$$

The second component of the gradient is a scalar multiple of $\Delta\mathbf{y}_k$, and the next point always lies on the local $\hat{y}_k$ axis, which is orthogonal to the previous direction of motion, in the increasing direction of the component of the gradient in $\hat{y}_k$.

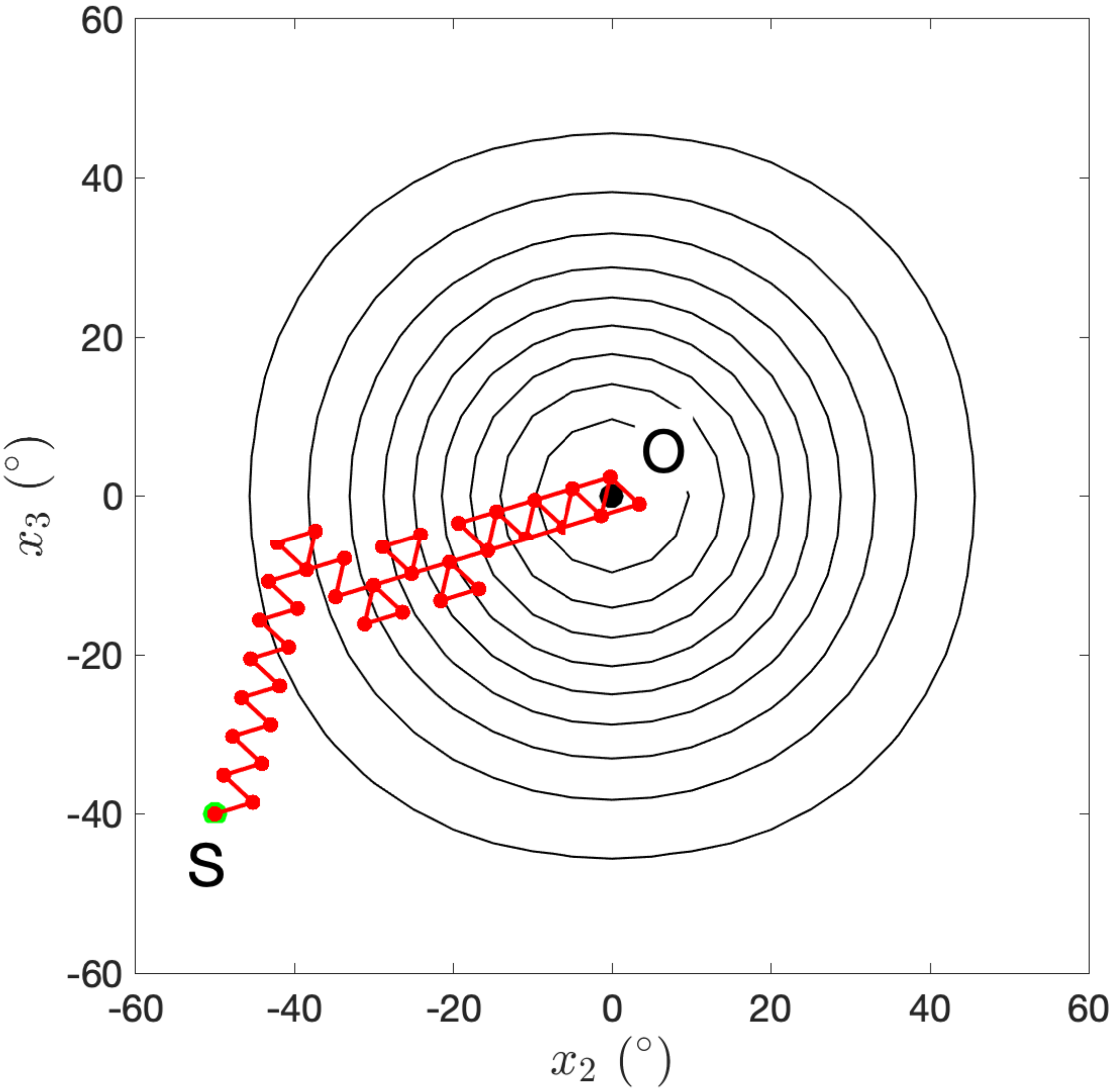


Figure 5.3: Illustration of the path of a simulation run of triangular-exploration algorithm.

5.3 Simulation Results

In this section, we first simulate the one-sided Triangular Exploration approach on MATLAB where a transceiver tracks a static light source. Then we simulate the bi-directional scenario on a system of two robots, the approach is implemented on each robot independently. We then compare the performance of the triangular exploration approach with the EKF based approach and the Extremum seeking approach.

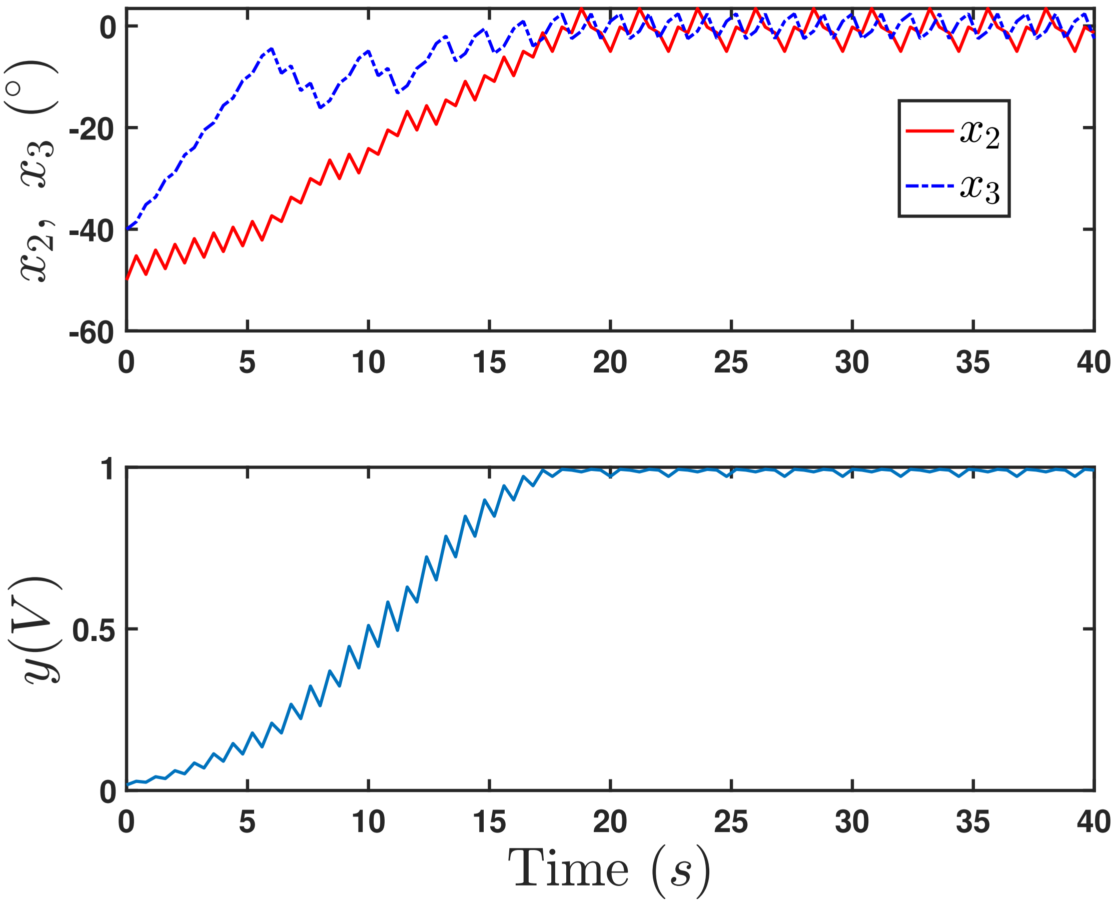


Figure 5.4: Illustration of the evolution of states and output for the simulation run of triangular-exploration algorithm corresponding to Figure 5.3.

Figure 5.4 illustrates the path of the states of the system, starting from the initial condition **S**. The path follows a zigzag pattern in the beginning for some time and then it corrects itself frequently to follow the gradient and reach the neighborhood of the origin. Figure 5.4 illustrates the evolution of states and the output of the system corresponding to the path in Figure 5.3. The states and the output converges to the neighborhood in about 20 s and then they continue to oscillate in the region. The output attains the steady state value of 1 V with some minor fluctuations.

Next, we consider the setup of two robots described in Section 3.3, where the robots are stationary and separated about two meters apart in the 3D space. Initially, none of the robots' heading direction is aligned with the LOS. The same setup is used to fairly evaluate the performance of

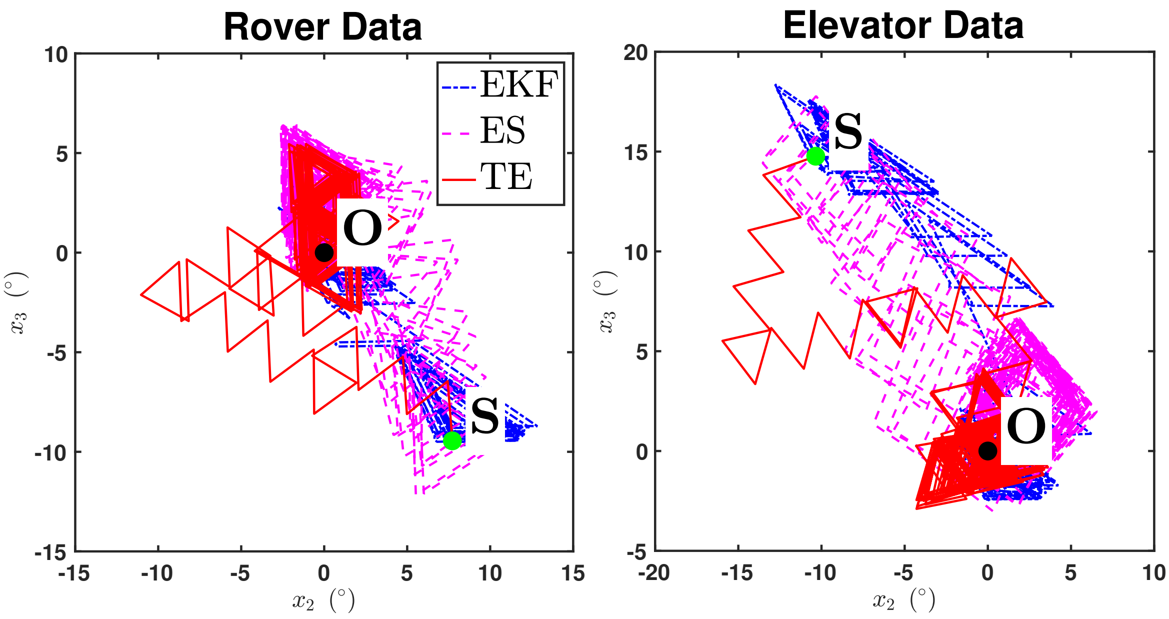


Figure 5.5: Illustration of paths of the states for the three algorithms for each robot, for a simulation run when the robots are stationary.

the Triangular Exploration (TE) Algorithm in comparison with the EKF and ES-based algorithms. Fig 5.5 shows the path of the states of each robots for the three algorithms. The states start from initial condition **S** and reaches a neighborhood of the origin **O**. It can be observed that for each of the robots, the path of TE approach initially follows a zigzag pattern and it gets corrected frequently such that it finally reaches a neighborhood around **O** and continuously oscillates within the neighborhood. Figure 5.6 illustrates the evolution of states and output for the three algorithms corresponding to Figure 5.5. It can be visually observed that the speed of convergence for the TE approach better than the other two approaches.

