UNIVERSITY OF CALIFORNIA
RIVERSIDE

Joint Visible Light Communication and Navigation via LEDs

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

by

Dongfang Zheng

December 2014

Dissertation Committee:

Dr. Jay A. Farrell , Chairperson
Dr. Matthew Barth
Dr. Amit Roy-Chowdhury

The Dissertation of Dongfang Zheng is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

This dissertation could not be finished without the help and support from many people. Here I would like to express my deepest gratitude to them.

First of all, I would like to thank my advisor, Dr. Jay A. Farrell, for his support, guidance, encouragement and patience during my Ph.D. years at UC Riverside. I still remember the first time I talked with him, and the first email I received from him accepting me to participate this research project. I am thankful not only for his instructive advice and useful suggestions on my research, but also for his encouragement and patience for me. His words of encouragement after my first presentation in a group meeting still make me feel strength today.

I also want to thank the committee members Prof. Matthew Barth and Prof. Amit Roy-Chowdhury in my final defense. The talking with Prof. Matthew Barth after my oral exam encouraged me to continue my research following the ideas. The questions from Prof. Amit Roy-Chowdhury in the group meeting really helped me a lot in learning new technologies. I want to thank Prof. Jie Chen for providing me this opportunity to study at UC Riverside. I also want to thank Prof. Christian Shelton for his strict attitude and helpful suggestions during my oral defense. I also want to thank Prof. Anastasios Mourikis for his help in the group meetings and classes. I am grateful to the help from Dr. Gang Chen and Prof. Albert Wang when I was preparing the demonstrations. Dr. Chen's expert knowledge on the hardware helps me solve lots of problems of the experimental platform.

I would like to express my gratitude to my co-works Yiming Chen, Lingchao Liao, Tian Lang, Zening Li, Bo Bai, Kaiyun Cui, Rathavut (Biggie) Vanitsthian and Dr. Paul Miller for their hard work. Without their help, it is impossible for me to have

the work completed. I appreciate for the help from the former students Dr. Anning Chen, Dr. Lili Huang, Dr. Qichi Yang, Dr. Yiqian Li and Dr. Arvind Ramanandan. They gave me so much help both in research and everyday life in my first few years in the US.

I also want to thank my classmates Mingyang Li, Haiyu Zhang, Sheng Zhao and Sarath Suvanah in the lab WCH 396 for discussing with me the technical problems in research. Thanks the current students Xing Zheng, Hongsheng Yu, Shukui Zhang and Paul Roysdon for making our lab a great place to learn and work. I am also grateful for having the friends Yulin Zhang, Dr. Ergude Bao, Ming Wang, Zhen Qin, Zongyu Dong, Lunshao Chai, Lin Wang, Kuili Fang during my Ph.D. years at UCR. They brought me so much happiness in the years passed.

At last, thanks my parents, my wife, for their unconditional love, encouragement and support to me. They always gave me hope when I was depressed, without which this work would never have been possible. They are always the most important people in my life.

To my parents and wife, for their love and support.

ABSTRACT OF THE DISSERTATION

Joint Visible Light Communication and Navigation via LEDs

by

Dongfang Zheng

Doctor of Philosophy, Graduate Program in Electrical Engineering
University of California, Riverside, December 2014
Dr. Jay A. Farrell , Chairperson

Visible light communications (VLC) based on illuminative LED lamps are attracting increasing attentions due to its numerous advantages. Compared with conventional lamps, LED illumination sources have long operational life, low power consumption, and are mechanically robust. Therefore LEDs are expected to be the next generation of lamps that will be widely installed to replace conventional lamps. Besides, LEDs can be modulated at very high-speeds, which allows the possibility of simultaneously providing communication while illuminating. The light modulation frequency is sufficiently high that it is undetectable by the human eye, yet detectable by arrays of photodiodes. Considering these facts, a system can be designed to receive (using a single photodiode or a camera) and analyze LED signals. Furthermore, such a system should be able to facilitate position estimation tasks either for people or vehicles.

Two kinds of sensors - single photo-detector and array photo-detector (camera or linear array) C can be used to detect the LEDs. The single photo-detector approach is used to detect the existence and the specific identity of an LED. The array photo-detector approach measures the angle-of-arrival of the LED signal. A single photo detector provides the simplest hardware approach and could communicate with LED's at a very

high data rate, but only offers the most basic level of positioning. Cameras provide much more informative position measurements; however, there are challenges to achieving high rate LED-to-camera data communications due to the current hardware architectures of cameras. Alternatively, linear PD arrays allow high sample rates, high accuracy and low cost. This dissertation will investigates the issues (observability, extrinsic parameter calibration, and vehicle state initialization) related to implementation of a positioning and communications system built on a linear optical sensor array.

The VLC based navigation methods require recovering the LED ID from a sequence of photo-detector scans. This ID will help the data association in the navigation system or the data communication in the VLC system. Recovering LED data (ID) requires identifying each LED's projection position and on-off status in each photo-detector scan. Identifying the LED projection is challenging because: 1) Clutter and noise corrupt the measurements; 2) The LED status will be "off" in some frames; 3) The predicted projection location sequence depends on the estimated vehicle state trajectory, which is uncertain. This dissertation presents two new methods determining the most probable data and LED position sequences simultaneously, using Bayesian techniques by maximizing posterior probabilities. The first method is based on Viterbi algorithm and developed under the assumption that the frame rate of the array sensor is high relative to the rover motion bandwidth. The second one is based on multiple hypothesis tracking (MHT) techniques with no assumption assumed. Both the two methods are analyzed theoretically and demonstrated by experimental results.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Visible light communications (VLC) based on illuminative LED lamps are attracting increasing attentions due to its numerous advantages. Compared with conventional lamps, LED illumination sources have long operational life, low power consumption, and are mechanically robust. It is viewed as the next generation lamps in the future. Fig. 1.1 shows the evolution of the illuminating lamps and their luminous efficacy comparison. Besides, LEDs can be modulated at very high-speeds, which allows the possibility of simultaneously providing communication while illuminating. The light modulation frequency is sufficiently high that it is undetectable by the human eye, yet detectable by arrays of photodiodes. For example, their fast switching rates ($> 300MHz$) enables LED's installed for illumination to also be used for communication and positioning [38]. Considering these facts, a system can be designed to receive (using a photo-detector) and analyze LED signals. Furthermore, such a system should be able to facilitate position estimation tasks either for people or vehicles. The multi-

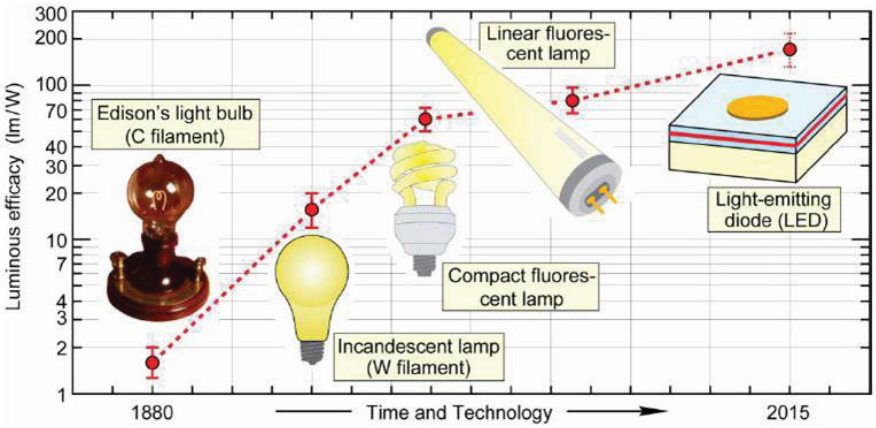


Figure 1.1: The evolution of illuminating lamps.

purpose operations of LED alleviate installing specific infrastructures for each purpose.

There is a growing need for position and navigation solutions other than Global Navigation Satellite System (GNSS), especially indoors or in urban environment where GPS does not function well. The indoor navigation applications include automated guided vehicles for office tasks, emergency guidance systems, equipment location determination, shopping assistance, and indoor navigation for the visually impaired [78]. The outdoor applications like positioning systems based on street light or stop lights can be a helpful compensate for GNSS. Besides, using LEDs as the detecting features could fully take advantage of their unique IDs [61, 85]. Compared with other data association algorithms such as Mahalanobis distance [53] and template matching [15], using the unique ID will be much more precise and reliable. Imagining that all the illuminating lamps indoors and outdoors are replaced by LEDs, the navigation and communication system built based on it will require installation of no other additional infrastructures.

The visible light communication (VLC) requires line-of-sight (LOS) communication channel [39]. That is, only when the LED sources are in the sight of receiver (photodiode or image sensor) that their signal has chances to be detected. Due to this

fact, VLC becomes more difficult when the receiver is moving. Taking camera for example, if it is installed on a vehicle that is traveling around, the LEDs' positions in the image are not fixed, then it is impossible to control the camera sampling a specific pixel to recover the data. On the other hand, searching each whole image to firstly find the LED's position before extracting the signal would make the communication inefficient. This difficulty in VLC encourages us the idea that combines the communication and navigation together. The navigation system estimates the position of camera, and further the LEDs' coordinates in the image are predicted. The navigation system also gives the error covariance of the predicted LED position. With this knowledge, the communication system could focus on a much smaller ellipse region in the image to find the LED and extract the signal. Communication requires knowledge of which LED ID's to expect for efficiency of demodulation and knowledge of where each LED is expected to be in the image for efficiency of image processing. Accurate navigation combined with an LED map enables both aspects.

1.2 Problem Statement

Even though the LOS requirement somehow increases the communication difficulty, it could be greatly useful for navigation purpose. For navigation purpose, a single photodiode provides the simplest hardware approach that could detect LED data at rates exceeding $300MHz$, but only offers the most basic level of positioning. The use of image sensors (cameras) makes it possible to detect the accurate direction of the incoming vector from a LED to the image sensor, but only samples at a low frame rate (about $30Hz$). A challenge in LED-based navigation system is how to improve the data transmission rate and the image processing efficiency while maintaining the capability

for accurate navigation. The large number of pixels (usually several mega-pixels) in the image sensor limits the frame rate. This dissertation investigates the ability of a linear array to preserve the cameras precise position measurement and simultaneously achieve high data rates.

Any kind of photo-detector based visible light communication and positioning systems require recovering the LED ID from a sequence of photo-detector scans. This ID will help the data association in the navigation system or the data communication in the VLC system. Recovering the LED data from the photo-detector's measurements is not so straightforward because: 1) Clutter and noise corrupt the measurements; 2) The LED status will be "off" in some frames; 3) The predicted projection location sequence depends on the estimated vehicle state trajectory, which is uncertain. To make matters worse, the LED projections in the sensor frame moves when the rover moves. The challenges of recovering the LED data in the sensor frames will be discussed and the potential solutions are also introduced.

1.3 Contributions

There are several contributions in the areas of Visible Light Communication, Navigation, and Computer Vision:

- Discussed the characteristics of VLC-based navigation system. Compared it with the other navigation method such as GPS, WiFi, radio and vision based methods.
- Proposed a new kind of sensor - linear array to overcome the shortcomings of photodiode and camera. The linear array will improve the data transmission rate and the image processing efficiency while maintaining the capability for accurate navigation result when the rover moves in a 2D plane.

- Described the basic physical structure of the linear array, and discussed its initialization and calibration, and analyzed the observability of the navigation system based on it.

- Analyzed the problems of the LED data recovery when using camera or linear array, and reviewed the existing methods.

- Proposed and analyzed two new methods based on Viterbi algorithm and multiple hypothesis tracking (MHT) algorithm, respectively. The Viterbi-based algorithm tries to recover the most probable LED path when the rover is stationary or the moving bandwidth is much smaller than the frame rate. The MHT-based algorithm recovers several possible LED paths other than one.

- Applied these two algorithms in our rover platform and tested the results.

1.4 Organization

This dissertation is organized as follows:

Chapter 2 introduces the background and the related issues in the VLC-based communication and navigation system. Chapter 3 proposes a new kinds of sensor - linear array to overcome the shortcomings of photodiode and camera. The basic physical structure will be described, and its initialization and calibration will be introduced, and the observability of the navigation system based on it will be analyzed. Chapter 4 analyzes the problems met in the LED data recovery processes by the camera or linear array. This chapter proposes a Viterbi-based algorithm to solve the data recovery problem when the photo-detector is stationary or moving with a bandwidth much lower than that of the frame rate. Chapter 5 further more proposes a more robust LED

data recovery algorithm which could be applied to the case that no matter the sensor is stationary or moving. Chapter 6 gives the conclusions of all the topics in this dissertation and discussed the future work.

Chapter 2

Background

2.1 Visible Light Communication (VLC)

Using visible light for data transmission is not a new idea. People started using smoke to send messages from thousands of years ago. The emperors in ancient China used the beacon-fires to call in the troops around the country when their capital was invaded. The fishermen in many places today are still using lighthouses to find their destinations on the sea. Today, the bandwidth and application areas of the visible light communication system are greatly expanded due to LED's many useful characteristics. Communication systems using photodiodes, cameras or linear arrays have been introduced in [58, 51, 4, 84]. Their applications include indoor information broadcasting system via ceiling lamps [44, 28], and communication systems between cars on road via car headlights, and so on. Using visible light to transmit music has been demonstrated in [48].

The number of LED lights in our surrounding is growing quickly as the prices are declining, which can be seen from Fig. 2.1. Now the LEDs are used as stop lights,

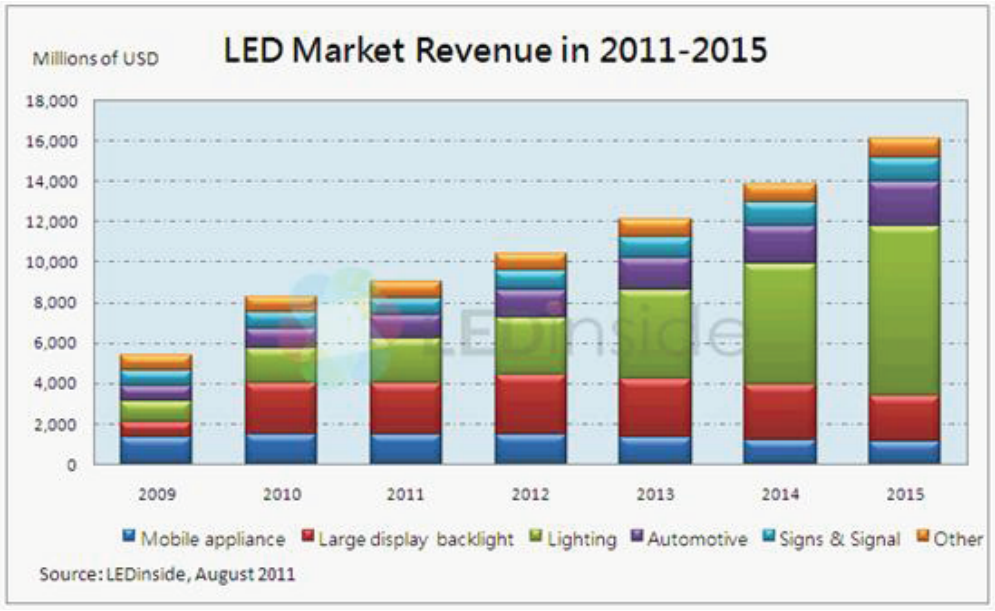


Figure 2.1: LED market revenue in 2011-2015

road lamps, and guiding lights in the airport, see Fig. 2.2 and 2.3. The indoor LED illuminating lights are also used more and more widely. This fact provides the opportunities for visible light communications to compensate or replace the radio-wave wireless communications indoors and outdoors. Compared with the radio-wave communication, the visible light part of the electromagnetic spectrum is still unlicensed and unregulated, and the power consumption of LED is lower than other radio wave transmitters [31]. Besides, the visible light has inherent security due to the line-of-sight (LOS) requirement of visible light communications. The light in room cannot be received outside because it will be blocked by the wall. On the other hand, the LOS requirement limits the VLC communication distance to 1 to 100 meters which is shorter than the radio-wave communication distance [32]. It also increases the complexities of receiving LED data when the photo-detector is moving, since different LEDs come and leave the photo-detector's field-of-view (FOV) and their projections on the photo-detector are also moving. However, the LOS property is useful for estimating the photo-detector's position and attitude,

which will be introduced in the following section.

(a) Stop lights.

(b) Street lights and airport guiding lights.

Figure 2.2: LED stop lights and street lights.

Figure 2.3: Airport LED guiding lights.

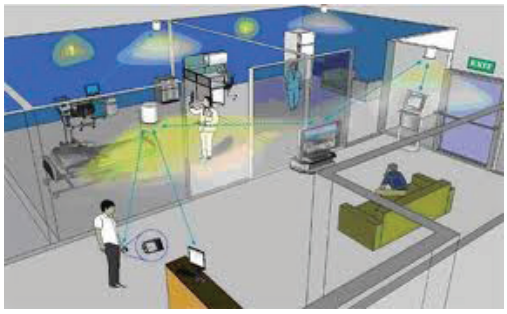
2.2 Visible Light Positioning (VLP)

As we have briefly introduced in Chapter 1, the LOS property could provide information of the photo-detector's relative position and rotation with respect to each detected LED. Once the global position of each LED transmitter is known, the photo-detector's global position will be able to be calculated. Such VLC-based navigation systems have utility in a variety of environments, especially indoors or in urban envi-

ronments, where the current generation of GNSS technologies does not function well. Indoor applications include emergency guidance systems, equipment location, personnel location, indoor navigation for the visually impaired, and automated guided vehicles for office tasks [78]. In the near future, it is even possible to help mobile users to obtain their real-time position information by receiving the LED data using the camera on smart-phone. For outdoor applications, the largest market would relate to a variety of automotive applications including mobile collision avoidance system, automatic guided vehicle, and so on. One most important advantage of VLC-based navigation systems is that the unique ID of each LED transmitter could help solve the data association problem. Navigation systems using the communication between LEDs' and photo-detectors have been discussed in [85, 86, 84]. Fig. 2.4 illustrates the indoor visible light communication and positioning system used in smart-phone or shopping cart. Fig. 2.5 illustrates the airport guidance system which determines the airplane's position by the LED guiding lights, and mobile collision avoidance system which is able to compute the relative position of a vehicle with respect to its vicinities by detecting the LEDs installed on the vehicles.

Fig. 2.6 gives the comparison between different navigation technologies. Here we assume that VLC belongs to the vision based methods, which will be explained later. From this figure, we will find that the VLC method will be much more accurate than that of using cell phone signal, Wi-Fi, and GPS.

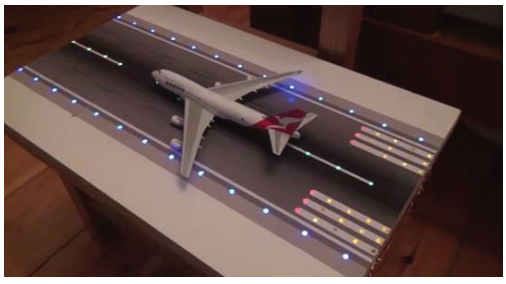
For navigation purpose, a single photodiode provides the simplest hardware approach that could detect LED data at rates exceeding 300MHz [38], but only offers the most basic level of positioning. This is basically a form of finger-printing [78, 27].



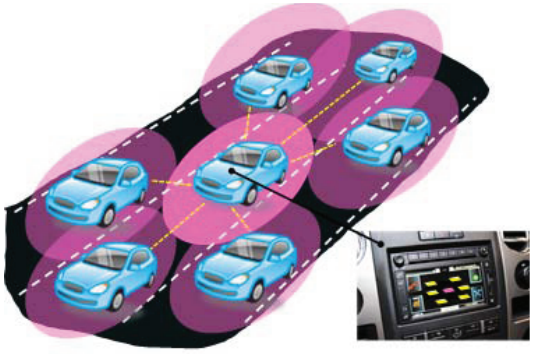
(a) Indoor visible light localization and communication systems using smart-phones and computers.

(b) Shopping assistant in the super markets.

Figure 2.4: Indoor applications.



(a) Airport guidance system using LED lamps.



(b) Mobile collision avoidance system by detecting the LEDs installed on the the vehicles.

Figure 2.5: Outdoor applications.

	Technology	Principle	Range Accuracy	Veh. Cost	Rdwy. Cost	Req'd Advances
	GPS	TOA	10 m	Low	Bases, comms	Modernization
	DGPS		1 m	Low	Bases, comms	Modernization
	CPDGPS		0.01 m	High	Bases, comms	Modernization
Terrestrial Radio Navigation	Pseudolites	TOA, TDOA	0.01 m	Med	High	> First generation
	-- Cell Phone	RSSI	km's	Low	Existing	
	-- Cell Phone	TOA, TDOA	100's m	Low	Existing	Physical layer timing
	-- TV - digital	TOA	10's m	??	Bases, comms	??
	-- Radio AM Analog	TOA	10's m	Low	Existing	??
	-- Radio FM Analog	TOA	??	Low	Existing	??
	-- Radio Digital	TOA	??	Low	Existing	??
	-- Packet Radios	TOA, TDOA	km's	Low	High	Physical layer timing
Feature Based	Vision	AOA		Low-Med	Feature Mapping	Feature Robustness
	Radar	AOA, RTOA	0.1 m	Low-Med	Feature Mapping	Feature Robustness
	Lidar	AOA, RTOA	0.01 m	High	Feature Mapping	Feature Robustness

Figure 2.6: Comparison between different navigation methods (Source: [79]).

Given a map of LED locations, when the photodiode through signaling knows that it detects the i-th LED, it can look up that LEDs location, and know that it is in the same vicinity. If multiple LEDs are detected, then the estimated location can be further refined. This approach is rather binary. Either it knows its position, or not. A camera provides a much more informative measurement for navigation purposes, by detecting the LEDs' positions in the image to form angle-of-arrival measurements. The camera also spatially separates the LEDs' signal so that they do not interfere. However, a camera's limited frame rate (up to thousands of frames per second) limits the rate of high speed communication. Compared with a single photodiode and the camera, the linear array could simultaneously provide high sampling rate and accurate estimation results. A linear array is a set of photodiodes that form one or several straight lines to receive the light signal. The number of pixels in a linear array is much smaller than that in a camera, since the linear array could be viewed as a small part of the camera sensor. The linear array preserves the sensitivity to angle-of-arrival relative to a single axis, so it is useful when the sensor moves in a 2D plane. The comparisons between

the photodiode, linear array, and camera are listed in Tab. 2.1. The analysis of the navigation and communication using a linear array, as well as the simulation results will be presented in Chapter 3. Due to the localization accuracy, we are more interested in the navigation system using photo-detector arrays (camera or linear array).

(a) Photodiode (b) Linear array (c) Image sensor

Figure 2.7: Photo-detectors

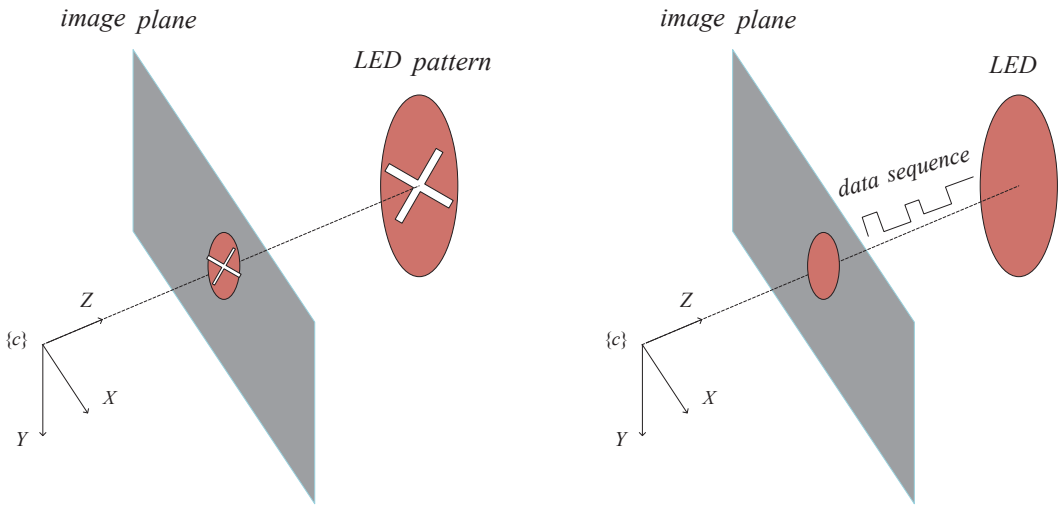
2.3 LED Detection and Data Recovery

Generally, there are two data transmission methods using LEDs: pattern based method and binary based method. The pattern based method uses an array of LEDs as the transmitter. Each time the LEDs with status "on" together represent a symbol which will be recognized by the photo-detector. The binary based method sends message only using the transmitter's "on" and "off" status, so that it could only send one bit each time. By comparison, the pattern based method has more communication efficiency but requires complex image processing technology to recognize the pattern. These two methods are illustrated in Fig. 2.8, where the photo-detector is an image sensor.

In this dissertation, we mainly focus on the binary based method so that the

Table 2.1: Comparison between photo-detectors

Photodiode	Linear array	Camera
>100MHz	100MHz per pixel	1KHz per frame
LED vicinity Not linearizable	Lateral projection u Linearly related to yaw and position error orthogonal to LOS vector	Lateral projection u, v Linearly related to yaw, pitch and position errors orthogonal to LOS vector
Inexpensive (<$1)	Inexpensive (∼$10)	Expensive (>$10k)
No image processing	No image processing	Reqs. image processing
All LEDs utilize same PD	Laterally separated LEDs use distinct PDs	Each LED uses a separated pixel (PD)



(a) Pattern based communication

(b) Binary based communication

Figure 2.8: LED data transmission methods

transmitters can be single LEDs. The LED data is modulated using as on-off keying (OOK) scheme [40]. Recovering the LED data requires extracting the "on" and "off" sequence of the LED from a record of consecutive scans. The turned on LEDs are point light sources with their projections bright blobs in a two-dimensional image, or bright segments in a linear array. The measured LEDs in a camera images and linear array are illustrated in Fig. 2.9.

(a) LED projection in the image

(b) LED projection in the linear array

Figure 2.9: The measured LEDs in the image and linear array.

For two-dimensional image measurements, various feature detectors and descriptors, including corner and blob detectors, have been developed to find and describe the features in the image. Both these existing corner and blob detectors can be used to search candidate LED projections in the image. Other simpler techniques like searching LEDs based on the pixel value or color can also be used, which are especially useful for linear array measurements where the existing corner or blob detector cannot be used. The LED detection methods mentioned above will extract all the features in the image besides LED's, so we cannot determine which ones are due to LEDs in a single image without any projection location information. For typical vision-aided navigation sys-

tems, in order to find the desired features (LEDs) in the image, their search regions are predicted based on the navigation state to narrow the searching area and reject unexpected measurements. The search region of each LED in the image is the inner of an ellipse defined by the residual covariance. However, due to the clutter measurements and noise, there still can be multiple potential measurements appearing in this search region. This can be the case even when the LED is off.

2.4 Related Work

The Visible Light Communications Consortium (VLCC) has been in established in 2003 aiming to publicize and standardize the visible light communication technology [59, 54]. The communication between the photodiode and a single white LED has been introduced in [74, 75, 46, 44, 80, 1]. In [44], up to 300Mb/s data rate can be realized using OOK modulation scheme, while a gross transmission rate of 513Mb/s is achieved in [80]. The communication between the photodiode and a single LED has been demonstrated by UC-Light lab. The music and video have been transmitted through VLC in the demonstration, and even the internet link can be connected by VLC. Toy cars equipped with LEDs exchange short messages is demonstrated in [16]. The communication between the photodiode and multiple LEDs are also discussed in [45, 49, 18]. Literature [49] uses on-off keying link to communicate with 16 LEDs simultaneously achieving an overall bandwidth of 25MHz. An OFDM (orthogonal frequency division multiplexing) modulation scheme is proposed for VLC in [1].

Using image sensor to communication with LEDs is another choice, which has advantage of spatially separating different LED sources to prevent interferences. An image sensor with an in-pixel demodulation function for detecting modulated LED light

is proposed and demonstrated in [60]. The method that decodes the LED data from a sequence of image is proposed in [64]. The random accessing characteristic of CMOS image sensor provides the opportunity of visual-MIMO (multiple-input multiple-output) where optical signal from multiple LEDs are received by multiple pixels in the CMOS image, which is introduced in [6]. The visual-MIMO demo that employs a LED array and a high speed camera is introduced in [76]. Another advantage of image sensor is the ability to detect LED patterns. Literature [62, 50] discussed a LED pattern based communication method by dividing the LED panel into regions to show different visual patterns. Literatures [5, 4, 58] proposed a hierarchical coding scheme to modulate the LED array and realized this method in the road-to-vehicle communication.

Even though the VLC based positioning research is still at a preliminary stage [66], there have been various methods proposed and implemented in many literatures. Due to its short range property (less than tens of meters), most proposed VLC based positioning methods are used indoors. These indoor positioning methods can be found in literatures [43, 83, 71, 29, 66, 65, 41, 42, 82, 87, 63]. Despite the difficulty, there are still some literatures like [69, 7] developed VLC based outdoor positioning methods, both of which detect the modulated automobile LED lightings to estimate the vehicle relative positions. Methods in [69, 71, 43, 7, 82, 87, 63] all use photodiodes to detect LEDs. Since the photodiode could not sensor the accurate angle-of-arrival of the LED signal, these methods mainly focus on the measured light intensity, time difference of arrival (TDOA) and phase difference of arrival (PDOA). The method in [69] modulates the automobile LED lighting - either taillights or headlights - to send high rate repetitive on-off keying pattern, then the phase difference from two LED lamps are detected, while methods in [7, 63] try to detect the time difference of arrival from two LED lights. Once the TDOA or PDOA is detected, the range difference from the receiver to the two LED lights can

be computed and further the position of the receiver can be determined. Methods in [71, 41, 42, 82, 87] try to model the visible light channel and infer the range between receiver and LED lights based on the measured light intensity, and then calculate the receiver's position by trilateration. A probabilistic positioning utilizing VLC is proposed in [43]. The basic idea of the algorithm is to sample the signal strength for selected orientations at each reference point during the off-line phase and combine a subset of these values to histogram in the online phase, so that an orientation-specific signal strength distribution can be computed and utilized to increase the accuracy of position estimate. Positioning methods using image sensors are introduced in [29, 83, 66, 65], all of which measure the LED coordinates in the image frame and calculate the receiver's position based on geometry methods. Most of the existing positioning method use photo-detector alone, while method in [43] uses compass to improve the accuracy of training data, and method in [71] uses 6-axes to help estimate azimuth and tilt angle. All the methods introduced above use the LED ID to recognize each light source. Some of them like [83, 66, 65] even modulate each LED's own global position coordinates in the signal, so that the receiver does not have to previous known each LED's position.