Next we compare the performance of the algorithms on repeated simulation runs over a range of distance and relative speed. We use the same simulation setup and consider the same performance metrics for evaluation as described in Section 3.3. Figure 5.7 shows the effect of increasing distance on the average intensity **I** and the average pointing error **E**. It is observed that the average error E for the TE approach stays relatively constant for the range of distances. For low distances the EKF

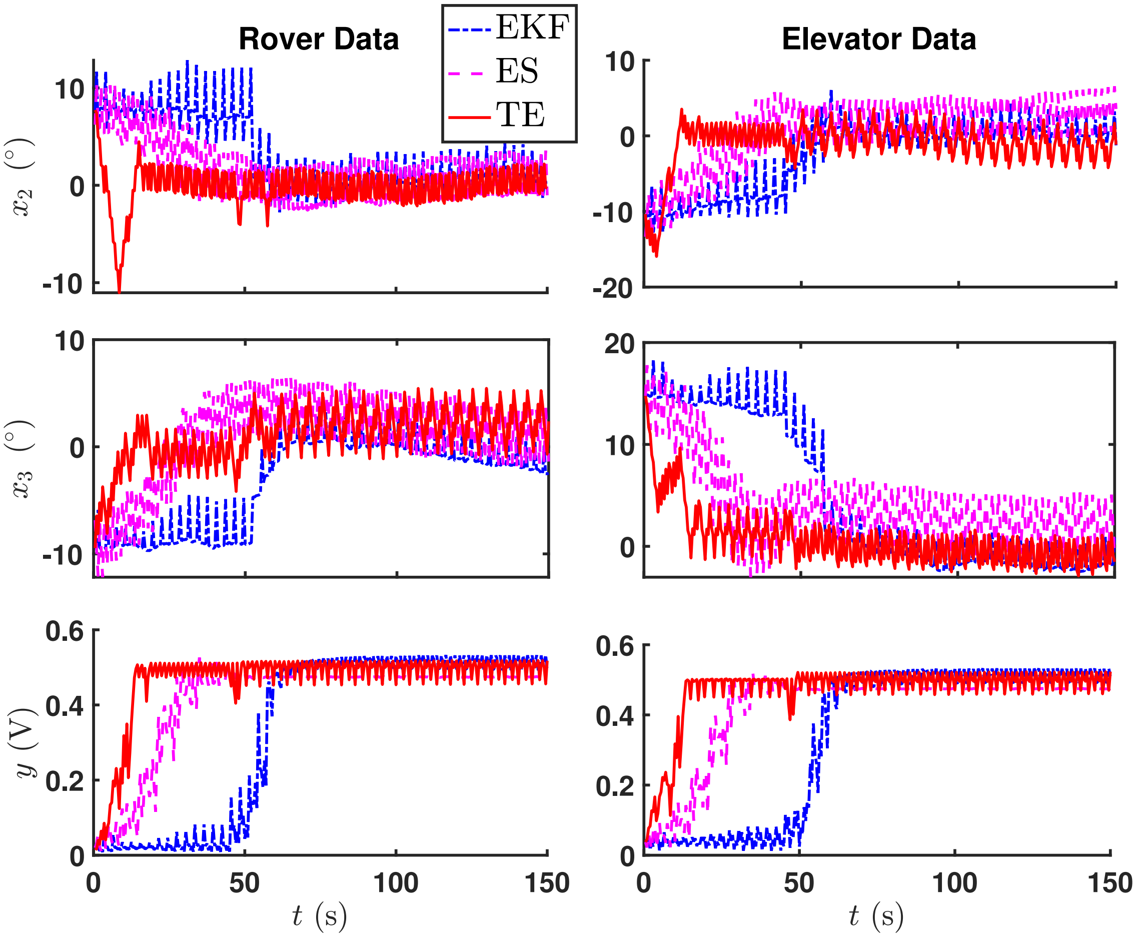


Figure 5.6: Illustration of evolution of the states and output for the three algorithms for each robot, for a simulation run corresponding to the Figure 5.5.

approach shows minimum error among the three approaches; however, the error of TE approach becomes the minimum beyond 4 m. The average intensity **I** of the TE approach shows a trend that is similar to **I** of the EKF, and is close to the maximum attainable intensity $\mathbf{I}_{LOS}$. These results validate the operational efficacy of the TE approach over a wide range of distance.

Figure 5.8 shows the average pointing error over a range of the relative speed v. The error for the TE approach stays constant and close to the error of the ES approach. The EKF approach has low error for lower range of speed; however, as the speed increases, the error starts increasing and becomes significantly high for higher speeds. Now, from the performance of the TE approach exhibited in Figure 5.7 and Figure 5.8, we can conclude that the TE approach is significantly better

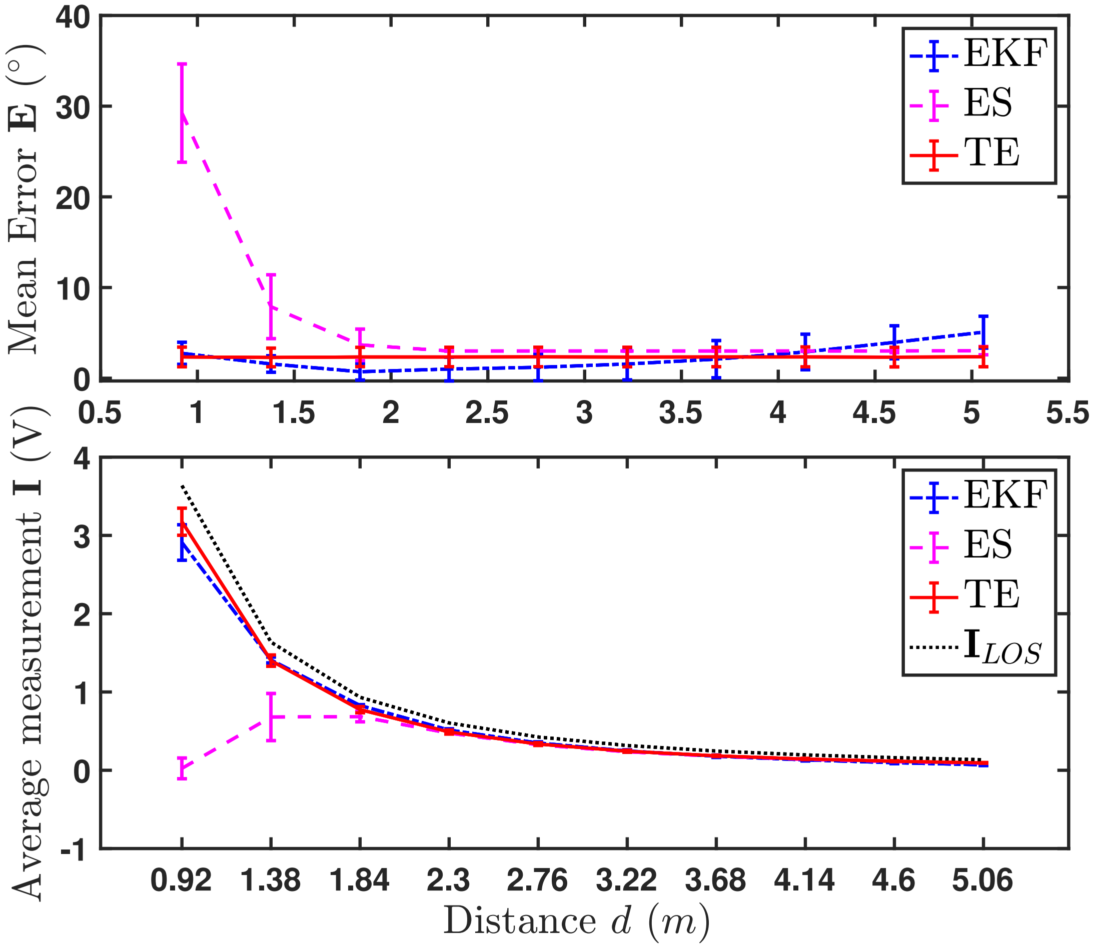


Figure 5.7: Tracking performance of the three algorithms in terms of average error **E** and average intensity **I** in simulation over a range of distances between the robots. The error bars denote the standard-deviation. The intensity at LOS ($\mathbf{I}_{LOS}$) is also shown for reference.

than the other two approaches in terms of its wide range of operation in both distance and speed.

5.4 In-air Experiment results

In this section, we evaluate the performance of the three algorithms on the experimental setup, which is discussed in Section 3.4. Figure 5.9 illustrate paths of the states of each robot for the three algorithm. The characteristics of the paths is similar to what is observed in simulation; however, the size of the steady-state neighborhood for the TE approach is bigger than what is observed in simulation. Additionally, for the elevator robot, the path of the TE approach initially goes away

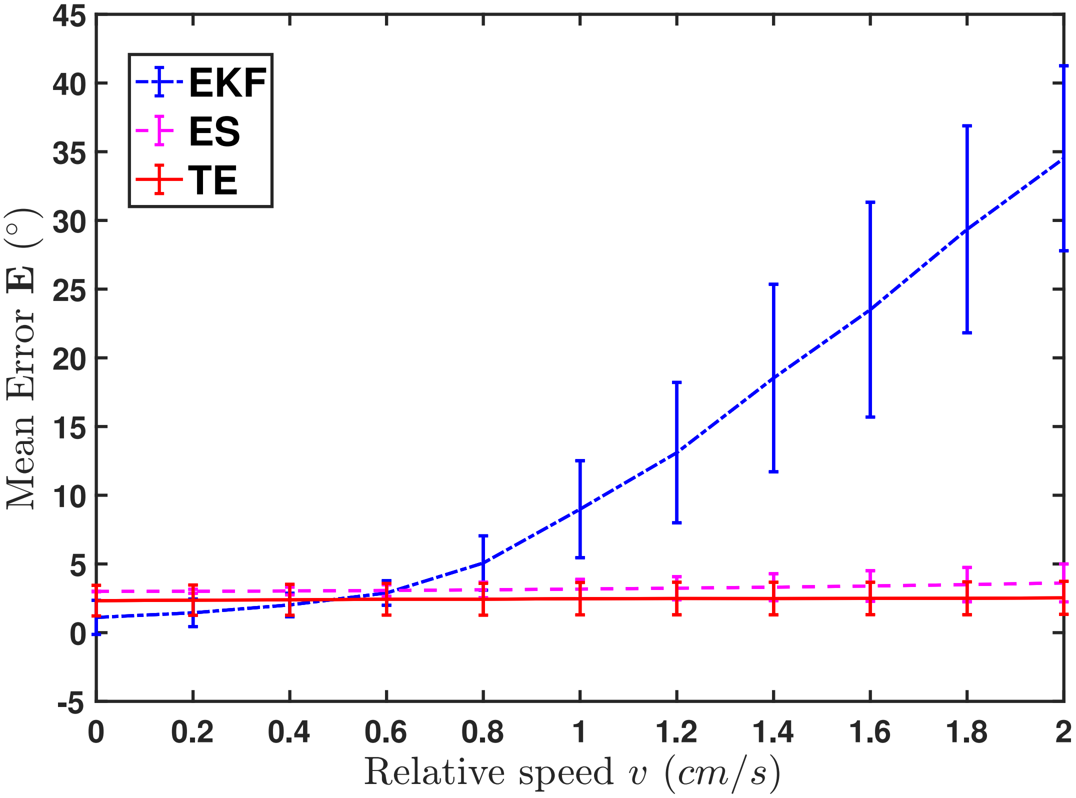


Figure 5.8: Illustration of tracking performance of the three algorithms in terms of average error **E** in simulation over a range of speeds of the robots.

from the origin and after few corrections, it gets steered to a neighborhood of the origin. Figure 5.10 shows the evolution of the states and the output corresponding to Figure 5.9. Here, it is observed that the speed of convergence for the TE approach is close to the speed of the ES approach.

Next, we consider the range of relative speeds of the robots. Figure 5.11 shows the plots of the average pointing error **E** over the range of speed v for the three algorithms. The qualitative characteristics of the plots are similar to what is observed in the simulation. Initially, at low speeds the EKF approach show minimum error and as the speed increases, due to the violation of the quasi-static assumption of the states, the EKF fails to converge and produces high tracking error. The error for the TE approach stays relatively constant through the range of the speed and stays minimum for mid-range and high range of speeds. For the ES approach, the error stays close to the error of the TE approach, but it starts increasing moderately with the increase of the relative speed.

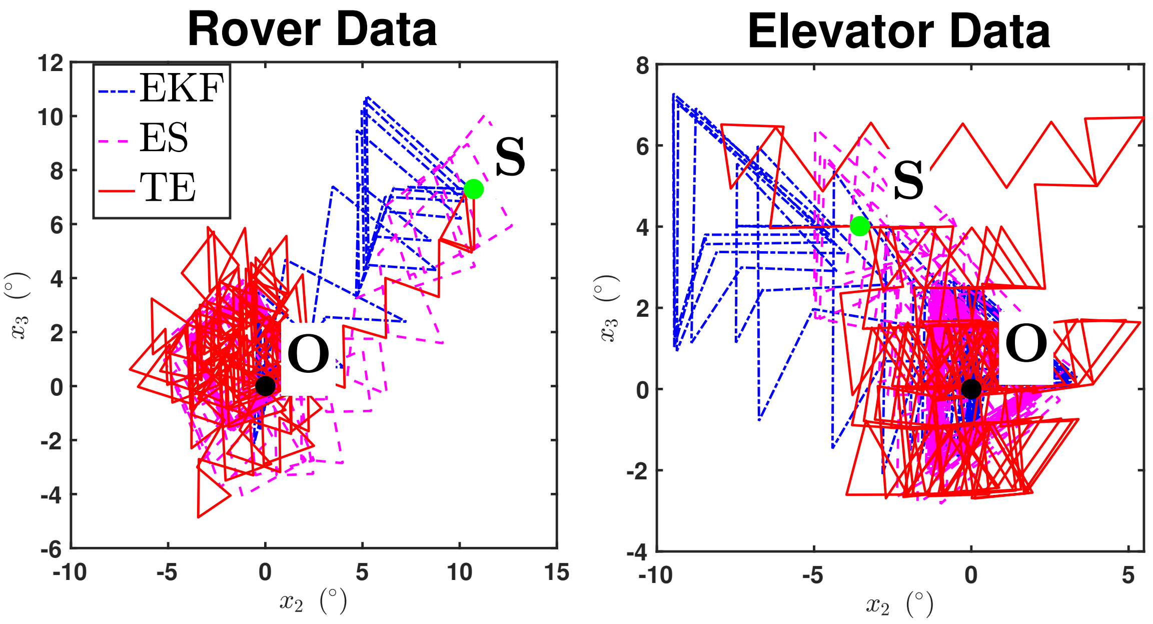


Figure 5.9: Illustration of paths of the states for the three algorithms for each robot, for experiment runs when the robots are stationary.

From the simulation and the experimental results, we establish that the TE approach shows optimal performance in terms of the mean tracking error over the range of speed and distance. Therefore, we used the triangular exploration algorithm for the alignment control in the underwater setup, which is discussed in the next section.

5.5 Underwater Experiments and Results

In this section, we present the experiments with simultaneous alignment control and communication that are performed in the real underwater scenario. First we present a uni-directional setup which comprises an underwater robot and an optical wireless joystick. Then we present a bi-directional scenario comprising of two-robots.

Figure 5.12 shows an underwater robot equipped with the transceiver module, described in Section 5.1, inside a transparent casing. It consists of three T100 thrusters from Blue Robotics: two for horizontal motion and the other one for vertical motion of the robot. The robot is kept neutrally buoyant for ease of operation. For on-board processing and computation, a BeagleBone™ blue

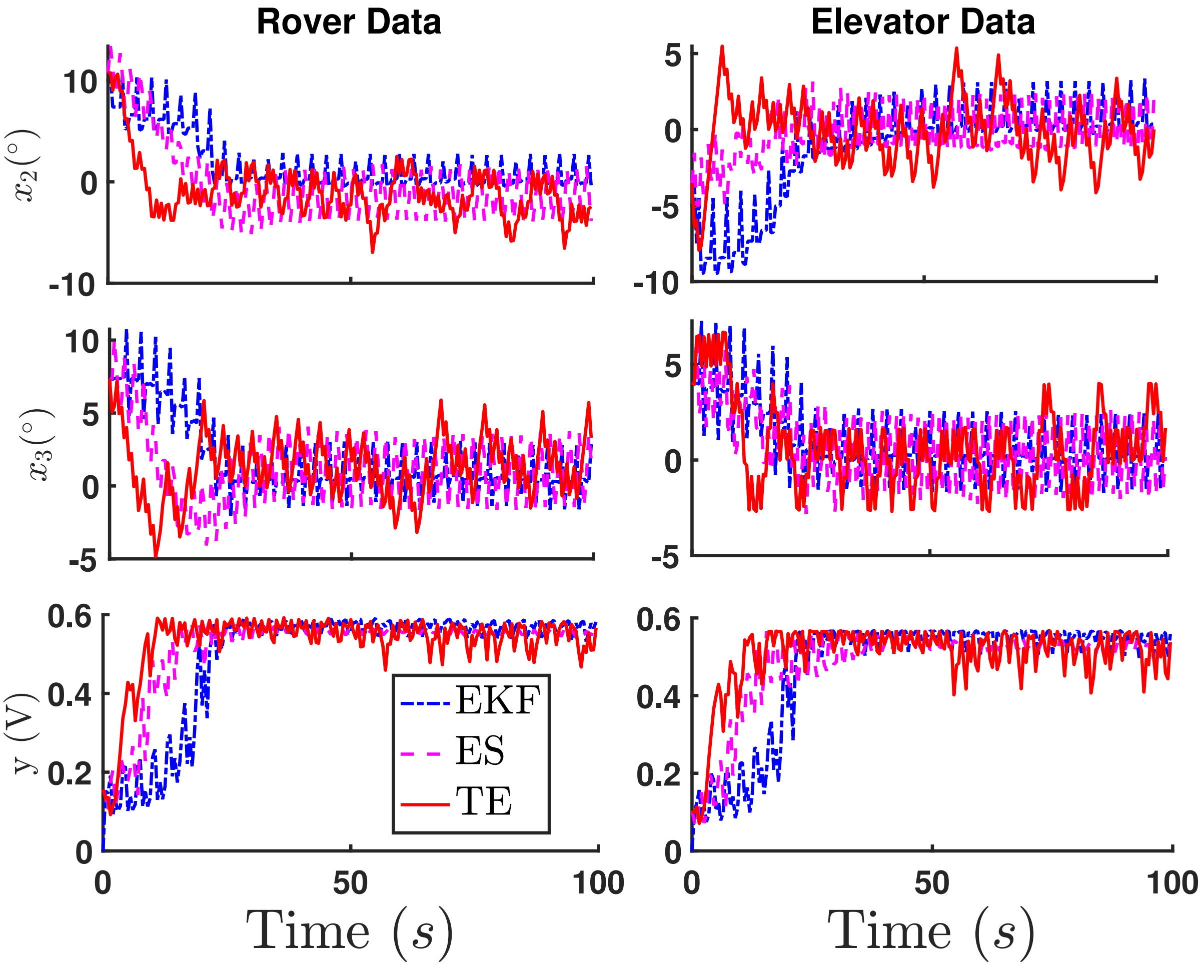


Figure 5.10: Illustration of evolution of the states and output for the three algorithms for each robot, for experiment runs corresponding to the Figure 5.10.

computer board is used.