All the visible light positioning methods mentioned above compute the receiver's position merely by the latest photo-detector's measurement, but the Markov property of the photo-detector's state is not considered. The inertial sensor and compass in [71, 43] are only used to help estimate the pose of receiver instead of predicting the receiver's state at the next time step. For the image sensor based positioning methods above, either the image sensor is stationary or no LED tracking algorithm is applied to extract the LED data. From the view of a vision aided navigation system [56], the positioning result can be greatly improved by incorporating motion and LED measurements together. One essential problem to combine vision aided navigation method with

VLC together is to track the LED in the image to extract its data. To solve this problem, the measurements in a sequence of images need to be considered jointly. The measurements of the same object (LED) at multiple time instances should be matched together to recover the data sequence.

As mentioned in Sec. 2.3, the existing corner detectors (Moravec [55], Forstner [24], Harris [30], Shi-Tomasi [72], and FAST [70]) or blob detectors (SIFT [52], SURF [12], and CENSURE [2]) can be used to match the projections of the same feature in different two-dimensional images. These detectors keep the description of each feature and these descriptions can be compared to determine whether multiple features are originated from the same object. However, the projection can be wrongly matched, especially when the sequence of images are long. Moreover, the existing feature descriptors cannot be applied to the linear array measurement, so that they cannot be used to match the measurements in different linear array scans.

For the global positioning system (GPS), the technology of receiver autonomous integrity monitoring (RAIM) [35] is developed to detect the fault measurements of the receiver. The RAIM technology performs consistency checks between all estimation solutions, and provides an alert to the system if the consistency checks fail [36]. Another outlier rejection method, named Least Soft-threshold Squares (LSS) [81], is newly developed. This method models the measurement error as the sum of Gaussian component and Laplacian component. The LSS method is efficient and easy to implement. Both of the two algorithms are designed to detect and reject the incorrect measurement, but cannot deal with multiple measurements.

Chapter 3

Navigation using Linear Photo Detector Arrays

3.1 Introduction

There are three kinds of optical sensors suitable for detecting the LED signal: single photo-detector (photodiode), two-dimensional photo-detector array (camera), and linear photo-detector array. As we have mentioned in Sec. 2.2, a single PD could communicate with LED's at rates exceeding 300MHz, but only offers the most basic level of positioning. The camera provides a much more precise measurement by detecting the LEDs positions in the image to form angle-of-arrival measurements, but could only work at low frame rate (30Hz). The low frame rate of camera will greatly restrict the information sent from the LEDs to camera.

A challenge in LED-based navigation system is how to improve the data transmission rate and the image processing efficiency while maintaining the capability for accurate navigation. To extract the ID, a sequence of images are grabbed and processed to find the on or off status of the LEDs. The frame rate f_s has to be at least two times

higher than the data rate f. Fig. 3.1 illustrates the data sequence of the LED sampled by the camera image, where the camera frame rate is two times of the LED data rate.

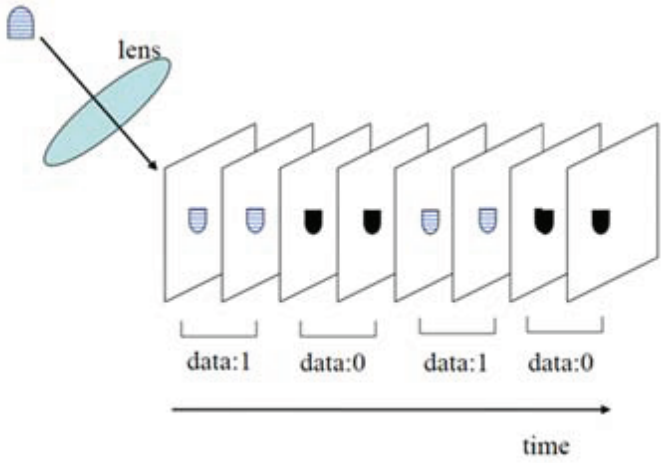


Figure 3.1: Camera continuously takes images of a scene with an LED light [31].

The large number of pixels (usually several mega-pixels) in the image sensor limits the frame rate. The frame rate of common commercial video camera is about only 30fps (frames per second), which indicates that the LED blinking rate has to be no higher than 15Hz which is too low. The frame rate of some special designed high-speed cameras is higher, for example the EoSens CL camera produced by Mikrotron GmbH could achieve 1800fps. The Eosens CL camera is shown in Fig. 3.2. Progress has been demonstrated in our experiments, where the LED signal is modulated at 800bps (bits per second) and the camera recovers the signal with a sampling rate of 1600fps [85, 86]. To further improve the frame rate, a solution is proposed in our former paper [85] that narrows the sampling area to a small portion of the whole image. To realize this proposition, the camera should have the capability to sample an arbitrary selected area in the image. The high-speed camera or the camera with small area sampling function usually has very high cost ($\geq$ 10k).

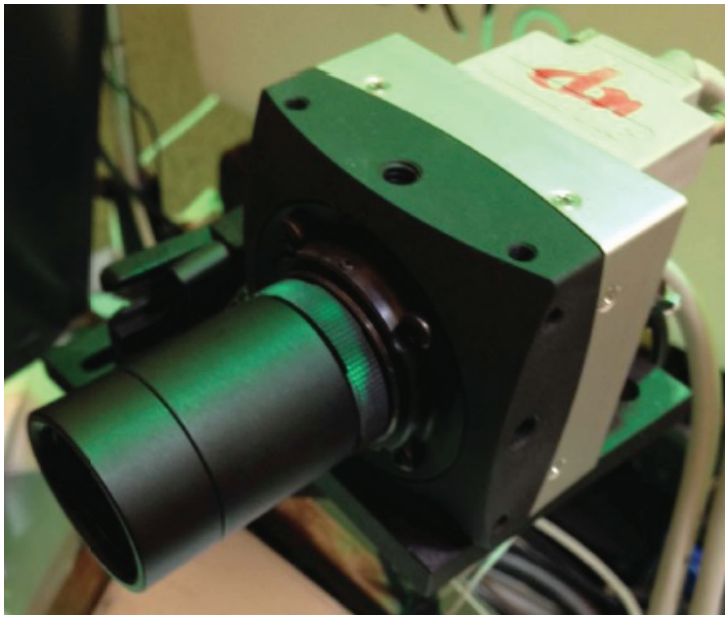


Figure 3.2: EoSens CL camera

This chapter investigates the ability of a linear array to preserve the cameras precise position measurement and simultaneously achieve high data rates. A linear array is a set of photodiodes that form one or several straight lines to receive the light signal. The number of pixels in a linear array is much smaller than that in a camera, since the linear array could be viewed as a small part of camera sensor. The linear array preserves the sensitivity to angle of arrival relative to a single axis. The advantages of linear array are:

- Fast sampling frequency: The low frame rate of camera is mainly due to its large number of pixels. In contrast, a linear array has only one or several rows of pixels.
- High accuracy: The measurement accuracy is mainly dependent on the number of pixels. Although the total pixel number of linear array is much smaller than that of camera, the pixel number in one direction could be much larger leading to a higher resolution in that direction.

- Low cost: The linear array has simple physical structure and small number of pixels.

On the other hand, having only one-dimensional sensitivity of the linear array also increases the difficulties of initialization, calibration and navigation since it cannot sense angular rotation around the axis of the array or linear motion in the direction to the plane defined by the sensitive axis and the LED array. A well-known result that the camera pose is observable [73] and can be computed in closed-form [3] when four or more known features are detected in each image will not apply to the linear array. In this chapter we analyze the extrinsic parameter calibration process and the navigation system state initialization for linear arrays.

Due to the one-dimensional sensitivity, navigation using the linear array is especially useful when the carrier is constrained to move in a 2D plane. For example, for cars equipped with linear arrays driving on roads, existing LED traffic lights are ideal navigation features. For vehicles moving in 3D space, by employing two linear arrays with their perpendicular sensitive directions, this problem can also be solved.

Currently, the linear array could be found in a variety of applications such as photocopiers, barcode scanners, and line scan cameras. Their light spectrum sensitivity ranges from short wave infrared to ultra violet, and the pixel number ranges from tens to thousands. The pixel read-out speed is up to 100MHz. Taking the Melexis's 3rd generation linear optical array MLX75306 as an example, it is 7.1mm long with 142 pixels, and has 12MHz read-out speed.

Unfortunately, none of the applications mentioned above is specifically designed for navigation and communication purposes. For this application we propose combining a linear array and cylindrical lens. Its physical structure is illustrated in Fig. 3.3.

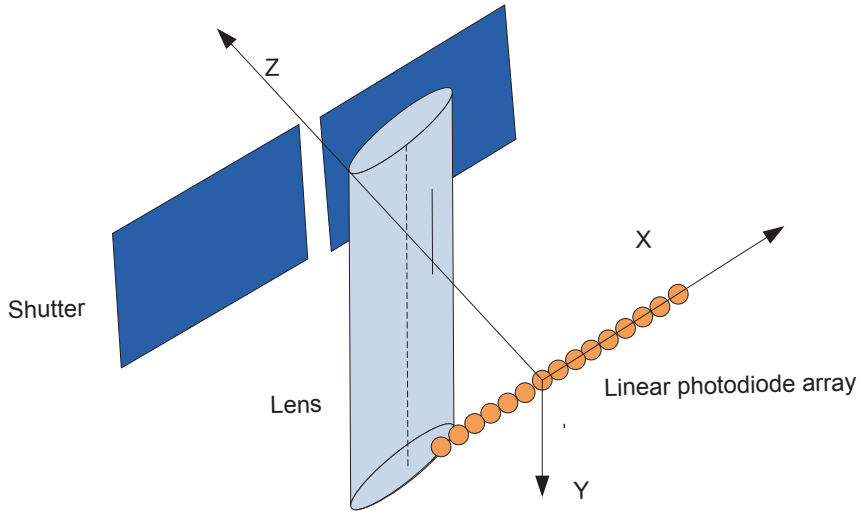


Figure 3.3: Physical structure of linear photodiode array sensor.

The basic physical structure includes a linear array, a convex-cylindrical lens and the shutter. The convex-cylindrical lens will focus the light that passes through onto a line instead of a point. The linear array will be put that perpendicular to the line which the lens focuses on. In the lateral plane, the elementary ray trace is similar with that of convex lens. The ray trace is illustrated in the right of Fig. 3.4. In the vertical plane, the cylindrical lens can be viewed as a flat glass plate which will not affect the ray trace. It is illustrated in the left of Fig. 3.4.

This chapter is organized as follows: Section 3.2 describes the problem details including the kinematic equation and measurement model of the system; Section 3.3 analyzes the nonlinear observability of this linear array based navigation system; Section 3.4 discusses the extrinsic calibration of the linear array and the initialization of the linear array based navigation system; Section 3.5 gives all the experimental results.

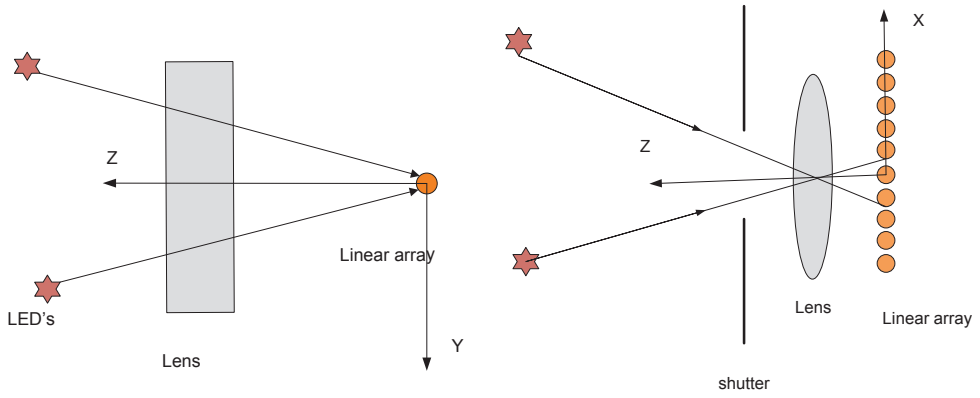


Figure 3.4: The side and top view of linear array.

3.2 Problem Description

In the following analysis, we firstly describe the general navigation system structure as well as kinematic equations and measurement models. After introducing the general model, we give descriptions of the specific navigation systems considered in this chapter.

3.2.1 General Navigation System

The structure of a general aided navigation system is illustrated in Fig. 3.5. The input of the system is the measurements of the high rate sensor and aiding sensors, and the output is the system state estimate at each time step. The high rate sensors include wheel encoders and IMU, which usually measure the velocity, acceleration and angular rate of the platform. The aiding sensors include GPS receiver, magnetometer, camera, thermometer and so on. In Fig. 3.5, the high rate part is colored by green, while the low rate part is red. The general kinematic model and aiding sensor model will be discussed in the following sections.

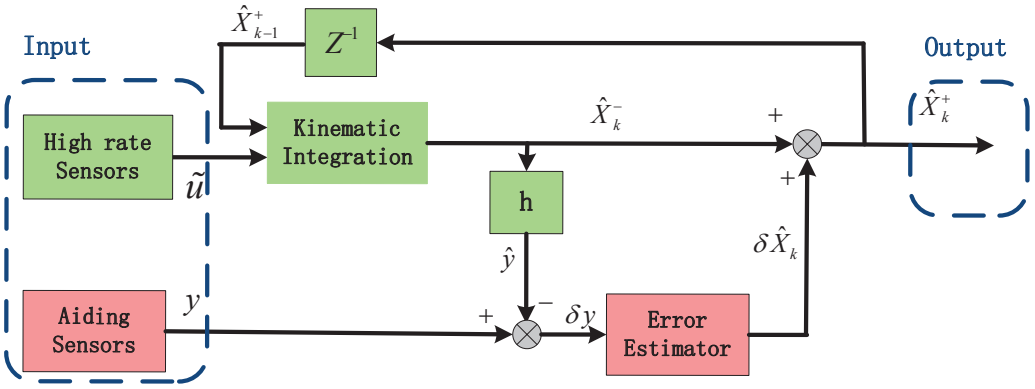


Figure 3.5: Aided navigation system

3.2.1.1 State Propagation

The aiding sensor is assumed to be part of a rigid system with kinematic state (i.e., position, velocity, attitude) denoted by $\boldsymbol{x}(t) \in \mathbb{R}^n$. We will refer to this rigid system as a *rover*. The rover trajectory evolves over time according to

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left(\boldsymbol{x}(t), \boldsymbol{u}(t)\right), \tag{3.1}$$

where $\boldsymbol{f} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is a known nonlinear mapping and $\boldsymbol{u}(t) \in \mathbb{R}^m$ is the system input. In a navigation system, the input $\boldsymbol{u}$ represents the rover's motion information, which could be measured by an encoder or inertial measurement unit (IMU).

Throughout this article, we use symbol $\hat{a}$ to represent the estimate of variable a. Given an initial condition $\boldsymbol{x}(0) \sim \mathcal{N}(\boldsymbol{x}_0, \boldsymbol{P}_0)$, the computer calculates an estimate $\hat{\boldsymbol{x}}$ of $\boldsymbol{x}$ according to

$$\dot{\hat{\boldsymbol{x}}}(t) = \boldsymbol{f}\left(\hat{\boldsymbol{x}}(t), \hat{\boldsymbol{u}}(t)\right), \tag{3.2}$$

where $\hat{\boldsymbol{u}}$ is the estimate of $\boldsymbol{u}$ computed from its measurement $\tilde{\boldsymbol{u}}$. In the simplest case, the measurement is modeled as $\tilde{\boldsymbol{u}} = \boldsymbol{u} + \boldsymbol{\omega}$, where the process noise $\boldsymbol{\omega}$ has power spectrum density (PSD) denoted by $\boldsymbol{Q}$.

Define $\delta x = x - \hat{x}$ as the state error, then its propagation model can be approximated by subtracting eqn. (3.2) from (3.1) and linearizing the result along the estimated trajectory:

$$\delta\dot{\boldsymbol{x}}(t) = \boldsymbol{F}(t)\delta\boldsymbol{x}(t) + \boldsymbol{G}(t)\boldsymbol{\omega}(t), \tag{3.3}$$

where

$$\boldsymbol{F}(t) = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}|_{\boldsymbol{x}=\hat{\boldsymbol{x}},\boldsymbol{u}=\hat{\boldsymbol{u}}} \tag{3.4}$$

$$\boldsymbol{G}(t) = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}|_{\boldsymbol{x}=\hat{\boldsymbol{x}},\boldsymbol{u}=\hat{\boldsymbol{u}}}. \tag{3.5}$$

If we are only concerned with the state estimates and their errors at the times t_k at which the aiding measurements happen, and use the subscript k as the shorthand notation for t_k, we can use the following error propagation model:

$$\delta\boldsymbol{x}(k) = \boldsymbol{\Phi}_{k-1}\delta\boldsymbol{x}(k-1) + \boldsymbol{\omega}_{k-1}, \tag{3.6}$$

where $\boldsymbol{x}(k)$ is the short for $\boldsymbol{x}(t_k)$, and $\boldsymbol{\omega}_{k-1} \sim \mathcal{N}(0, \boldsymbol{Q}_{k-1})$. Computation of the state error transition matrix $\boldsymbol{\Phi}_{k-1}$ and process noise covariance $\boldsymbol{Q}_{k-1}$ are discussed in [22]. From (3.6), the state error covariance $\boldsymbol{P}_k$ evolves over time according to

$$\boldsymbol{P}_k = \boldsymbol{\Phi}_{k-1}\boldsymbol{P}_{k-1}\boldsymbol{\Phi}_{k-1}^{\top} + \boldsymbol{Q}_{k-1}. \tag{3.7}$$

According to eqn. (3.7), the state error can be accumulated quickly because of the process noise $\boldsymbol{\omega}_k$ Usually after only a few second, the estimation error could become unacceptable, which is illustrated in Fig. 3.6. That is why we need aiding sensors to correct the state estimation.

3.2.1.2 Aiding Sensor Model

The general aiding sensor measurement model is

$$\boldsymbol{z}(k) = \boldsymbol{h}\left(\boldsymbol{x}(k)\right) + \boldsymbol{n}_k, \tag{3.8}$$

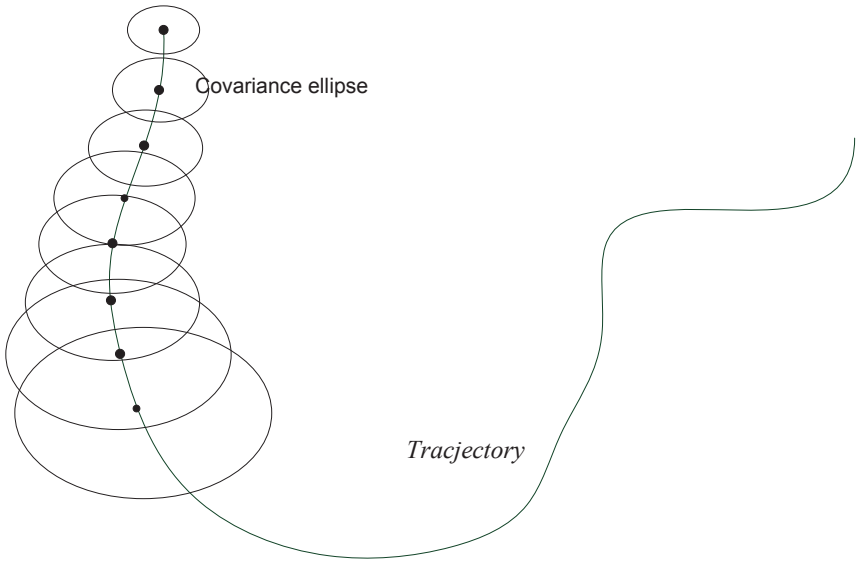


Figure 3.6: Growing of the state error covariance due to the process noise.

where $\boldsymbol{z}(k) \in \mathbb{R}^p$, and $\boldsymbol{h} : \mathbb{R}^n \to \mathbb{R}^p$ is the measurement function, and $\boldsymbol{n}_k \sim \mathcal{N}(0, \boldsymbol{R}_k)$ is the measurement noise. Given the state estimate at time step k, it is straightforward to compute both the predicted measurement $\hat{\boldsymbol{z}}(k)$ according to

$$\hat{\boldsymbol{z}}(k) = \boldsymbol{h}(\hat{\boldsymbol{x}}(k)). \tag{3.9}$$

The prediction error is defined as $\delta\boldsymbol{z} = \boldsymbol{z} - \hat{\boldsymbol{z}}$, and its model can be approximated by subtracting eqn. (3.9) from (3.8):

$$\delta\boldsymbol{z}(k) = \boldsymbol{H}_k \delta\boldsymbol{x}(k) + \boldsymbol{R}_k, \tag{3.10}$$

where $\boldsymbol{H}_k \in \mathbb{R}^{p \times n}$ is the linearized measurement matrix obtained from

$$\boldsymbol{H}_k = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}|_{\boldsymbol{x}=\hat{\boldsymbol{x}}(k)}. \tag{3.11}$$

From eqn. (3.10), its error covariance $\boldsymbol{S}(k)$ is computed as

$$\boldsymbol{S}(k) \quad = \quad \boldsymbol{H}_k \boldsymbol{P}_k \boldsymbol{H}_k^\top + \boldsymbol{R}_k. \tag{3.12}$$

The quantities $\hat{\boldsymbol{z}}(k)$ and $\boldsymbol{S}(k)$ define a prior distribution for the feature measurement.

3.2.1.3 Sensor Fusion

As showed in Fig. 3.6, the state estimate error will accumulate if the state estimate is calculated only by integrating the measurements from the motion sensor. The extended Kalman filter (EKF) is frequently used to fuse the measurements from motion sensor and aiding sensors. The state estimate propagates based on eqn. (3.2) where $\hat{\boldsymbol{u}}(t)$ is viewed as a constant between each time interval $[t_{k-1}, t_k]$. Integrating eqn. (3.2), the state estimate propagates according to

$$\hat{\boldsymbol{x}}(k) = \hat{\boldsymbol{x}}(k-1) + \int_{t_{k-1}}^{t_k} \boldsymbol{f}\left(\hat{\boldsymbol{x}}(t), \hat{\boldsymbol{u}}(t)\right) dt \tag{3.13}$$

The state error propagates according to eqn. (3.7).

The state computed from eqn. (3.13) and the error covariance propagated from eqn. (3.7) are the prior state estimate covariance denoted by $\hat{\boldsymbol{x}}^-(k)$ and $\boldsymbol{P}_k^-$, respectively. When a measurement arrives at time t_k, the state and its error covariance are updated according to

$$\begin{aligned} \boldsymbol{K}_k &= \boldsymbol{P}_k^- \boldsymbol{H}_k^\top (\boldsymbol{H}_k \boldsymbol{P}_k^- \boldsymbol{H}_k^\top + \boldsymbol{R}_k)^{-1} \\ \hat{\boldsymbol{x}}^+(k) &= \hat{\boldsymbol{x}}^-(k) + \boldsymbol{K}_k(\boldsymbol{z}(k) - \boldsymbol{h}(\hat{\boldsymbol{x}}_k)) \\ \boldsymbol{P}_k^+ &= \boldsymbol{P}_k^- - \boldsymbol{K}_k \boldsymbol{H}_k \boldsymbol{P}_k^-, \end{aligned} \tag{3.14}$$

where $\boldsymbol{K}_k$ is the EKF gain evaluated at time t_k, and $\hat{\boldsymbol{x}}^+$ and $\boldsymbol{P}^+$ are the posterior state estimate and error covariance. Fig. 3.7 illustrates the how the state error ellipse evolves over time when the aiding sensor measurements are added in the estimation process. The state error decreases when the aiding measurement happens, so that state error keeps in a tolerable range when the rover moves.

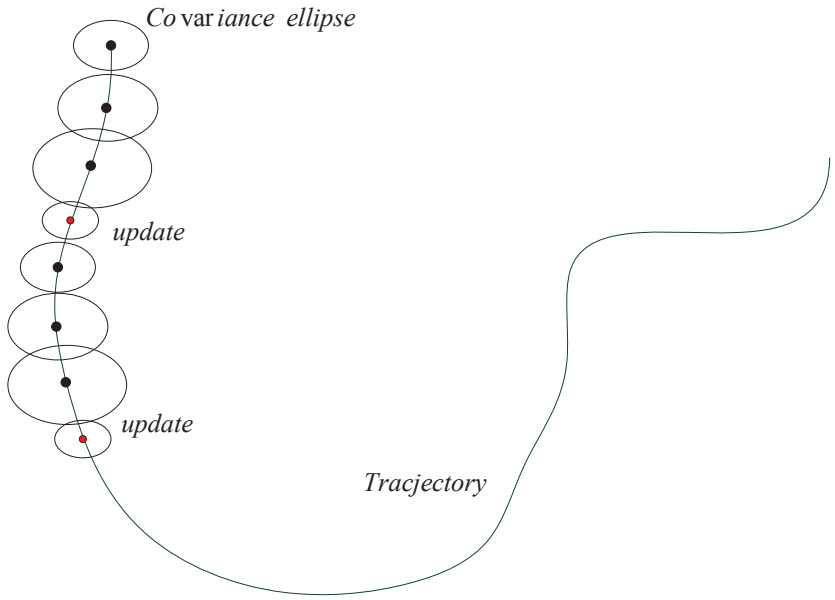


Figure 3.7: The state error ellipse evolves over time when the aiding sensor measurements are added.

3.2.2 Kinematic Equations

In this chapter, we consider a vehicle moving in a 2D plane where the vehicle is equipped with a linear array. The vehicle also has linear and rotational velocity inputs (μ and ω) which are measured via additional sensors like wheel encoders. The goal is to estimate the vehicle position coordinates n (north), e (east) and heading ψ in the world frame. The state vector is $\boldsymbol{x} = [n, e, \psi]^\top$, and input is $\boldsymbol{u} = [\mu, \omega]^\top$. The kinematic equations are

$$\begin{bmatrix} \dot{n} \\ \dot{e} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos\psi \\ \sin\psi \\ 0 \end{bmatrix} \mu + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega. \tag{3.15}$$

The state estimate vector is written as $\hat{\boldsymbol{x}} = \left[\hat{n}, \hat{e}, \hat{\psi}\right]^\top$, and it propagates discretely according to eqn. (3.13). The state error vector is defined as $\delta\boldsymbol{x} = [\delta n, \delta e, \delta\psi]^\top$ and

$$\delta\boldsymbol{x} = \boldsymbol{x} - \hat{\boldsymbol{x}} = \begin{bmatrix} n - \hat{n} \\ e - \hat{e} \\ \psi - \hat{\psi} \end{bmatrix}. \tag{3.16}$$

The state error propagate equation is defined in (3.3), where the linearized matrix $\boldsymbol{F}$ and $\boldsymbol{G}$ are

$$\boldsymbol{F} = \begin{bmatrix} 0 & 0 & -\sin\psi \\ 0 & 0 & \cos\psi \\ 0 & 0 & 0 \end{bmatrix}, \boldsymbol{G} = \begin{bmatrix} \cos\psi & 0 \\ \sin\psi & 0 \\ 0 & 1 \end{bmatrix}. \tag{3.17}$$

The state error covariance is propagated according to eqn. (3.7).

3.2.3 Measurement Equation

The linear array measurement, viewed as the lateral portion of a camera measurement. So firstly we will describe the pinhole camera measurement model. Let

$${}^{c}\boldsymbol{p}_{cL} = \begin{bmatrix} {}^{c}X & {}^{c}Y & {}^{c}Z \end{bmatrix}^\top \tag{3.18}$$

denote the coordinates of an LED in camera frame. This vector is computed as

$${}^{c}\boldsymbol{p}_{cL} = \begin{bmatrix} {}^{c}_{w}\boldsymbol{R} & {}^{c}\boldsymbol{p}_{cw} \end{bmatrix} \begin{bmatrix} {}^{w}\boldsymbol{p}_{wL} \\ 1 \end{bmatrix} \tag{3.19}$$

$${}^{c}_{w}\boldsymbol{R} = {}^{c}_{b}\boldsymbol{R}\,{}^{b}_{w}\boldsymbol{R}$$

$${}^{c}\boldsymbol{p}_{cw} = {}^{c}_{b}\boldsymbol{R}\left({}^{b}\boldsymbol{p}_{cb} - {}^{b}_{w}\boldsymbol{R}\,{}^{w}\boldsymbol{p}_{wb}\right), \tag{3.20}$$

where ${}^{b}\boldsymbol{p}_{cb}$ and ${}^{c}_{b}\boldsymbol{R}$ are the camera extrinsic calibration parameters, and ${}^{w}\boldsymbol{p}_{wL}$ is the known LED location in navigation frame. Throughout this dissertation, the symbol ${}^{b}_{a}\boldsymbol{R}$

denotes the rotation matrix transforming a vector from frame a to frame b. The symbol ${}^{c}\boldsymbol{p}_{ab}$ denotes the vector pointing from location a to location b represented in frame c.

The measurement $y_d = \begin{bmatrix} u_d & v_d \end{bmatrix}^\top$ accounting for camera distortion is modeled as follows:

$$u = \frac{{}^{c}X}{{}^{c}Z} \tag{3.21}$$

$$v = \frac{{}^{c}Y}{{}^{c}Z} \tag{3.22}$$

$$r^2 = u^2 + v^2 \tag{3.23}$$

$$\begin{aligned} {}^{c}X' &= u(1 + k_1 r^2 + k_2 r^4) \\ &\quad + 2p_1 uv + p_2(r^2 + 2u^2) \end{aligned} \tag{3.24}$$

$$\begin{aligned} {}^{c}Y' &= v(1 + k_1 r^2 + k_2 r^4) \\ &\quad + 2p_2 uv + p_1(r^2 + 2v^2) \end{aligned} \tag{3.25}$$

$$u_d = f_x \cdot {}^{c}X' + c_x \tag{3.26}$$

$$v_d = f_y \cdot {}^{c}Y' + c_y \tag{3.27}$$

where (f_x, f_y, c_x, c_y) are camera intrinsic parameters and (k_1, k_2, p_1, p_2) are the distortion parameters. The intrinsic parameters and distortions are determined offline using standard calibration methods in [14]. For convenience, we usually use the undistorted camera measurements defined in eqn. (3.22). The undistorted camera measurements can be computed from the distorted measurements if the camera intrinstic parameters are known. Several approaches are introduced in linteratures [33].

From the introductions above, the linear array measurement without considering the distortions is modeled as

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}) = \frac{{}^{l}x}{{}^{l}z}, \tag{3.28}$$

where ${}^{l}\boldsymbol{p}_{lf} = \begin{bmatrix} {}^{l}x & {}^{l}y & {}^{l}z \end{bmatrix}^\top$ is the feature position in the linear array frame which is

similar to the definition of ${}^{l}\boldsymbol{p}_{lf}$. It is modeled as

$$
\begin{aligned}
{}^{l}\boldsymbol{p}_{lf} &= {}^{l}_{b}\boldsymbol{R}\left({}^{b}\boldsymbol{p}_{bf} - {}^{b}\boldsymbol{p}_{bl}\right), && (3.29)\\
{}^{b}\boldsymbol{p}_{bf} &= {}^{b}_{w}\boldsymbol{R}\left({}^{w}\boldsymbol{p}_{wf} - {}^{w}\boldsymbol{p}_{wb}\right), && (3.30)
\end{aligned}
$$

where the letter w and b in the above equations represent the world frame and body frame, respectively.