Figure 5.13 shows an optical communication-based joystick controller, where a PS4 joystick is integrated with the LED communication circuitry. The joystick is inspired by *diver interface module* [22], where high-level commands were transmitted by a human scuba diver to a robotic fish using acoustic communication. The joystick in this work is intended to direct the aforementioned robot in the underwater scenario by sending commands *"go up"*, *"go forward"*, *"turn left"*, etc. The commands corresponding to the buttons pressed on the joystick are received by an on-board BeagleBone™ blue computer through a USB cable. The board then generates on-off keying (OOK) signals by the UART (Universal Asynchronous Receiver/Transmitter) protocol at a baud-rate of

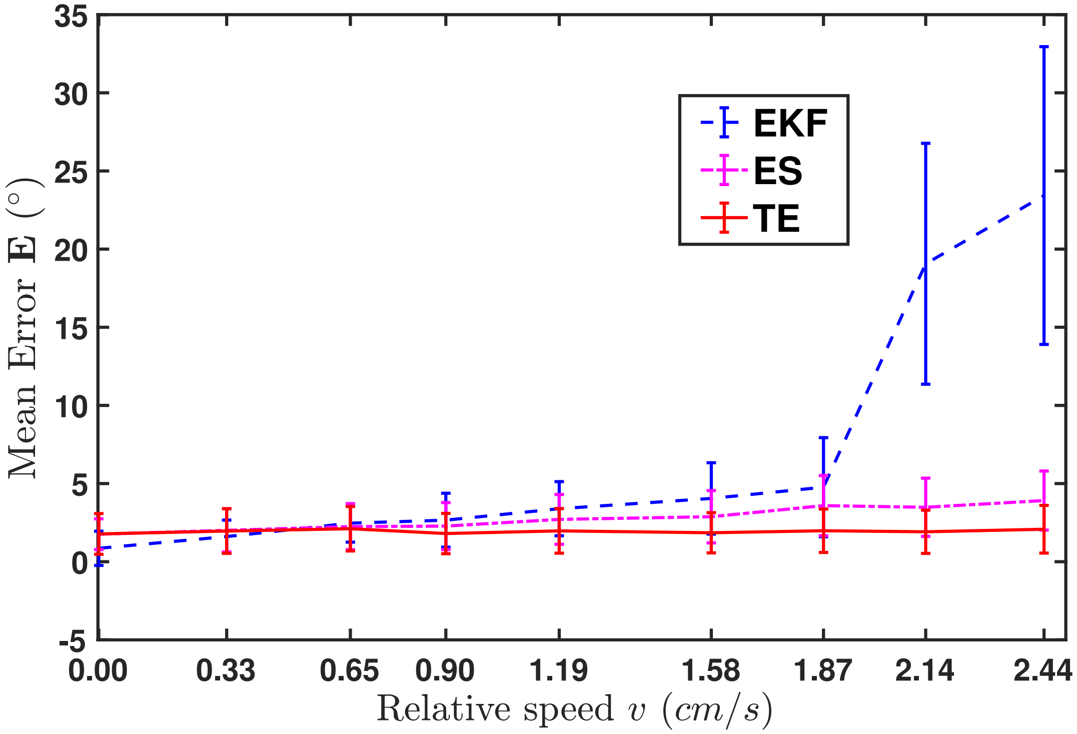


Figure 5.11: Tracking performance of the algorithms in terms of average pointing error **E** over a range of speeds of the robots in experiments.

$115,200$ bits per second (bps), which are then transmitted through the LED of the joystick-controller. In addition, a predefined *dummy* string of characters is always transmitted repetitively from the LED to generate a constant average light intensity that facilitates active alignment control at the receiver end of the robot.

On the robot's end, the optical signals are received by the photo-diode and the two components, the average value y, and the information signal, are passed on to the on-board computer, which then uses the triangular exploration algorithm to steer the transceiver to align with the LOS. Figure 5.14 shows the plots of the received signal intensity y along with bit-rates of reception and error rates over the course of an algorithm run. For this experiment, the robot is held steady underwater, and the LED-joystick is held about a meter away, pointing at the transceiver's location of the robot. The controller only transmits the dummy string, so that it is easier to compute the bit error rate on the other end. The received data-rate is computed by dividing the number of bits received by the length

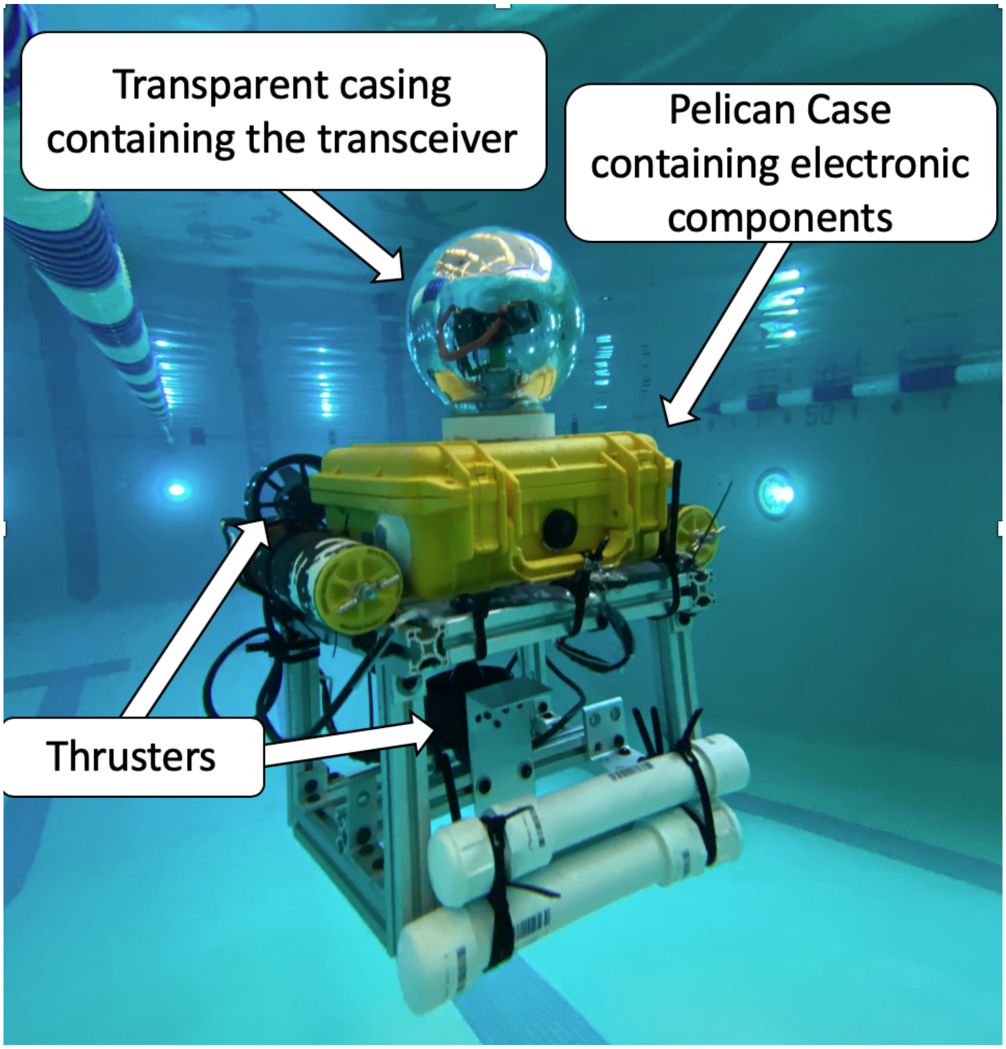

Transparent casing containing the transceiver
Pelican Case containing electronic components
Thrusters

Figure 5.12: An underwater robot equipped with the active transceiver.

of the iteration interval (0.5 s). Since the expected incoming string is known, the bit-error-rate is computed by calculating the hamming distance between the received and expected bit strings.

At the beginning of the experiment, the transceiver of the robot is pointing away from the LED-joystick, and hence the received signal signal-strength is too low (less than the threshold of the comparator) for enabling communication. As time progresses, the signal strength increases, and at about the 21 second mark, the receiver starts receiving some bits. However, a major portion of the bits has error at this point. Here, the signal strength must be comparable to the comparator threshold resulting in a high signal to noise ratio (SNR). Moving forward, at about 25 seconds, as the signal strength increases further, the bit error eventually drops to zero, and the received information

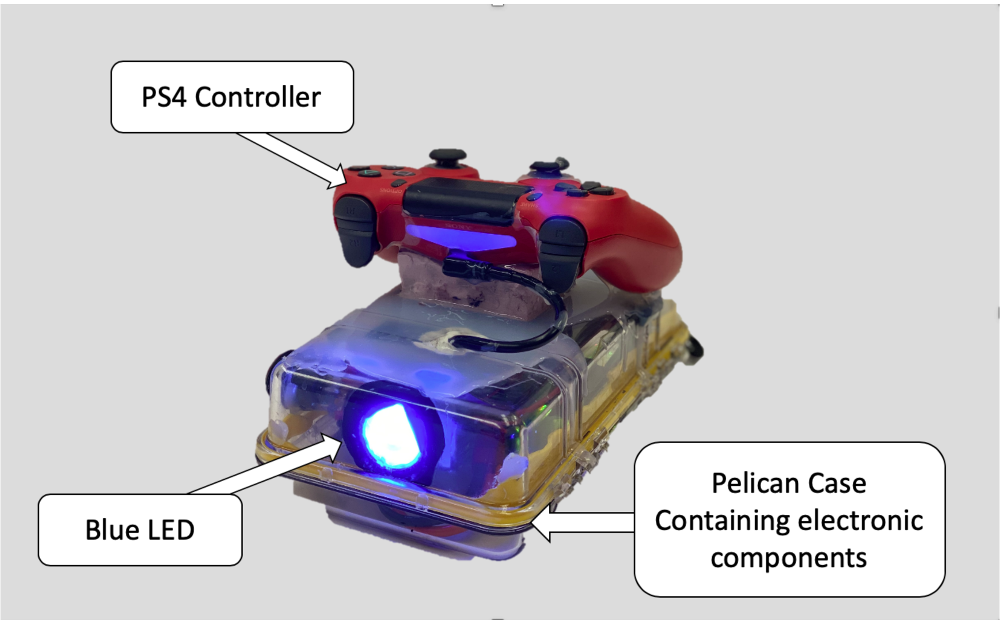


Figure 5.13: A uni-directional LED-communication based joystick controller designed for this work.

rate stabilizes to a value of about 100 kbps, which is close to the baud-rate of transmission. The difference between the rates is attributed to a parity bit, which is used for built-in error correction by the UART protocol. At about 30 seconds, the signal strength gets saturated around a constant intensity; at this point, it is also visually observed that the transceiver pointing direction oscillates around the LOS with the LED-controller.