From the eqns. (3.18–3.22), the linearized measurement matrix $\boldsymbol{H}_c$ for the camera measurement model is

$$
\boldsymbol{H}_c = \frac{\partial \boldsymbol{h}}{\partial {}^{c}\boldsymbol{p}_{cL}} \frac{\partial {}^{c}\boldsymbol{p}_{cL}}{\partial \boldsymbol{x}} = \boldsymbol{\mathcal{H}}_1 \boldsymbol{\mathcal{H}}_2, \tag{3.31}
$$

where

$$
\boldsymbol{\mathcal{H}}_1 = \begin{bmatrix} \frac{1}{{}^{c}Z} & 0 & -\frac{{}^{c}X}{({}^{c}Z)^2} \\ 0 & \frac{1}{{}^{c}Z} & -\frac{{}^{c}Y}{({}^{c}Z)^2} \end{bmatrix} \tag{3.32}
$$

has two rows as the vectors in the two directions orthogonal to ${}^{c}\boldsymbol{p}_{cL}$. From equations (3.19) and (3.20), we have

$$
\boldsymbol{\mathcal{H}}_2 = \begin{bmatrix} -{}^{c}_{n}\boldsymbol{R}(:, 1:2) & {}^{c}_{b}\boldsymbol{R}\boldsymbol{J}_{\psi}{}^{b}\boldsymbol{p}_{bL} & 0_{3\times 2} \end{bmatrix}, \tag{3.33}
$$

where

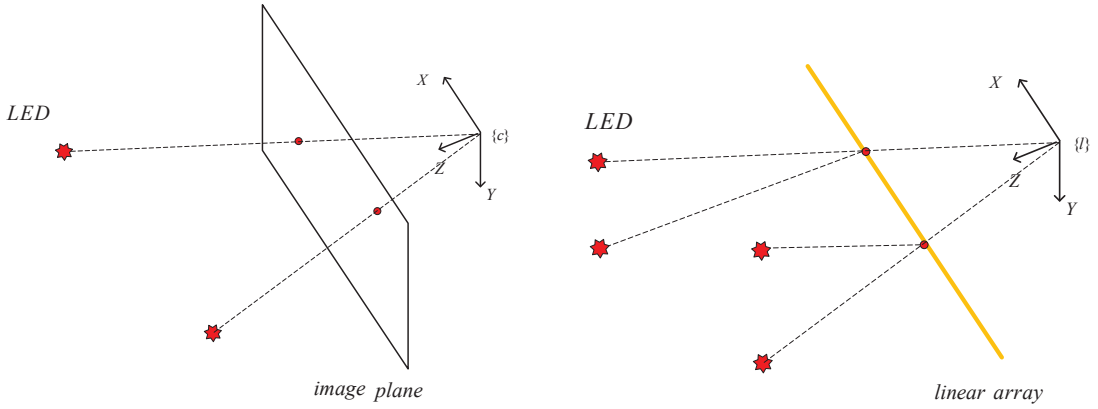
$$
\boldsymbol{J}_{\psi} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.34}
$$

and ${}^{c}_{n}\boldsymbol{R}(:, 1:2)$ is the first two columns of matrix ${}^{c}_{n}\boldsymbol{R}$.

The linearized measurement matrix $\boldsymbol{H}_l$ for the linear array measurement model corresponds the first row of matrix $\boldsymbol{H}_c$, and it can also be decomposed as

$$
\boldsymbol{H}_l = \boldsymbol{\mathcal{H}}_{1,l} \boldsymbol{\mathcal{H}}_2, \tag{3.35}
$$

where $\boldsymbol{\mathcal{H}}_{1,l}$ corresponds the first row of $\boldsymbol{\mathcal{H}}_1$. The measurement model of the camera and linear array are illustrated in Fig. 3.8.



(a) Camera model.

(b) Linear array model.

Figure 3.8: Camera and linear array measurement model

3.3 Observability Analysis

In this section, we consider the observability of the vehicle state using the measurements $\boldsymbol{z}$ of the linear array for the system of Eqn. (3.15) with inputs μ and ω. We prove that the state $\boldsymbol{x}$ is fully observable when at least two horizontally separated LED's are measured simultaneously.

3.3.1 Nonlinear Observability

Consider the nonlinear system

$$\dot{\boldsymbol{x}} = \boldsymbol{f}_0(x) + \sum_{i=1}^{m} \boldsymbol{f}_i(x)\, u_i, \tag{3.36}$$

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}), \tag{3.37}$$

where $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{z} \in \mathbb{R}^p$, $\boldsymbol{f}_i : \mathbb{R}^n \to \mathbb{R}^n$, and $u_1, \ldots, u_m$ are m scalar inputs. The nonlinear system described by (3.36)-(3.37) is locally weakly observable over a region D if its observability matrix has full rank on D [34]. The observability matrix is defined as a matrix with rows:

$$\boldsymbol{\mathcal{O}} \triangleq \{\boldsymbol{\nabla}\boldsymbol{\mathcal{L}}^{\ell}_{\boldsymbol{f}_{i_1}\cdots\boldsymbol{f}_{i_\ell}}\boldsymbol{h}_k(\boldsymbol{x}) \mid k = 1, \ldots, p; \ell \in \mathbb{N}\}, \tag{3.38}$$

where $\boldsymbol{\mathcal{L}}^{\ell}_{\boldsymbol{f}_{i_1}\cdots\boldsymbol{f}_{i_\ell}}$ is the Lie-derivative of order ℓ with respect to $\boldsymbol{f}_{i_1}\cdots\boldsymbol{f}_{i_\ell}$, and $\boldsymbol{h}_k(\boldsymbol{x})$ is the kth element of $\boldsymbol{h}(\boldsymbol{x})$. For the system (3.15), we have $m = 2$, $p = 1$, $\boldsymbol{f}_0 = 0$ $\boldsymbol{f}_1 = \begin{bmatrix} \cos\psi & \sin\psi & 0 \end{bmatrix}^\top$, and $\boldsymbol{f}_2 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\top$.

3.3.2 Observability Analysis

Since the vehicle moves in the 2D plane, the height component of the vehicle position in the world frame is fixed which is 0 in our frame definition. Using the following standard notations

$$\begin{aligned}
{}^{b}_{w}\boldsymbol{R} &= \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, && (3.39)\\
{}^{w}\boldsymbol{p}_{wf} &= \begin{bmatrix} x_f & y_f & z_f \end{bmatrix}^\top, && (3.40)\\
{}^{w}\boldsymbol{p}_{wb} &= \begin{bmatrix} n & e & 0 \end{bmatrix}^\top, && (3.41)
\end{aligned}$$

to reorganize matrix $\boldsymbol{\mathcal{H}}_2$ in (3.33), we have

$$\boldsymbol{\mathcal{H}}_2 = {}^{l}_{b}\boldsymbol{R} \begin{bmatrix} -\cos\psi & -\sin\psi & a_1 \\ \sin\psi & -\cos\psi & a_2 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.42}$$

where

$$\begin{aligned} a_1 &= -(x_f - n)\sin\psi + (y_f - e)\cos\psi, \\ a_2 &= -(x_f - n)\cos\psi - (y_f - e)\sin\psi. \end{aligned}$$

From Eqn. (3.42)-(3.43), if we use $\boldsymbol{h}_i$ to represent the ith column of $\boldsymbol{H}_l$, we have

$$\boldsymbol{h}_3 = -(y_f - e)\,\boldsymbol{h}_1 + (x_f - n)\,\boldsymbol{h}_2. \tag{3.43}$$

For the system (3.15), if three rows (columns) in the observability matrix $\boldsymbol{\mathcal{O}}$ are found to be linearly independent, the state $\boldsymbol{x}$ will be fully observable. Obviously, the first row of $\boldsymbol{\mathcal{O}}$ is the linearized measurement matrix $\boldsymbol{H}$. Before discussing the rank of $\boldsymbol{\mathcal{O}}$, we firstly state the following lemma:

Lemma 1 *For the system defined in (3.15), let $\boldsymbol{G} : \mathbb{R}^3 \to \mathbb{R}^p$ be a function defined on the state space. Suppose equation*

$$\frac{\partial \boldsymbol{G}}{\partial \psi} = -(y_f - e)\frac{\partial \boldsymbol{G}}{\partial n} + (x_f - n)\frac{\partial \boldsymbol{G}}{\partial e} \tag{3.44}$$

holds for G, then (3.44) still holds when we replace $\boldsymbol{G}$ by $\boldsymbol{\mathcal{L}}_{\boldsymbol{f}_i}\boldsymbol{G}, i = 1, 2$.

Proof. Firstly, let $\boldsymbol{f}_i = \boldsymbol{f}_1$, then

$$\boldsymbol{\mathcal{L}}_{\boldsymbol{f}_1}\boldsymbol{G} = \frac{\partial \boldsymbol{G}}{\partial n}\cos\psi + \frac{\partial \boldsymbol{G}}{\partial e}\sin\psi. \tag{3.45}$$

Then

$$\frac{\partial \boldsymbol{\mathcal{L}}_{\boldsymbol{f}_1}\boldsymbol{G}}{\partial n} = \frac{\partial^2 \boldsymbol{G}}{\partial n^2}\cos\psi + \frac{\partial^2 \boldsymbol{G}}{\partial n \partial e}\sin\psi \tag{3.46}$$

$$\frac{\partial \boldsymbol{\mathcal{L}}_{\boldsymbol{f}_1}\boldsymbol{G}}{\partial e} = \frac{\partial^2 \boldsymbol{G}}{\partial e \partial n}\cos\psi + \frac{\partial^2 \boldsymbol{G}}{\partial^2 e}\sin\psi \tag{3.47}$$

$$\begin{aligned} \frac{\partial \boldsymbol{\mathcal{L}}_{\boldsymbol{f}_1}\boldsymbol{G}}{\partial \psi} &= \frac{\partial^2 \boldsymbol{G}}{\partial \psi \partial n}\cos\psi + \frac{\partial^2 \boldsymbol{G}}{\partial \psi \partial e}\sin\psi \\ &\quad -\frac{\partial \boldsymbol{G}}{\partial n}\sin\psi + \frac{\partial \boldsymbol{G}}{\partial e}\cos\psi. \end{aligned} \tag{3.48}$$

Take the partial derivative with respect to n for the both left and right hand sides of eqn. (3.44), we have

$$\begin{aligned}\frac{\partial^2 \boldsymbol{G}}{\partial\psi\partial n} &= \frac{\partial^2 \boldsymbol{G}}{\partial n\partial\psi} = \frac{\partial}{\partial n}\left(\frac{\partial \boldsymbol{G}}{\partial\psi}\right) \\ &= -(y_L - e)\frac{\partial^2 \boldsymbol{G}}{\partial n^2} + (x_L - n)\frac{\partial^2 \boldsymbol{G}}{\partial n\partial e} - \frac{\partial \boldsymbol{G}}{\partial e}. \end{aligned} \tag{3.49}$$

Substitute eqn. (3.49) into the right hand side of eqn. (3.48) and reorganize it, we have

$$\begin{aligned}\frac{\partial \boldsymbol{\mathcal{L}}_{\boldsymbol{f}_1}\boldsymbol{G}}{\partial\psi} &= -(y_L - e)\left(\frac{\partial^2 \boldsymbol{G}}{\partial n^2}\cos\psi + \frac{\partial^2 \boldsymbol{G}}{\partial e\partial n}\sin\psi\right) \\ &\quad + (x_L - n)\left(\frac{\partial^2 \boldsymbol{G}}{\partial n\partial e}\cos\psi + \frac{\partial^2 \boldsymbol{G}}{\partial e^2}\sin\psi\right) \\ &= -(y_L - e)\frac{\partial \boldsymbol{\mathcal{L}}_{\boldsymbol{f}_1}\boldsymbol{G}}{\partial n} + (x_L - n)\frac{\partial \boldsymbol{\mathcal{L}}_{\boldsymbol{f}_1}\boldsymbol{G}}{\partial e}. \end{aligned} \tag{3.50}$$

Secondly, following the similar analysis above, let $\boldsymbol{f}_i = \boldsymbol{f}_2$.

$$\boldsymbol{\mathcal{L}}_{\boldsymbol{f}_2}\boldsymbol{G} = \frac{\partial \boldsymbol{G}}{\partial\psi}. \tag{3.51}$$

Then

$$\frac{\partial \boldsymbol{\mathcal{L}}_{\boldsymbol{f}_2}\boldsymbol{G}}{\partial n} = \frac{\partial^2 \boldsymbol{G}}{\partial\psi\partial n} \tag{3.52}$$

$$\frac{\partial \boldsymbol{\mathcal{L}}_{\boldsymbol{f}_2}\boldsymbol{G}}{\partial e} = \frac{\partial^2 \boldsymbol{G}}{\partial\psi\partial e} \tag{3.53}$$

$$\frac{\partial \boldsymbol{\mathcal{L}}_{\boldsymbol{f}_2}\boldsymbol{G}}{\partial\psi} = \frac{\partial^2 \boldsymbol{G}}{\partial\psi^2}. \tag{3.54}$$

Take the partial derivative with respect to ψ for the both left and right hand sides of eqn. (3.44), we have

$$\begin{aligned}\frac{\partial^2 \boldsymbol{G}}{\partial\psi^2} &= \frac{\partial}{\partial\psi}\left(-(y_f - e)\frac{\partial \boldsymbol{G}}{\partial n} + (x_f - n)\frac{\partial \boldsymbol{G}}{\partial e}\right) \\ &= -(y_L - e)\frac{\partial^2 \boldsymbol{G}}{\partial n\partial\psi} + (x_L - n)\frac{\partial^2 \boldsymbol{G}}{\partial e\partial\psi}. \end{aligned} \tag{3.55}$$

Substitute eqn. (3.55) into the right hand side of eqn. (3.54), we have

$$\begin{aligned}\frac{\partial \boldsymbol{\mathcal{L}}_{\boldsymbol{f}_2}\boldsymbol{G}}{\partial\psi} &= -(y_L - e)\frac{\partial^2 \boldsymbol{G}}{\partial n\partial\psi} + (x_L - n)\frac{\partial^2 \boldsymbol{G}}{\partial e\partial\psi} \\ &= -(y_L - e)\frac{\partial \boldsymbol{\mathcal{L}}_{\boldsymbol{f}_2}\boldsymbol{G}}{\partial n} + (x_L - n)\frac{\partial \boldsymbol{\mathcal{L}}_{\boldsymbol{f}_2}\boldsymbol{G}}{\partial e}. \end{aligned} \tag{3.56}$$

Then the proof is finished. ■

From the lemma, when property (3.44) holds for a function, it also holds for its Lie-derivative, and of course its Lie-derivative of any order.

Theorem 2 *The observability matrix defined in (3.38), for a single LED, does not have full rank for the system in (3.15).*

Proof. Eqn. (3.43) indicates that the measurement function $\boldsymbol{h}(x)$ has property (3.44). Then from Lemma 1, $\boldsymbol{\mathcal{L}}^{\ell}_{\boldsymbol{f}_{i_1}\dots\boldsymbol{f}_{i_\ell}}\boldsymbol{h}\,(\boldsymbol{f}_{i_1}\dots\boldsymbol{f}_{i_\ell}=1,2.)$ satisfies (3.44) for any $\ell\in\mathbb{N}$, so that each column in the observability matrix satisfies eqn. (3.44). This means the third column of the observability matrix in (3.38) is always the linear combination of its first two columns, so it does not have full rank. ■

Next, we are going to prove that the observability matrix has exactly rank 2 when only one LED is measured. Since Theorem 2 says $rank(\boldsymbol{\mathcal{O}})\leq 2$, we only need to show $rank(\boldsymbol{\mathcal{O}})\geq 2$ in the following analysis.

Considering the row $\boldsymbol{\nabla\mathcal{L}}_{\boldsymbol{f}_1}\boldsymbol{h}$ in $\boldsymbol{\mathcal{O}}$, we have

$$\boldsymbol{\mathcal{L}}_{\boldsymbol{f}_1}\boldsymbol{h} = \boldsymbol{H}_l\boldsymbol{f}_1 = -\frac{r_{11}}{l_z}+\frac{r_{31}{}^l x}{l_z{}^2} \tag{3.57}$$

$$\begin{aligned}\boldsymbol{\nabla\mathcal{L}}_{\boldsymbol{f}_1}\boldsymbol{h} &= \begin{bmatrix} \frac{r_{31}}{l_z{}^2} & 0 & \frac{r_{11}}{l_z{}^2}-\frac{2r_{31}{}^l x}{l_z{}^3}\end{bmatrix}\boldsymbol{\mathcal{H}}_2\\ &= \frac{r_{31}}{l_z}\boldsymbol{H}+\begin{bmatrix} 0 & 0 & \frac{r_{11}}{l_z{}^2}-\frac{r_{31}{}^l x}{l_z{}^3}\end{bmatrix}\boldsymbol{\mathcal{H}}_2,\end{aligned} \tag{3.58}$$

where r_{ij} is the element of matrix ${}^l_b\boldsymbol{R}$ at ith row and jth column. Then $rank(\boldsymbol{\mathcal{O}})\geq 2$ is true when $\boldsymbol{\nabla\mathcal{L}}_{\boldsymbol{f}_1}\boldsymbol{h}$ is linearly independent of $\boldsymbol{H}$, since $\boldsymbol{\nabla\mathcal{L}}_{\boldsymbol{f}_1}\boldsymbol{h}$ and $\boldsymbol{\mathcal{O}}$ are two different rows in the observability matrix. Eqn. (3.58) shows that $\boldsymbol{\nabla\mathcal{L}}_{\boldsymbol{f}_1}\boldsymbol{h}$ is the linear combination of $\boldsymbol{H}$ and the third row of $\boldsymbol{\mathcal{H}}_2$, which indicates that $\boldsymbol{\nabla\mathcal{L}}_{\boldsymbol{f}_1}\boldsymbol{h}$ is linearly independent of $\boldsymbol{H}$ if and only if the third row of $\boldsymbol{\mathcal{H}}_2$ is linearly independent of $\boldsymbol{H}$. From Eqn. (3.31), (3.32) and (3.35), $\boldsymbol{H}$ itself is the linear combination of the first and third

rows of $\boldsymbol{\mathcal{H}}_2$, then $\boldsymbol{H}$ is linearly independent on the third row of $\boldsymbol{\mathcal{H}}_2$ if and only if the first and third rows of $\boldsymbol{\mathcal{H}}_2$ are linearly independent. From Eqn. (3.42), the independency of the first and third rows in $\boldsymbol{\mathcal{H}}_2$ requires the first and third rows of ${}^l_b\boldsymbol{R}$ are independent. Moreover, since the third row of the matrix in the right hand side of (3.42) is 0, then the independency of the first and third rows in $\boldsymbol{\mathcal{H}}_2$ only requires that the following matrix

$$\begin{bmatrix} r_{11} & r_{12} \\ r_{31} & r_{32} \end{bmatrix} \tag{3.59}$$

is nonsingular. This matrix is dependent on the pose of the linear array on the carrier.

If more than one LED's are measured, and not all of the LED's have the same x_f and y_f coordinates, then the observability matrix $\boldsymbol{\mathcal{O}}$ will have full rank since Eqn. (3.43) is no longer true. To estimate state $\boldsymbol{x}$, at least two horizontally separated LED's should be observed simultaneously.

3.4 Calibration and Initialization

The calibration of the extrinsic parameters of linear array is a little more difficult than that of the standard camera because of its one-dimensional sensitivity. The well-known result that the camera pose is observable [73] and can be computed in closed-form [3] when four or more known features are detected in each image will not apply to the linear array. In this section, we are going to estimate the extrinsic parameters using the nonlinear optimization method, initialized with approximate prior knowledge of these parameters. The initialization method follows the method in [86].

3.4.1 Extrinsic Parameter Calibration

Being different from the extrinsic calibration of camera, the extrinsic parameters of linear array cannot be estimated only by its measurement without prior knowl-

edge. Taking the computation of ${}^{b}\boldsymbol{p}_{bl}$ for example, even though when all the other parameters except ${}^{b}\boldsymbol{p}_{bl}$ in Eqn. (3.28)-(3.29) are perfectly known, the vector ${}^{b}\boldsymbol{p}_{bl}$ still could not be obtained. Assume there are m LEDs are measured with measurement set $\{z_1, z_2, \ldots, z_m\}$ and their corresponding coordinates in the body frame are

$$ {}^{b}\boldsymbol{p}_{bf_i} = \begin{bmatrix} f_{i,1}, f_{i,2}, f_{i,3} \end{bmatrix}^{\top}, for\ i = 1, \ldots, m. \tag{3.60} $$

Use p_i and r_{ij} to represent the elements in ${}^{b}\boldsymbol{p}_{bl}$ and ${}^{b}\boldsymbol{R}_{bl}$, respectively. Now we can construct the equation set with the p_i as the unknown parameters.

$$ \begin{cases} (z_1 r_{31} - r_{11})(p_1 - f_{1,1}) + (z_1 r_{32} - r_{12})(p_2 - f_{1,2}) + (z_1 r_{33} - r_{13})(p_3 - f_{1,3}) = 0 \\ \vdots \\ (z_m r_{31} - r_{11})(p_1 - f_{m,1}) + (z_m r_{32} - r_{12})(p_2 - f_{m,2}) + (z_m r_{33} - r_{13})(p_3 - f_{m,3}) = 0 \end{cases} $$

The coefficient matrix is

$$ \begin{bmatrix} z_1 r_{31} - r_{11} & z_1 r_{32} - r_{12} & z_1 r_{33} - r_{13} \\ \vdots & \ddots & \vdots \\ z_m r_{31} - r_{11} & z_m r_{32} - r_{12} & z_m r_{33} - r_{13} \end{bmatrix}, \tag{3.61} $$

which is not full rank since each row in this matrix is the linear combination of the first and third rows of matrix ${}^{b}\boldsymbol{R}_{bl}$. This fact indicates that there will be no unique solution no matter how many LED's are measured simultaneously.

The prior values of ${}^{l}_{b}\boldsymbol{R}$ and ${}^{b}\boldsymbol{p}_{bl}$ can be measured roughly by a tape. The vehicle's pose is accurately known by putting the vehicle at a well measured position. Nonlinear optimization methods could be employed to obtain the accurate extrinsic parameter estimation. Some nonlinear optimization methods such as Gradient Descent, Newton method and Gauss-Newton method are introduced in Appendix A.

Using ${}^{l}\bar{\boldsymbol{q}}_{b}$ to denote the quarternion of ${}^{l}_{b}\boldsymbol{R}$, and defining $\boldsymbol{x}_e = \begin{bmatrix} {}^{l}\bar{\boldsymbol{q}}_{b} & {}^{b}\boldsymbol{p}_{bl} \end{bmatrix}^{\top}$,

the estimation of the extrinsic parameters is calculated according to

$$\hat{\boldsymbol{x}}_e^{k+1} = \hat{\boldsymbol{x}}_e^k + \delta\hat{\boldsymbol{x}}_e^k \tag{3.62}$$

$$\delta\hat{\boldsymbol{x}}_e^k = \left(\boldsymbol{H}_g^\top \boldsymbol{R}_g^{-1}\boldsymbol{H}_g\right)^{-1}\boldsymbol{H}_g^\top \boldsymbol{R}_g^{-1}\left(\boldsymbol{z} - \boldsymbol{h}(\hat{\boldsymbol{x}}_e^k)\right) \tag{3.63}$$

$$\boldsymbol{P}_e = \left(\boldsymbol{H}_g^\top \boldsymbol{R}_g^{-1}\boldsymbol{H}_g\right)^{-1}, \tag{3.64}$$

where$\boldsymbol{R}_g$ and $\boldsymbol{H}_g$ are constructed in Sec. 3.5.1, $\hat{\boldsymbol{x}}_e^k$ is the estimation of $\boldsymbol{x}_e$ at the kth iteration.

3.4.2 Initialization

We firstly introduce the initialization method using the camera measurements, then the similar method using linear array measurements will be derived. According to the definition of u and v in Eqn. (3.22), using (3.18) and (3.19), we have

$${}^cZ\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = {}^c\boldsymbol{p}_{cL} = \begin{bmatrix} {}^c_n\boldsymbol{R} & {}^c\boldsymbol{p}_{cn} \end{bmatrix}\begin{bmatrix} {}^n\boldsymbol{p}_{nL} \\ 1 \end{bmatrix} \tag{3.65}$$

Reorganizing Eqn. (3.65), we have

$${}^b_c\boldsymbol{R}\begin{bmatrix} {}^cZ\ u \\ {}^cZ\ v \\ {}^cZ \end{bmatrix} = \begin{bmatrix} {}^b_n\boldsymbol{R} & {}^b\boldsymbol{p}_{cb} - {}^b_n\boldsymbol{R}\ {}^n\boldsymbol{p}_{nb} \end{bmatrix}\begin{bmatrix} {}^n\boldsymbol{p}_{nL} \\ 1 \end{bmatrix}, \tag{3.66}$$

where (u, v), ${}^cZ$, ${}^b_c\boldsymbol{R}$, ${}^b\boldsymbol{p}_{cb}$ and ${}^n\boldsymbol{p}_{nL}$ are all known, ${}^b_n\boldsymbol{R}$ and ${}^n\boldsymbol{p}_{nb}$ are

$${}^b_n\boldsymbol{R} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.67}$$

$${}^n\boldsymbol{p}_{nb} = \begin{bmatrix} n & e & 0 \end{bmatrix}^\top. \tag{3.68}$$

Define matrix $\mathcal{P} = \begin{bmatrix} {}^{b}_{n}\boldsymbol{R} & {}^{b}\boldsymbol{p}_{cb} - {}^{b}_{n}\boldsymbol{R}{}^{n}\boldsymbol{p}_{nb} \end{bmatrix}$. By (3.67) and (3.68), $\mathcal{P} = [p_{ij}]$ is

$$\begin{bmatrix} \cos\psi & \sin\psi & 0 & {}^{b}x_{cb} - n\cos\psi - e\sin\psi \\ -\sin\psi & \cos\psi & 0 & {}^{b}y_{cb} + n\sin\psi - e\cos\psi \\ 0 & 0 & 1 & {}^{b}z_{cb} \end{bmatrix}. \tag{3.69}$$

From (3.69), only elements $[p_{11}, p_{12}, p_{14}, p_{24}]$ are unknown. Using the following notation

$$a = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = {}^{n}\boldsymbol{p}_{nL} \tag{3.70}$$

$$b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = {}^{b}_{c}\boldsymbol{R} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{3.71}$$

to reorganize Eqn. (3.66), we have

$$\begin{cases} a_1 p_{11} + a_2 p_{12} + p_{14} = \dfrac{b_1}{b_3}(a_3 + p_{34}) \\ a_2 p_{11} - a_1 p_{12} + p_{24} = \dfrac{b_2}{b_3}(a_3 + p_{34}) \end{cases} \tag{3.72}$$

Notice that this equation array has two linear equations and four unknown variables $(p_{11}, p_{12}, p_{14}, p_{24})$. In order to solve (3.72), at least two LEDs should be measured simultaneously. The solution is found by the standard least square (LS). A deeper insight of eqn. (3.72) will find that at least two laterally separated (different a_1 and a_2) LEDs required to solve this equation set. Fig. 3.9 illustrates the situation that the two vehicle positions can not be distinguished by merely measuring these two LEDs.

An additional known constraint is that the unknown variables in (3.72) should strictly satisfy

$$p_{11}^2 + p_{12}^2 = 1. \tag{3.73}$$

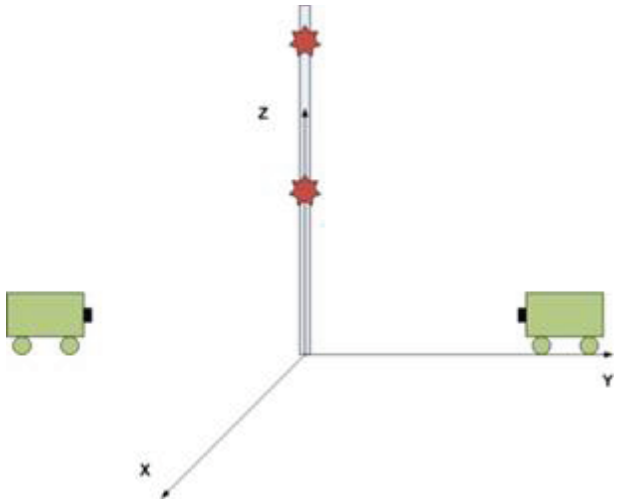


Figure 3.9: Two LEDs not laterally separated.

Due to its nonlinearity, we didn't mention this equation when calculating the unknown variables using the linear model (3.72). When this constraint is incorporated, it becomes a nonlinear nonconvex optimization problem that can be initialized with the estimate of (n, e, ψ) computed by the solution of (3.72). Results will be discussed in Section 3.5 and our former paper [86].

After some modifications, the initialization method still works when the camera is replaced by the linear array. Here the initialization process requires at least 4 LED's measurements due to its one-dimensional sensitivity, compared with two LEDs needed for the camera measurements. Similar to eqn. (3.66), the linear array measurement equation could be written as

$$\begin{bmatrix} {}^{l}z_m \boldsymbol{u}_m \\ {}^{l}z_m \end{bmatrix} = \boldsymbol{J}\,{}^{l}_{b}\boldsymbol{R} \begin{bmatrix} {}^{b}_{w}\boldsymbol{R} & {}^{b}\boldsymbol{p}_{lb} - {}^{b}_{w}\boldsymbol{R}\,{}^{w}\boldsymbol{p}_{wb} \end{bmatrix} \begin{bmatrix} {}^{w}\boldsymbol{p}_{wf} \\ 1 \end{bmatrix} \tag{3.74}$$

where

$$\boldsymbol{J} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.75}$$

and $\boldsymbol{u}_m$ denotes the mth measurement. Reorganizing (3.74), we have

$$\begin{aligned}(r_{11}a_{1m} + r_{12}a_{2m})p_{11} + (r_{11}a_{2m} - r_{12}a_{1m})p_{12} \\ +r_{11}p_{14} + r_{12}p_{24} - u_m{}^{l}z_m = -r_{13}(a_{3m} + {}^{b}z_{lb})\end{aligned} \tag{3.76}$$

$$\begin{aligned}(r_{31}a_{1m} + r_{32}a_{2m})p_{11} + (r_{31}a_{2m} - r_{32}a_{1m})p_{12} \\ +r_{31}p_{14} + r_{32}p_{24} - {}^{l}z_m = -r_{33}(a_{3m} + {}^{b}z_{lb})\end{aligned} \tag{3.77}$$

where the definition of r_{ij}, a_i and p_{ij} are same with that in eqn. (3.61) and (3.72). For each LED measurement, we obtain two linear equations. Since ${}^{l}z_m$ varies for different LED's, to solve the linear equation set, at least 4 LED's should be measured simultaneously. Similarly, these 4 LEDs can not simultaneously have the same x and y coordinates.