The setup is further tested in a swimming pool facility at the Michigan State University. The LED-controller is made operational underwater by sealing inside a transparent plastic bag so that the buttons remain accessible, and the light of LED is not blocked. Figure 5.15 shows a human operator directing the robot to navigate inside the pool. The robot is commanded to move forward, backward, turn, and move upward/downward. It is to be noted that the human supervisor sometimes manually adjusts the pointing direction of the joystick to re-establish LOS with the robot's transceiver as the LOS gets disturbed due to fast relative motion between the robot and the controller.

Next, we present the bi-directional alignment and communication performance, for the underwater scenario, Figure 5.16 shows two copies of the underwater robot placed about 1.5 m apart in water. The transmission beams of both the robots are visible due to the scattering of light and

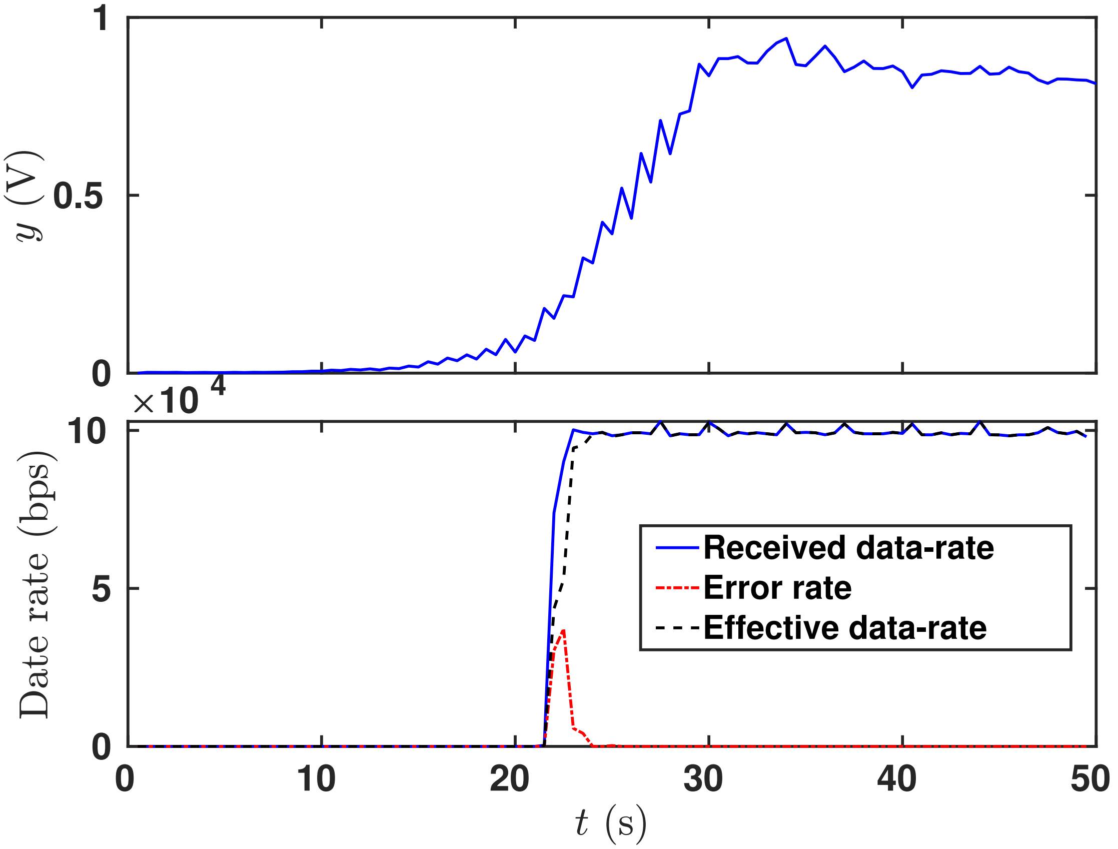


Figure 5.14: Evolution of light intensity and data rates for an experiment run of triangular exploration algorithm on the experiment setup. The received data-rate is correlated with the signal-strength.

hence they are considered as visual indicators of the alignment.

Figure 5.17 shows the plots of received signal strength, received data-rate and error-rate. In the beginning of the experiment, both of the robots are pointing away from the direction of LOS, and each of the robots is transmitting the dummy string from its transceiver at $57,600$ bps. From the plots, it can be observed that, initially the signal strength for both the robots was low (less than 0.1 V) and as the time progresses the strength increases and oscillates between 0.2 to 0.5 V. The received signal rate during these oscillation gets stable between 30 and 40 kbps, which is close to the two-third of the transmission rate. The difference here is due to four extra bits (two parity and two stop bits) per eight-bit sequence, which results in the effective data rate of $38,400$ bps. The observed magnitude of oscillations in light intensity is higher than what is observed in the *in-air*

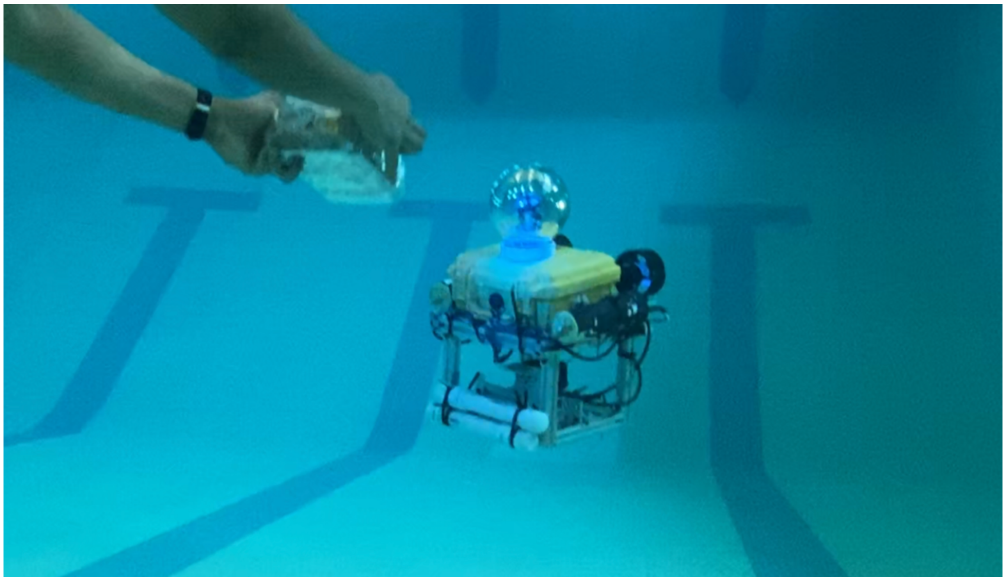


Figure 5.15: Underwater robot being commanded by a human operator using the LED-joystick controller inside a swimming pool.

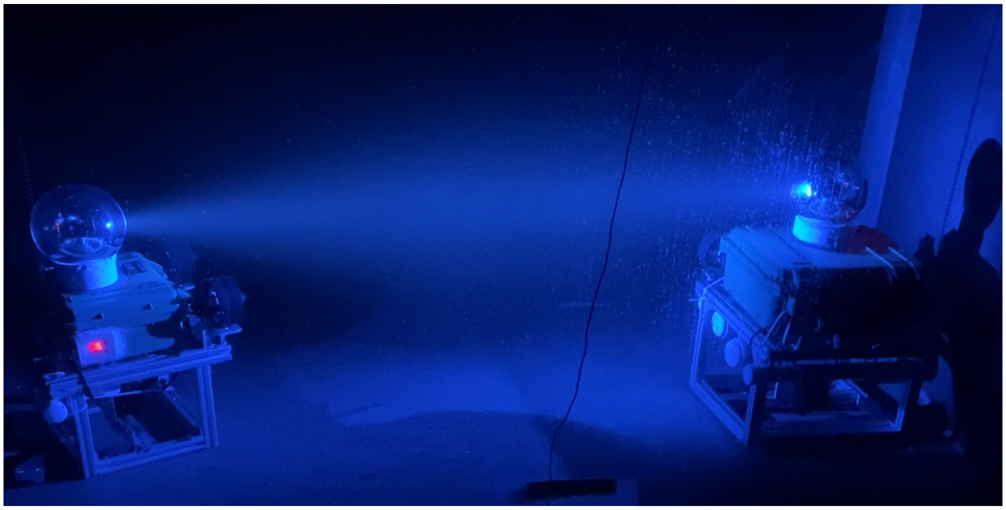


Figure 5.16: Setup of two underwater robots which are communicating and actively aligning with each other.