3.5 Results

The matrix $\boldsymbol{H}_e$ is the linearized matrix that evaluated at $\hat{\boldsymbol{x}}_e^k$.

$$\boldsymbol{H}_e = \boldsymbol{\mathcal{H}}_1\boldsymbol{\mathcal{H}}_3 \tag{3.78}$$

$$\begin{aligned}\boldsymbol{\mathcal{H}}_3 &= \begin{bmatrix} \frac{\partial^l \boldsymbol{p}_{lf}}{\partial^l \bar{\boldsymbol{q}}_b} & \frac{\partial^l \boldsymbol{p}_{lf}}{\partial^b \boldsymbol{p}_{bl}} \end{bmatrix} \\ \frac{\partial^l \boldsymbol{p}_{lf}}{\partial^l \bar{\boldsymbol{q}}_b} &= \Psi\left({}^{l}\bar{\boldsymbol{q}}_b, {}^{b}\boldsymbol{p}_{bf} - {}^{b}\boldsymbol{p}_{bl}\right) \\ \frac{\partial^l \boldsymbol{p}_{lf}}{\partial^b \boldsymbol{p}_{bl}} &= -{}^{l}_{b}\boldsymbol{R}.\end{aligned} \tag{3.79}$$

If several lED's are measured simultaneously, $\boldsymbol{H}_e$ is constructed by stacking the $\boldsymbol{H}_e$ calculated above together.

3.5.1 Extrinsic Parameters Calibration

Let φ represent the rotation from the computed frame $\hat{l}$ to the true frame l.

$$\begin{aligned} {}^{\hat{l}}_{l}\boldsymbol{R} &= \boldsymbol{I} + \frac{\sin\varphi}{\varphi}[\varphi\times] + \frac{1-\cos\varphi}{\varphi^2}[\varphi\times][\varphi\times] \\ &\approx \boldsymbol{I} + [\varphi\times]. \end{aligned} \tag{3.80}$$

Then we have ${}^{l}_{b}\boldsymbol{R} = {}^{l}_{\hat{l}}\boldsymbol{R}\,{}^{\hat{l}}_{b}\boldsymbol{R} \approx (\boldsymbol{I} - [\varphi\times])\,{}^{\hat{l}}_{b}\boldsymbol{R}$. If we define $\delta\boldsymbol{x}_e = \begin{bmatrix} \delta{}^{b}\boldsymbol{p}_{bl} & \varphi \end{bmatrix}^\top$, then $\boldsymbol{\mathcal{H}}_3 = \begin{bmatrix} -{}^{l}_{b}\hat{\boldsymbol{R}} & [{}^{l}\hat{\boldsymbol{p}}_{lf}\times] \end{bmatrix}$.

Assuming N LED's are measured by the linear array, and the initial covariance of the extrinsic parameters is $\boldsymbol{P}_{e0} \in \mathbb{R}^{6\times 6}$, the Gauss-Newton correction is computed as

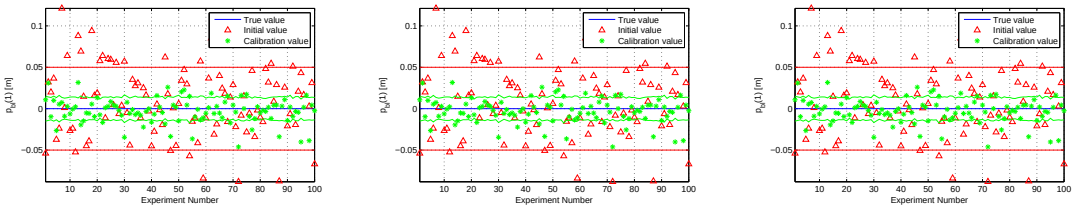
$$\delta\hat{\boldsymbol{x}}_e = \left(\boldsymbol{H}_g^\top \boldsymbol{R}_g^{-1}\boldsymbol{H}_g\right)^{-1}\boldsymbol{H}_g^\top \boldsymbol{R}_g^{-1}\boldsymbol{r}_e, \tag{3.81}$$

where

$$\boldsymbol{H}_g = \begin{bmatrix} \boldsymbol{H}_{e1}^\top & \cdots & \boldsymbol{H}_{eN}^\top & \boldsymbol{I}_{6\times 6} \end{bmatrix}^\top \tag{3.82}$$

$$\boldsymbol{R}_g = diag\{\begin{bmatrix} \boldsymbol{R}_1^\top & \cdots & \boldsymbol{R}_N^\top & \boldsymbol{P}_{e0} \end{bmatrix}\} \tag{3.83}$$

$$\boldsymbol{r}_e = \begin{bmatrix} \tilde{z}_1 - \hat{z}_1 & \cdots & \tilde{z}_N - \hat{z}_N & \boldsymbol{x}_{e0} - \hat{\boldsymbol{x}}_e \end{bmatrix}^\top \tag{3.84}$$

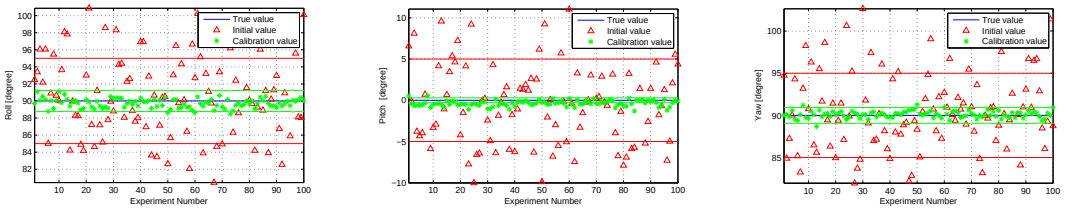


(a) Estimated ${}^{l}p_{bl}(1)$. (b) Estimated ${}^{l}p_{bl}(2)$. (c) Estimated ${}^{l}p_{bl}(3)$.

Figure 3.10: Estimated translation ${}^{l}p_{bl}$.

The first three figures show the calibration results of ${}^{l}\boldsymbol{p}_{bl}$. The red and green lines represent the standard deviation of the initial guess and calibration results, respectively. To generate the result, 8 LED's are measured by the linear array. The

experiment is repeated for 100 trials. In each trial, we randomly choose the initial guess value according to its true value and covariance $\boldsymbol{P}_{e0}$. In the figures, ${}^{l}\boldsymbol{p}_{bl}(3)$ representing the height of linear array in the body frame is unchanged after the calibration process, which is due to the pose of the linear array and its one-dimensional sensitivity.



(a) Estimated roll. (b) Estimated pitch. (c) Estimated yaw.

Figure 3.11: Estimated Euler angles.

3.5.2 Initialization

We firstly test the initialization result using camera measurements. The vehicle was placed at $n = -0.575\,\text{m}$, $e = -2.046\,\text{m}$ and $\psi \approx 0.401\,\text{deg}$, which is known to an accuracy of 0.002 m, 0.006 m and 0.002 deg, respectively. In this section, these coordinate will be referred to as 'correct.' The vehicle is stationary at the stated location throughout the results reported in this section. There are 8 LEDs in front of the vehicle (camera). The LED navigation frame position vectors are listed in Table 3.1.

The initialization algorithm is firstly tested using different two-LED measurement combinations to compute the state of the vehicle. The computed coordinates and yaw angle computed using eqn. (3.72, 3.73) and their error standard deviation as computed using eqn. (3.64) are given in Table 3.2, along with the LED identifiers. The initialization state works well for all combinations with the exception of LED combinations: (1,2), (3,4),(6,8). For each of the three combinations, both LED's have the

Table 3.1: LED coordinates in meters in the NED frame.

LED ID	n	e	d
1	2.797	-1.500	-1.500
2	2.797	-1.500	-2.000
3	2.797	-0.500	-2.000
4	2.797	-0.500	-1.500
5	2.797	-4.250	-2.000
6	2.797	-3.500	-2.000
7	2.797	-2.500	-2.000
8	2.797	-3.500	-1.500

same n and e coordinates differing only in their d coordinate. In this case, the line that connects the two LED's is parallel to the vehicle frame z-axis and the NED frame d-axes. Such LED combinations cause the equations in (3.72) being linear dependent. Due to this, the vehicle's state can not be uniquely determined.

There are many combinations of three or more LED's. Representative initialization results using three and all eight LEDs are shown in Table. 3.3. The data in these Tables confirms that the accuracy of initialization result depends on the number and spatial distribution of the LEDs that are used.

Fig. 3.12 contains scatter plots of the computed vehicle coordinates (red dots) for the 2 (top) and 3 (bottom) LED combinations. The blue asterisk ('*') marks the coordinates considered to be correct. The positions computed using 3 LED's are more concentrated.

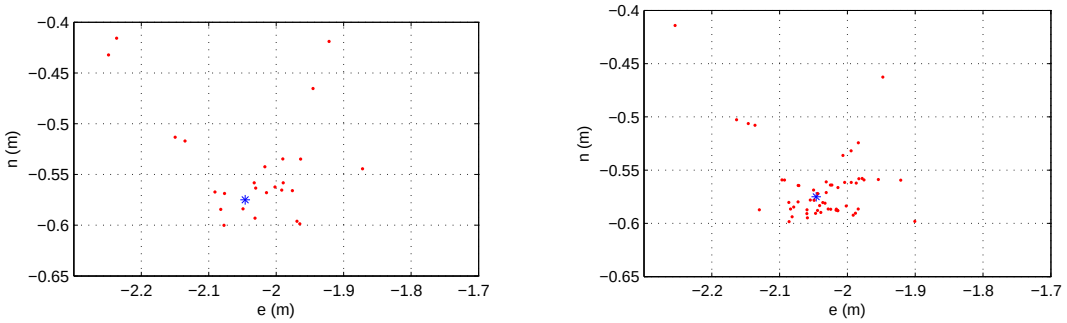
Next, we will show the initialization results using linear array measurements. The correct vehicle state is approximately $\begin{bmatrix} -2.5 & 0 & -45 \end{bmatrix}^\top$. The state is firstly

LED	n	e	ψ	σ_n	σ_e	σ_ψ
1,2	NaN	NaN	NaN	3.7e+4	7.3e+3	1.2e+7
1,3	-0.432	-2.248	3.610	0.092	0.150	2.789
1,4	-0.416	-2.237	3.511	0.096	0.150	2.795
1,5	-0.565	-1.992	-1.325	0.055	0.114	1.756
1,6	-0.562	-2.002	-1.146	0.046	0.150	2.387
1,7	-0.558	-1.990	-1.328	0.033	0.261	4.362
1,8	-0.544	-1.872	-3.221	0.055	0.166	2.672
2,3	-0.513	-2.150	1.599	0.056	0.124	2.151
2,4	-0.517	-2.135	1.340	0.056	0.132	2.247
2,5	-0.568	-2.014	-1.052	0.030	0.069	1.034
2,6	-0.563	-2.030	-0.753	0.026	0.096	1.508
2,7	-0.558	-2.033	-0.664	0.025	0.185	3.096
2,8	-0.566	-1.976	-1.628	0.028	0.109	1.712
3,4	NaN	NaN	NaN	0.000	7.5e+17	0.000
3,5	-0.593	-2.031	-0.641	0.016	0.066	0.852
3,6	-0.584	-2.049	-0.329	0.016	0.075	1.064
3,7	-0.569	-2.077	0.160	0.030	0.090	1.434
3,8	-0.599	-1.965	-1.560	0.018	0.100	1.414
4,5	-0.600	-2.077	-0.061	0.020	0.092	1.190
4,6	-0.584	-2.082	0.119	0.017	0.093	1.324
4,7	-0.567	-2.091	0.361	0.030	0.099	1.565
4,8	-0.596	-1.969	-1.519	0.019	0.148	2.098
5,6	-0.465	-1.945	-2.750	0.084	0.122	2.165
5,7	-0.535	-1.990	-1.619	0.047	0.075	1.276
5,8	-0.419	-1.922	-3.447	0.113	0.135	2.569
6,7	-0.542	-2.017	-1.118	0.054	0.143	2.438
6,8	NaN	NaN	NaN	7.5e+11	0.000	0.000
7,8	-0.535	-1.964	-2.016	0.056	0.150	2.537

Table 3.2: initialization results using 2 LEDs

LED	n	e	ψ	σ_n	σ_e	σ_ψ
1,2,3	-0.503	-2.164	1.873	0.050	0.107	1.889
1,2,4	-0.506	-2.146	1.573	0.050	0.113	1.962
1,3,5	-0.588	-2.014	-0.866	0.016	0.047	0.647
1,3,7	-0.564	-2.072	0.116	0.027	0.078	1.279
1,4,5	-0.588	-2.017	-0.843	0.019	0.056	0.781
1,4,8	-0.590	-1.988	-1.264	0.018	0.078	1.162
1,5,8	-0.583	-2.041	-0.612	0.038	0.085	1.328
2,3,4	-0.508	-2.136	1.453	0.055	0.116	1.999
2,3,8	-0.588	-2.014	-0.894	0.016	0.066	0.979
2,4,6	-0.578	-2.055	-0.271	0.016	0.065	0.960
2,5,7	-0.561	-2.004	-1.216	0.024	0.052	0.813
3,5,7	-0.590	-2.039	-0.550	0.014	0.042	0.590
3,5,8	-0.595	-2.059	-0.300	0.014	0.053	0.722
5,6,7	-0.524	-1.984	-1.767	0.042	0.070	1.208
6,7,8	-0.536	-2.007	-1.354	0.054	0.134	2.271
8LEDs	-0.586	-2.036	-0.591	0.010	0.032	0.465

Table 3.3: Initialization results using 3 and 8 LEDs

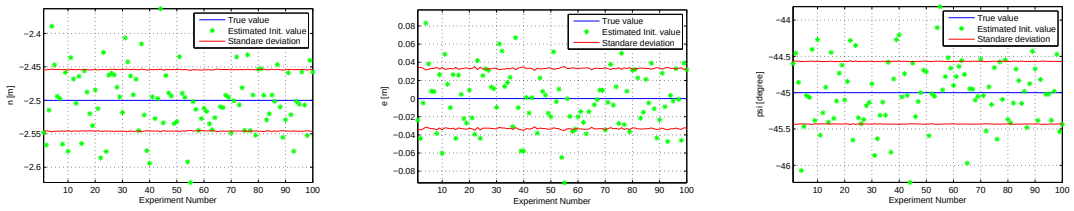


(a) Initialization results using 2 LEDs.

(b) Initialization results using 3 LEDs.

Figure 3.12: Initialization results. The blue asterisk marks the location that is considered to be correct.

computed according to the linear equations (3.76) and (3.77). The estimation results are further refined by minimizing the residual using nonlinear optimization methods. In these figures, the results are optimized using Gauss-Newton method. The error covariance of the estimation result is also computed based on the covariance of the linear array measurement noise.



(a) Estimated n. (b) Estimated e. (c) Estimated ψ.

Figure 3.13: Initialization results.

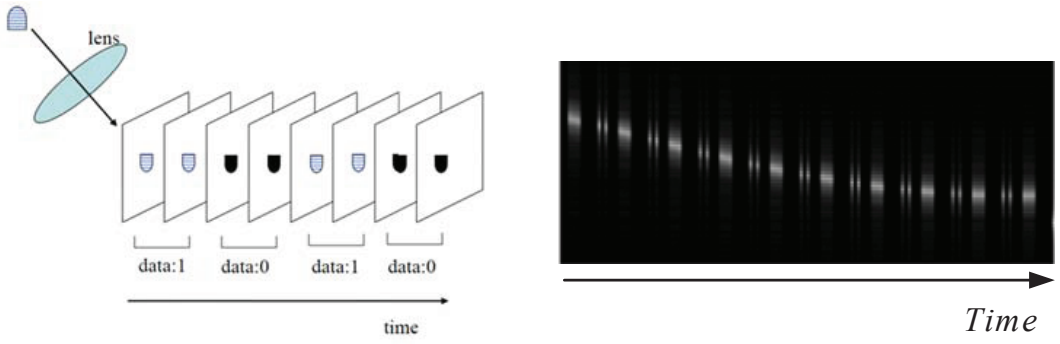
Chapter 4

Data Recovery in a Stationary Sensor

4.1 Introduction

LED's are bright point light sources, the pixels that they project onto have high intensity values. The LED projections are bright blobs and segmentations in the image sensor and linear array, which are shown in Fig. 4.1. As the linear array platform moves, the LED projection moves along the one-dimensional pixel array.