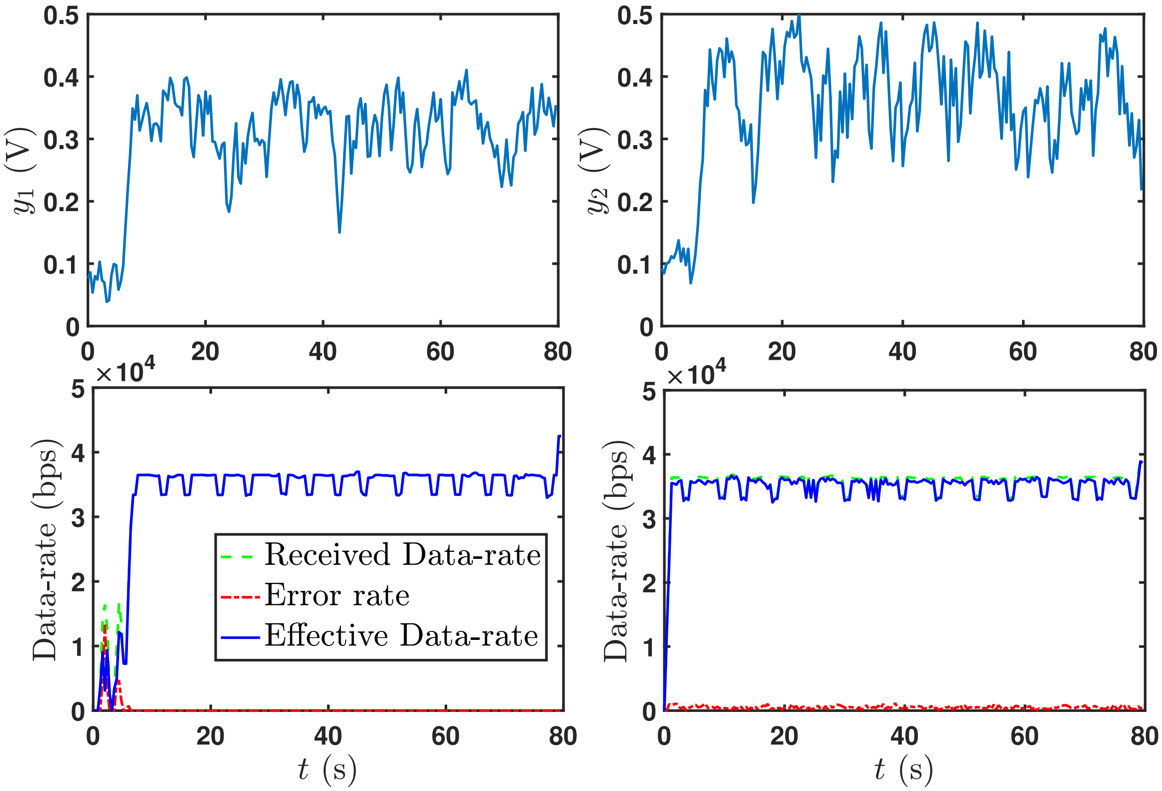


Figure 5.17: Illustration of light intensity measurements and data rates for an experiment run on the setup shown in Figure 5.16.

experiments. This difference can be attributed to change in light-intensity model, largely due to refraction caused by multiple changes of medium in the path of the transmitted beam in its path before reaching the photo-diode at the receiving end.

Figure 5.18 shows the setup of two robots in a swimming pool setting at a depth of 3.7 m. The setup is intended to perform the bi-directional alignment control and communication experiments. The robots maintained the LOS for some time but failed to maintain the alignment throughout the run. Possible reasons for the failure include the interference from ambient sunlight in the pool and the leakage of water inside one of the robots affecting the overall functionality of the robot.

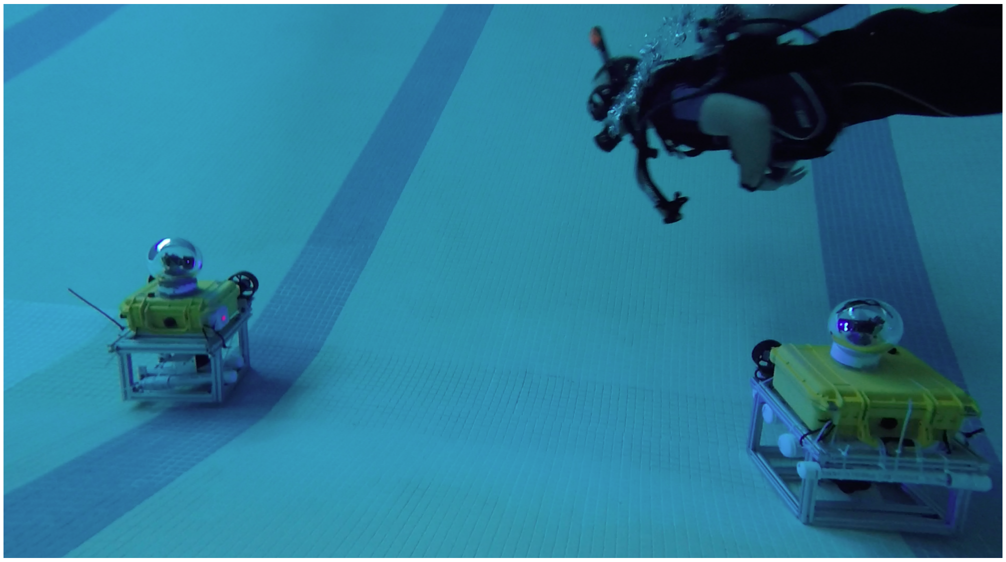


Figure 5.18: Setup of two underwater robots for bi-directional alignment and communication in the swimming pool.

5.6 Chapter Summary

In this work, we present a model-free active alignment control approach that is directly applicable to bi-directional setting. Given that the dependence of received light intensity on the local angles is unimodal, we propose a triangular-exploration algorithm for real-time alignment control to achieve and maintain LOS. The algorithm ensures that the last three points always form an equilateral triangle so that it has a good estimate of the local gradient, and it uses the last three measurements to decide its next step in the gradient direction while maintaining the equilateral triangle shape. The tracking performance of the TE approach is compared with the previously proposed EKF method and the ES control method in simulation and experiments. The TE approach is found to be optimal and effective in terms of its convergence speed for a wide range of relative speed and distance between the robots.

The efficacy of the overall system is also tested in underwater scenario where communication is performed simultaneously with the alignment control, and the dependence of data-rate on signal-strength is also demonstrated. First, an application of uni-directional system is demonstrated where a human operator wirelessly controlled a robot in an underwater scenario using an LED communication-based joystick. Then bi-directional alignment control and communication is shown between two underwater robots. Although the presented results are validated on the specialized hardware setup, the model-free and generic nature of the approach makes the results widely useful for scenarios of active alignment of beam signals.

CHAPTER 6

SUMMARY & FUTURE WORK

6.1 Summary

In this dissertation, a novel active-alignment control based LED communication system design was presented where each transceiver on an agent consisted of only a single photo-diode and a single LED. Starting with a simple 2D scenario, an EKF-based algorithm was proposed to estimate the relative orientation between the heading angle and the LOS direction, which was subsequently used for alignment control. A scanning technique was introduced to obtain successive intensity measurements that ensured the full observability of the underlying system. The approach was later extended to a 3D scenario with improvements in both the hardware and the algorithm. A new circular scanning technique was proposed for the 3D setting, where the amplitude of the scanning was modulated according to the alignment performance to achieve a sound trade-off between estimation accuracy, signal strength, and energy consumption. The efficacy of the approach was tested and verified against an extremum-seeking approach via simulation and experiments involving two robots with relative 3D motion. The results established the superiority of the EKF approach approach over the ES approach in terms of efficacy across a range of distances.

Moving forward, we formulated the bidirectional optical beam tracking problem as a two-agent discrete-time dynamical system, where the origin corresponds to the LOS and is the point of maximum output function for each of the agents. We proposed a model-free output feedback control law for a class of systems that follow certain assumptions with the constraint that the control command of an agent can depend only on the information accessible to that agent. Through rigorous analysis, we showed that for any initial condition, the proposed control law drives the physical system to an appropriately characterized neighborhood of the LOS in a finite number of steps. From simulation results, the proposed algorithm was shown to be superior than the two contending approaches based on the EKF and the ES, in terms of convergence speed, robustness to unknown disturbances, and handling large initial condition. The proposed approach was also validated on an experimental setup consisting of two robots in the presence of a constant external disturbance.

Finally, we proposed a model-free approach for bi-directional alignment control in the 3D setting. The approach always maintains an equilateral triangle shape in the angular space by its last three pointing directions, which gives a good estimate of the local gradient, which is used to take the next step in the increasing gradient direction while maintaining the triangle shape. The efficacy of the proposed triangular exploration approach was tested against the EKF and the ES approach in simulation and experiments and was found to be better in terms of higher convergence speed for a wide range of relative speed and distance between two underlying robots. The overall system was also tested in the underwater scenario where communication was performed simultaneously with the alignment control, and the dependence of data-rate on signal-strength was also demonstrated. First, an application of the uni-directional system was demonstrated where a human operator wirelessly controlled a robot in an underwater scenario using an LED communication-based joystick. Then simultaneous bi-directional alignment control and communication was shown between two underwater robots.

6.2 Future Work

For future work, additional experimental trials of simultaneous alignment and communications between two moving underwater robots shall be conducted to test the efficacy of our proposed approach in more practical scenarios and exploring the limit of the communication bandwidth for moving robots. It will also be of interest to provide analytical results and guarantees of convergence for the triangular-exploration approach on the four-dimensional (4D) system of two robots (two states of each robot).

Since the alignment approaches presented in this work do not rely on the information exchange between the communication parties, a future direction of this research could include exploration of algorithms that use some of the bandwidth of communication to exchange useful information between the communicating robots, that can be used to enhance the tracking performance. Furthermore, the framework of two robots can be extended to a multi-agent system that emulates a fleet of underwater robots on a mission. As the robots can communicate only in pairs based on their physical location and orientation of their transceivers, thus the overall network becomes intermittent and techniques like agreement protocol and formation control will be challenging to analyze and implement and would require exploration of advanced algorithms.

APPENDICES

APPENDIX A

GENERATION OF MOTOR COMMANDS

In this section, we discuss the details of generating the motor commands from control terms at each iteration. The motor commands represent the differences between the present states of motors, and their values at the next step. The details of the control terms for the EKF, the ES and the TE approach are provided in Eq. (3.17), Figure 3.6 and Eq. (5.4) respectively.

$$\begin{bmatrix} \Delta\phi_{M,k} \\ \Delta\theta_{M,k} \end{bmatrix} = \begin{bmatrix} \phi_{M,k+1} \\ \theta_{M,k+1} \end{bmatrix} - \begin{bmatrix} \phi_{M,k} \\ \theta_{M,k} \end{bmatrix}, \tag{A.1}$$

where $\phi_{M,k}$ and $\theta_{M,k}$ are the azimuthal and elevation angles of the motor-system at the kth iteration respectively.

Figure A.1 illustrates the relation between multiple coordinate-systems, which share a common origin, and are related by rotation matrices. The *mean* and *scan* terms are specific to EKF based approach and their details are included in Chapter 3, Section 3.2. For the extremum-seeking

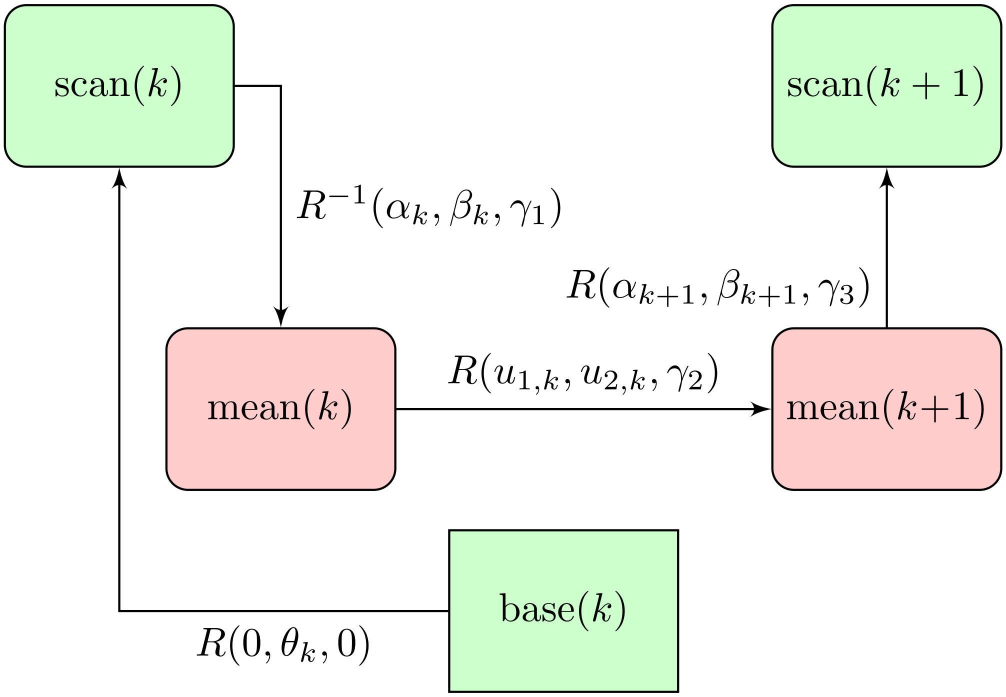


Figure A.1: Block diagram illustrating the relation between coordinate systems.

approach and the triangular-exploration approach, since the notion of *scan* is not applicable, the *mean* and *scan* coordinate systems of each iteration are same and can be merged together. The motor commands (angles $\Delta\phi_{M,k+1}$ and $\Delta\theta_{M,k+1}$) are the differences between the current pointing direction (x-axis of the scan(k) co-ordinate system) and the next pointing direction (x-axis of the scan($k+1$) co-ordinate system) measured in the coordinate system of base frame. The rotational transformation R is defined as,

$$R(\alpha, \beta, \gamma) = R_y(\beta) R_z(\alpha) R_x(\gamma),$$

where

$$R_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix},$$
$$R_z(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix},$$
$$R_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & -\sin(\gamma) \\ 0 & \sin(\gamma) & \cos(\gamma) \end{bmatrix}.$$

Now, at any given time iteration k, without the loss of generality, we can assume that the base(k)-coordinate system and the scan(k)-coordinate system have their z-axes aligned. Hence, the current scan has motor angles as $\begin{bmatrix} 0, \theta_k \end{bmatrix}^T$. It is assumed that the motor system has access to its elevation angle θ_k at every instant; however, the access to the azimuthal angle is not required. The next scan direction in the base-coordinate systems (**s**) can be computed by the following series of rotational transforms on the unit vector corresponding to the x-axis of the local scan($k+1$) coordinate systems:

$$\begin{aligned} \mathbf{s} = {} & R(0, \theta_k, 0) R^{-1}(\alpha_k, \beta_k, \gamma_1) \\ & R(u_{1,k}, u_{2,k}, \gamma_2) R(\alpha_{k+1}, \beta_{k+1}, \gamma_3) \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T. \end{aligned} \tag{A.2}$$

Additionally, in all of the local coordinate systems, there is a requirement that the local z axis should be in the xz plane. This constraint essentially means that none of the coordinate systems has any roll angle with respect to the base coordinate system. This constraint requires a roll correction at every stage, and hence the roll angle γ_1 can be computed by solving the following:

$$\begin{aligned}
&R(0,\theta_k,0)R^{-1}(\alpha_k,\beta_k,\gamma_1)_{(2,3)} = 0\\
&\Rightarrow \sin(\gamma_1)\cos(\theta_k)\cos(\beta_k) + \cos(\gamma_1)\cos(\theta_k)\\
&\quad \sin(\alpha_k)\sin(\beta_k) - \sin(\theta_k)\cos(\alpha_k)\sin(\beta_k) = 0.
\end{aligned}$$

Once γ_1 is known, γ_2 is computed by solving

$$R(0,\theta_k,0)R^{-1}(\alpha_k,\beta_k,\gamma_1)R(u_{1,k},u_{2,k},\gamma_2)_{(2,3)} = 0.$$

It has been noted that Eq. (A.2) is independent of γ_3, hence the computation of γ_3 is not required. Once the vector **s** is obtained, the motor angles for the next iteration are

$$\begin{aligned}
\phi_{M,k+1} &= \text{atan2}(-\mathbf{s}_z,\mathbf{s}_x), \text{ and}\\
\theta_{M,k+1} &= \text{atan2}(\mathbf{s}_y,\sqrt{\mathbf{s}_x^2+\mathbf{s}_z^2}).
\end{aligned}$$

Now using Eq. (A.1), one can finally compute the motor commands as

$$\begin{bmatrix}\Delta\phi_{M,k}\\ \Delta\theta_{M,k}\end{bmatrix} = \begin{bmatrix}\text{atan2}(-\mathbf{s}_z,\mathbf{s}_x)\\ \text{atan2}(\mathbf{s}_y,\sqrt{\mathbf{s}_x^2+\mathbf{s}_z^2}) - \theta_k\end{bmatrix}. \tag{A.3}$$

APPENDIX B

CALCULATION OF THE HEADING OFFSET ANGLE AND ITS DERIVATIVES

This appendix discusses the computation of ξ (and its gradient), which is introduced in Eq. (3.15). At any given instant, in the mean(k) coordinate system, $\xi(\cdot)$ computes the angle of the current pointing direction (spherical angles-(β, α)) with the LOS (spherical angles-(x_2, x_3)), which is the inverse cosine of the dot product of the unit vectors in these two directions.

$$\begin{aligned} \xi\left(x_2, x_3, \alpha, \beta\right) &= \arccos\left(\chi\left(x_2, x_3, \alpha, \beta\right)\right), \\ \chi\left(x_2, x_3, \alpha, \beta\right) &= \begin{bmatrix} \cos x_2 \cos x_3 \\ \sin x_3 \\ \sin x_2 \cos x_3 \end{bmatrix} \cdot \begin{bmatrix} \cos\beta \cos\alpha \\ \sin\alpha \\ \sin\beta \cos\alpha \end{bmatrix}. \end{aligned}$$

For notational convenience, we define $\tilde{\mathbf{x}}$ as $[x_2, x_3]^T$, $\tilde{\xi}$ as $\xi\left(x_2, x_3, \alpha, \beta\right)$ and $\tilde{\chi}$ as $\chi\left(x_2, x_3, \alpha, \beta\right)$. Next, we compute the gradient of $\tilde{\xi}$ with respect to $\tilde{\mathbf{x}}$:

$$\begin{aligned} \frac{\partial \tilde{\xi}}{\partial \tilde{\mathbf{x}}} &= \frac{1}{\sqrt{1-\chi^2}} \left[\tfrac{\partial \chi}{\partial x_2}, \tfrac{\partial \chi}{\partial x_3}\right], \\ \text{with } \frac{\partial \tilde{\chi}}{\partial x_2} &= -\cos\alpha \cos\beta \cos x_2 \sin x_3 + \\ &\quad \cos\alpha \sin\beta \cos x_2 \cos x_3, \\ \text{and } \frac{\partial \tilde{\chi}}{\partial x_3} &= -\cos\alpha \cos\beta \sin x_2 \sin x_3 \\ &\quad + \sin\alpha \cos x_2 - \cos\alpha \sin\beta \sin x_2 \sin x_3. \end{aligned}$$

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] M. Al-Rubaiai and X. Tan. Design and development of an LED-based optical communication system with active alignment control. In *2016 IEEE International Conference on Advanced Intelligent Mechatronics*, pages 160–165, July 2016.

[2] Mohammed Al-rubaiai. Design and development of an LED-based optical communication system. Master's thesis, Michigan State University, Aug 2015.

[3] Stephen B. Alexander. *Optical Communication Receiver Design*. SPIE Publications, Bellingham, 1997.

[4] Davide Anguita, Davide Brizzolara, and Giancarlo Parodi. Building an underwater wireless sensor network based on optical ommunication: Research challenges and current results. In *Sensor Technologies and Applications, 2009. SENSORCOMM'09. Third International Conference on*, pages 476–479. IEEE, 2009.

[5] Davide Anguita, Davide Brizzolara, and Giancarlo Parodi. Optical wireless communication for underwater wireless sensor networks: Hardware modules and circuits design and implementation. In *OCEANS 2010*, pages 1–8. IEEE, 2010.

[6] E. Biyik and M. Arcak. Gradient climbing in formation via extremum seeking and passivity-based coordination rules. In *2007 46th IEEE Conference on Decision and Control*, pages 3133–3138, Dec 2007.

[7] L. Briñón-Arranz, L. Schenato, and A. Seuret. Distributed source seeking via a circular formation of agents under communication constraints. *IEEE Transactions on Control of Network Systems*, 3(2):104–115, 2016.

[8] H. Brundage. Designing a wireless underwater optical communication system. Master's thesis, Massachusetts Institute of Technology, 2010.

[9] Lloyd Butler. Underwater radio communication. *Amateur Radio*, 1987.

[10] Wenqi Cai and Ibrahima N Doye. An extremum seeking control based approach for alignment problem of mobile optical communication systems. (May), 2020.

[11] Jongeun Choi, Songhwai Oh, and Roberto Horowitz. Distributed learning and cooperative control for multi-agent systems. *Automatica*, 45(12):2802 – 2814, 2009.

[12] Nikhil Chopra and Mark W. Spong. *Passivity-Based Control of Multi-Agent Systems*, pages 107–134. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[13] Salvador Climent, Antonio Sanchez, Juan Vicente Capella, Nirvana Meratnia, and Juan Jose Serrano. Underwater acoustic wireless sensor networks: Advances and future trends in Physical, MAC and routing layers. *Sensors*, 14(1):795, 2014.

[14] Brandon Cochenour, Linda Mullen, William Rabinovich, and Rita Mahon. Underwater optical communications with a modulating retro-reflector. *Ocean Sensing and Monitoring*, 7317(April 2009):73170G, 2009.

[15] M. Doniec, M. Angermann, and D. Rus. An end-to-end signal strength model for underwater optical communications. *Oceanic Engineering, IEEE Journal of*, 38(4):743–757, Oct 2013.

[16] M. Doniec and D. Rus. Bidirectional optical communication with AquaOptical II. In *Communication Systems (ICCS), 2010 IEEE International Conference on*, pages 390–394, Nov 2010.

[17] Peter G. Goetz, William S. Rabinovich, Rita Mahon, James L. Murphy, Mike S. Ferraro, Michele R. Suite, Walter R. Smith, Ben B. Xu, Harris R. Burris, Christopher I. Moore, Warren W. Schultz, Barry M. Mathieu, Kurt Hacker, Shad Reese, Wade T. Freeman, Steve Frawley, and Michael Colbert. Modulating retro-reflector lasercom systems at the Naval Research Laboratory. *Proceedings - IEEE Military Communications Conference MILCOM*, pages 1601–1606, 2010.

[18] R.M. Hagem, David V. Thiel, S.G. O'Keefe, A. Wixted, and T. Fickenscher. Low-cost short-range wireless optical FSK modem for swimmers feedback. In *Proceedings of 2011 IEEE Sensors Conference*, pages 258–261, Oct 2011.

[19] Frank Hanson and Stojan Radic. High bandwidth underwater optical communication. *Appl. Opt.*, 47(2):277–283, Jan 2008.

[20] Mahdi Jadaliha, Joonho Lee, and Jongeun Choi. Adaptive control of multiagent systems for finding peaks of uncertain static fields. *Journal of Dynamic Systems, Measurement, and Control*, 134(5), 07 2012. 051007.

[21] Sungho Jeon and Hiroshi Toshiyoshi. MEMS tracking mirror system for a bidirectional free-space optical link. *Applied Optics*, 56(24):6720, aug 2017.

[22] Robert K. Katzschmann, Joseph DelPreto, Robert MacCurdy, and Daniela Rus. Exploration of underwater life with an acoustically controlled soft robotic fish. *Science Robotics*, 3(16), 2018.

[23] A. Komaee. Cooperative stochastic control for optical beam tracking. In *2007 41st Annual Conference on Information Sciences and Systems*, pages 673–678, 2007.

[24] Arash Komaee, P.S. Krishnaprasad, and Prakash Narayan. Active pointing control for short range free-space optical communication. *Communications in Information and Systems*, 7(2):177–194, 2007.

[25] M. Krstic and K. B. Ariyur. *Multiparameter Extremum Seeking*, chapter 2, pages 21–45. John Wiley & Sons, Ltd, 2004.

[26] Naomi E. Leonard, Derek A. Paley, Russ E. Davis, David M. Fratantoni, Francois Lekien, and Fumin Zhang. Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in Monterey Bay. *Journal of Field Robotics*, 27(6):718–740, 2010.

[27] N. Li and J. R. Marden. Designing games for distributed optimization. *IEEE Journal of Selected Topics in Signal Processing*, 7(2):230–242, April 2013.

[28] Xiaoyan Liu, Suyu Yi, Ran Liu, Lirong Zheng, and Pengfei Tian. 34.5 m Underwater optical wireless communication with 2.70 Gbps data rate based on a green laser with NRZ-OOK modulation. In *2017 14th China International Forum on Solid State Lighting: International Forum on Wide Bandgap Semiconductors China, SSLChina: IFWS 2017*, volume 2018-January, 2018.

[29] Feng Lu, Sammy Lee, Jeffrey Mounzer, and Curt Schurgers. Low-cost medium-range optical underwater modem: Short paper. In *Proceedings of the Fourth ACM International Workshop on UnderWater Networks*, WUWNet '09, pages 11:1–11:4, New York, NY, USA, 2009. ACM.

[30] D. Meng, M. Fazel, and M. Mesbahi. Proximal alternating direction method of multipliers for distributed optimization on weighted graphs. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 1396–1401, Dec 2015.

[31] F.P. Miller, A.F. Vandome, and J. McBrewster. *Beer-Lambert Law*. VDM Publishing, 2009.

[32] B. J. Moore and C. Canudas-de-Wit. Source seeking via collaborative measurements by a circular formation of agents. In *Proceedings of the 2010 American Control Conference*, pages 6417–6422, 2010.

[33] Shigeki Muta, Takeshi Tsujimura, and Kiyotaka Izumi. Laser beam tracking system for active free-space optical communication. *Proceedings of the 2013 IEEE/SICE International Symposium on System Integration*, pages 879–884, 2014.

[34] A. Nedic and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, Jan 2009.

[35] P. Ogren, E. Fiorelli, and N. E. Leonard. Cooperative control of mobile sensor networks:adaptive gradient climbing in a distributed environment. *IEEE Transactions on Automatic Control*, 49(8):1292–1302, 2004.

[36] J Onathan F H Olzman. All-optical retro-modulation for free-space optical communication. 26(4):5031–5042, 2018.

[37] Hassan Makine Oubei, Changping Li, Ki-Hong Park, Tien Khee Ng, Mohamed-Slim Alouini, and Boon S. Ooi. 23 Gbit/s underwater wireless optical communications using directly modulated 520 nm laser diode. *Optics Express*, 23(16):20743, 2015.

[38] C. Pontbriand, N. Farr, J. Ware, J. Preisig, and H. Popenoe. Diffuse high-bandwidth optical communications. In *OCEANS 2008*, pages 1–4, Sept 2008.

[39] K. Reif, S. Gunther, E. Yaz, and R. Unbehauen. Stochastic stability of the discrete-time extended Kalman filter. *IEEE Transactions on Automatic Control*, 44(4):714–728, Apr 1999.

[40] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 edition, 2003.

[41] I.C. Rust and H.H. Asada. A dual-use visible light approach to integrated communication and localization of underwater robots with application to non-destructive nuclear reactor inspection. *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 2445–2450, May 2012.

[42] Giuseppe Schirripa Spagnolo, Lorenzo Cozzella, and Fabio Leccese. Underwater Optical Wireless Communications: Overview. *Sensors (Basel, Switzerland)*, 20(8):2261, apr 2020.

[43] J. S. Shamma and G. Arslan. Dynamic fictitious play, dynamic gradient play, and distributed convergence to nash equilibria. *IEEE Transactions on Automatic Control*, 50(3):312–327, March 2005.

[44] J.A. Simpson, B.L. Hughes, and J.F. Muth. Smart transmitters and receivers for underwater free-space optical communication. *Selected Areas in Communications, IEEE Journal on*, 30(5):964–974, June 2012.

[45] P. B. Solanki, M. Al-Rubaiai, and X. Tan. Extended Kalman filter-aided alignment control for maintaining line of sight in optical communication. In *2016 American Control Conference*, pages 4520–4525, July 2016.

[46] P. B. Solanki, M. Al-Rubaiai, and X. Tan. Extended kalman filter-based active alignment control for led optical communication. *IEEE/ASME Transactions on Mechatronics*, 23(4):1501–1511, Aug 2018.

[47] P. B. Solanki, S. D. Bopardikar, and X. Tan. A bidirectional alignment control approach for planar led-based free-space optical communication systems. In *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 1949–1955, 2020.

[48] P. B. Solanki and X. Tan. Experimental implementation of extended Kalman filter-based optical beam tracking with a single receiver. In *2016 IEEE International Conference on Advanced Intelligent Mechatronics*, pages 1103–1108, July 2016.

[49] P. Bhanu Solanki and X. Tan. Extended Kalman filter-based 3D active-alignment control for LED communication. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4481–4488, May 2018.

[50] P.B. Solanki, S.D. Bopardikar, and X. Tan. Active alignment control-based led communication for underwater robots. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1692–1698, Oct 2020.

[51] Pratap Bhanu Solanki and Xiaobo Tan. Extended Kalman Filter-Aided Active Beam Tracking for LED Communication in 3D Space. In *Dynamic Systems and Control Conference*, volume 2, 10 2017. V002T04A009.

[52] Gokhan Soysal and Murat Efe. Kalman filter aided cooperative optical beam tracking. *Radioengineering*, pages 242–248, June 2010.

[53] M. Stojanovic. Recent advances in high-speed underwater acoustic communications. *IEEE Journal of Oceanic Engineering*, 21(2):125–136, April 1996.

[54] Sheri N. White, Alan D. Chave, and George T. Reynolds. Investigations of ambient light emission at deep-sea hydrothermal vents. *Journal of Geophysical Research: Solid Earth*, 107(B1):EPM 1–1–EPM 1–13, 2002.

[55] Tsai Chen Wu, Yu Chieh Chi, Huai Yung Wang, Cheng Ting Tsai, and Gong Ru Lin. Blue laser diode enables underwater communication at 12.4 gbps. *Scientific Reports*, 7(January):1–10, 2017.

[56] Russell B. Wynn, Veerle A.I. Huvenne, Timothy P. Le Bas, Bramley J. Murton, Douglas P. Connelly, Brian J. Bett, Henry A. Ruhl, Kirsty J. Morris, Jeffrey Peakall, Daniel R. Parsons, Esther J. Sumner, Stephen E. Darby, Robert M. Dorrell, and James E. Hunt. Autonomous underwater vehicles (AUVs): Their past, present and future contributions to the advancement of marine geoscience. *Marine Geology*, 352:451 – 468, 2014. 50th Anniversary Special Issue.

[57] K. Yoshida and T. Tsujimura. Tracking control of the mobile terminal in an active free-space optical communication system. In *2006 SICE-ICASE International Joint Conference*, pages 369–374, Oct 2006.