In the linear array-LED navigation system discussed in Chapter 3, each LED's ID was used to assist data association. Extracting the LED ID is equivalent to recovering the LED's on-off data pattern from a sequence of linear array measurements. The LED's on-off status can be recovered only after its position in each scan is found. Finding the LED projection is not straightforward since: 1) There be multiple pixels measuring high intensity (e.g., LED's, clutter, and noise). The incorrect measurements are referred to as false detections. 2) The position of a turned-off LED in the image can not be extracted



(a) Camera continuously takes images of a scene with an LED light [31].

(b) Linear array continuously scans a scene with an LED light.

Figure 4.1: LED data in a sequence of photo-detector scans

by evaluating the intensity value of each pixel.

LED status recovery is a process that first identifies which measurement is from each LED in each time step. A simple approach is, in each scan, to choose the measurement that is closest to the predicted measurement position in a Mahalanobis sense. This fails because the LED is off in some scans. Another method that avoids this problem might use the probabilistic data association filter [10]. At each scan, it would compute the probability of each association hypothesis (which one is from the LED or none is from LED) and choose the most probable one. Though this method works, there is a very low probability to obtain a correct ID, because correctly extracting an ID requires that the data association in each time step of a scan sequence is correct, which could be extremely hard especially when the ID is long.

To solve this problem, the measurements in a sequence of images need to be considered jointly. The measurements of the same object (LED) at multiple time instances should be matched together to recover the data sequence. However, the existing descriptors for image features can not be applied to the linear array measurement, so that they can not be used to match the measurements in different linear array scans.

More sophistic methods could be constructed within the framework of multiple hypotheses tracking (MHT) [67, 9, 47]. MHT takes a deferred-decision approach in which they maintain the complete set of possible data associations within a sliding window, putting off hard decisions as long as possible [25]. The ultimate decision is made by evaluating the probability of each sequence of data association hypothesis. Because the number of possible sequences increases exponentially, this approach is expensive to implement, especially for a system having fast frame rate.

To accurately and efficiently recover the LED status, we proposed a method based on Viterbi algorithm [23, 77]. The Viterbi algorithm is a dynamic programming algorithm that can be adapted to generate the most likely state path given a sequence of observations. In this chapter, we focus on the typical case where the frame rate of the linear array sensor is high relative to the array motion bandwidth. This implies that the pixel projection changes smoothly from one frame to the next. With this assumption, the LED projection on the linear array at current time step is highly dependent on that in the former time step. This yields a Markov model representation. The Markov property leaves opportunity to break the LED status recovery problem down into simpler subproblems.

This chapter is organized as follows: Section 4.2 defines the state and linear array measurement, and formulates the problem mathematically. Section 4.3.1 briefly reviews the Viterbi algorithm. Section 4.3.2 and Section 4.4.2 analyze the state transition and measurement model, respectively. Section 4.4 shows the experimental results.

4.2 Problem Formulation

In this chapter, we use symbol $\boldsymbol{\chi}$ instead of $\boldsymbol{x}$ to represent the kinematic state of the rover. The new definition of symbol $\boldsymbol{x}$ in this chapter will be introduced in Sec. 4.2.3. The rover's kinematic state $\boldsymbol{\chi}$ evolves over time according to eqn. (3.1). The l-th LED location $\boldsymbol{F}_l$ is put explicitly into the measurement model defined in (3.8), so the LED's projection is

$$\boldsymbol{z}(t) = \boldsymbol{h}(\boldsymbol{\chi}(t), \boldsymbol{F}_l). \tag{4.1}$$

Each LED is switching its "on" or "off" status to communicate information to the rover.

Given a time interval $\lambda \in [t_s, t]$, each projected LED position $\boldsymbol{z}(\lambda)$ defines a trajectory across the linear array. Detection of the projected LED position depends on the LED "on" or "off" status, as well as environmental conditions and interference from other light sources. The accuracy of each LED's recovered on-off status is highly dependent on the accuracy of our knowledge of this LED trajectory. Therefore, the data recovery and trajectory estimation problems are coupled. Improvement in the solution of either problem enhances the solution of the other.

The purpose of this chapter is to develop an algorithm that simultaneously estimates the most likely LED trajectory and data sequence for each time interval $\lambda \in [t_s, t]$. The algorithm that we develop is general and independent of vehicle motion assumptions. To simplify algorithm implementation, we make the following assumption which simplifies the pixel state transition model. Assume that the change in the projected LED position is less that ϵ pixels per frame:

$$\dot{\boldsymbol{z}}(\lambda) \leq F_I \epsilon, \forall \lambda \in [t_s, t], \tag{4.2}$$

where F_I is the frame rate. This is equivalent to

$$\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\chi}}\dot{\boldsymbol{\chi}} \leq F_I\epsilon, \forall\lambda \in [t_s, t], \tag{4.3}$$

which will allow derivation of a diagonally dominant transition matrix in Section 4.4.2.

4.2.1 Linear Array Measurement

In the former chapters, we have discussed the linear array measurement model. Here we will introduce more details of the light intensity measurement for each pixel in the linear array. A linear array measures the light intensity

$$\boldsymbol{Y}(t) \triangleq \{y_i(t)\}_{i=1}^{N}$$

received by each of N pixels. Throughout this chapter, we use the counter i to indicate the value associated with the i-th pixel. Pixels at the LED projection location have higher intensity values when the LED is on. In this chapter, we threshold the intensity values

$$\boldsymbol{\zeta}_i(t) = \begin{cases} 1 & \text{if } y_i \geq \tau \\ 0 & \text{otherwise,} \end{cases} \tag{4.4}$$

to generate a set of candidate LED pixel positions:

$$\boldsymbol{Z}(t) \triangleq \{z_j(t)\}_{j=1}^{m_t}, \tag{4.5}$$

where m_t is the total number of pixels whose intensity values are greater than the threshold τ at time t, and $\boldsymbol{z}_j(t)$ is the index of the jth "on" pixel. Note $\boldsymbol{z}_j(t) \in [1, N]$ is an integer.

The pixel index z_j represents a position in the linear array. A pixel in the set $\boldsymbol{Z}(t)$ may be the projection of an "on" LED or a false detection due to noise or

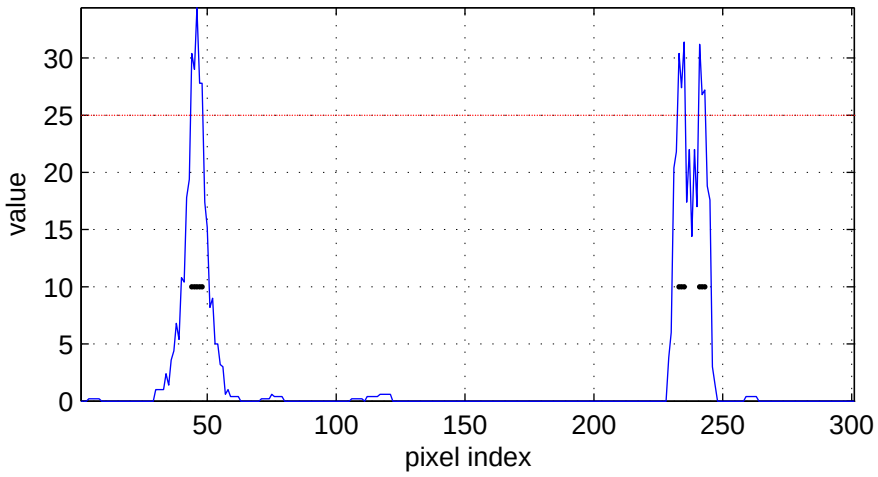


Figure 4.2: Linear array measurement

clutter. The other pixels ($i \notin \boldsymbol{Z}(t)$) may each be the projection of an "off" LED, a missed detection, or a pixel onto which no LED projects.

Fig. 4.2 illustrates an original linear array scan (blue) and the processed measurements (black). The threshold τ is graphed as the constant red curve. For the processed measurements, each black dot is an element of the set $\boldsymbol{Z}$ defined in eqn. (4.5). The value of black dots are set as 10 to be highlighted.

For theoretical convenience, an LED is assumed to project on only one pixel. We also assume that other lights sources (clutter) and noise project independently to each pixel. For example, a bright segment that is composed of n connected pixels will be viewed as n independent measurements. Fig. 4.2 illustrates that these simplifying assumptions are not strictly valid. Section 4.3.3 proves that the final result is insensitive to the assumptions.

4.2.2 LED Measurement Prior

Our approach assumes a navigation system [22, 84, 85, 86] that maintains an estimate $\hat{\boldsymbol{\chi}}(t)$ of the system state $\boldsymbol{\chi}(t)$ in realtime. This system state estimate enables prediction of the trajectory of each LED across the linear array. For example, the l-th LED at location $\boldsymbol{F}_l$, is expected to project to $\hat{\boldsymbol{z}}(t) = h(\hat{\boldsymbol{\chi}}(t), \boldsymbol{F}_l)$ at time t. The covariance of the projected LED position is $\boldsymbol{S}_l(t) = \boldsymbol{H}\boldsymbol{P}\boldsymbol{H}^\top + \boldsymbol{R}$ where $\boldsymbol{H}$ is the Jacobean of $\boldsymbol{h}$, $\boldsymbol{P}$ is the prior covariance of $\hat{\boldsymbol{\chi}}(t)$, and $\boldsymbol{R}$ is the measurement noise covariance matrix. Each of these quantities is computed by the navigation system for each LED.

Given time interval $\lambda \in [t_s, t]$, it is straightforward to compute both $\hat{\boldsymbol{z}}(\lambda)$ and $\boldsymbol{S}_l(\lambda)$. These quantities define the prior distribution for the LED trajectory that can focus the algorithm. In the following presentation, we define $t_s = 0$ and $\lambda \in [0, K]$, without loss of generality. We also will only consider a single LED; therefore, we drop the l subscript.

4.2.3 LED State Definition

In the first linear array scan (i.e., $t = 0$), the predicted LED position $\hat{z}_0$ and its error covariance $\boldsymbol{S}_0$ are known. Define the detection range to be the interval $\left[\lfloor \hat{z}_0 - \rho\sqrt{\boldsymbol{S}_0} \rfloor, \lceil \hat{z}_0 + \rho\sqrt{\boldsymbol{S}_0} \rceil\right]$ where $\lfloor a \rfloor$ and $\lceil a \rceil$ represent the largest integer smaller than a and the smallest integer larger than a, respectively. The parameter ρ determines the probability that the actual LED projection is in the specified segment. For example, setting $\rho \geq 3$ yields the probability being greater than 0.997.

Given the assumption in eqn. (4.3), the sequence of detection intervals

$$\left[\lfloor \hat{z}_\lambda - \rho\sqrt{\boldsymbol{S}_\lambda} \rfloor, \lceil \hat{z}_\lambda + \rho\sqrt{\boldsymbol{S}_\lambda} \rceil\right]$$

changes very little. Therefore, to find the most likely LED trajectory, it is sufficient to

focus on this smaller array segment, instead of the whole linear array. Effectively, this decreases the search space and the computational requirements of the algorithm.

Assume there are N pixels in the detection range, and they are indexed from left to right. The LED state space is

$$\boldsymbol{X}(\lambda) \in \boldsymbol{X}, \text{ where } \boldsymbol{X} \triangleq \{1, 2, 3, \ldots, N\}. \tag{4.6}$$

LED state $\boldsymbol{X}(\lambda) = i$ means that the LED projects onto the i-th linear array pixel at step λ. Fig. 4.3 illustrates the detection range and LED state.

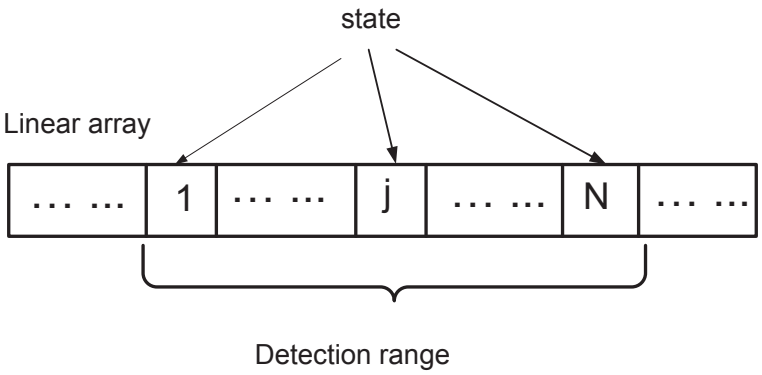


Figure 4.3: LED state definition.

4.2.4 Problem Statement

Given a sequence of linear array thresholded measurement sets for $\lambda \in [0, K]$:

$$\boldsymbol{Z}^K = \{\boldsymbol{Z}(\lambda)\}_{\lambda=1}^K,$$

where $\boldsymbol{Z}(\lambda)$ is defined in eqn. (4.5), find the optimal state sequence $\boldsymbol{X}^K = \{\boldsymbol{X}(\lambda)\}_{\lambda=1}^K$ such that

$$\boldsymbol{X}^* = \arg\max_{\boldsymbol{X}^K} \boldsymbol{p}\left(\boldsymbol{X}^K \mid \boldsymbol{Z}^K\right), \tag{4.7}$$

where $\boldsymbol{p}\left(\boldsymbol{X}^K \mid \boldsymbol{Z}^K\right)$ is the posterior probability mass function (PMF) of $\boldsymbol{X}^K$. Once $\boldsymbol{X}^*$ is available, it is straightforward to extract the LED data sequence and to correct the rover vehicle trajectory $\boldsymbol{\chi}(\lambda)$.

4.3 Method

This section proposes a method to solve the problem defined in Sec. 4.2. This method is based on the Viterbi algorithm [23, 77], which is a dynamic programming algorithm that can be adapted to generate the most likely state trajectory (i.e., path) given a sequence of observations. Sec. 4.3.1 reviews the Viterbi algorithm. Sec. 4.4.2 and Sec. 4.3.2 discuss the transition matrix and measurement model, respectively. Sec. 4.3.3 analyzes the influence of optical distortion that affects the width of the LED projection and presents the path recovery algorithm.

4.3.1 Viterbi Algorithm

The Viterbi algorithm is described in Table 4.1. A brief review derivation is as follows.

The probability of a state sequence $\boldsymbol{X}^t$ given the first t observation sets $\boldsymbol{Z}^t$ is

$$
\begin{aligned}
& \boldsymbol{p}(\boldsymbol{X}^t \mid \boldsymbol{Z}^t) \\
= \; & \frac{1}{c}\boldsymbol{p}(\boldsymbol{X}^t, \boldsymbol{z}(t) \mid \boldsymbol{Z}^{t-1}) && (4.8) \\
= \; & \frac{1}{c}\boldsymbol{p}(\boldsymbol{z}(t) \mid \boldsymbol{X}^t, \boldsymbol{Z}^{t-1})\boldsymbol{p}(\boldsymbol{X}^t \mid \boldsymbol{Z}^{t-1}) && (4.9) \\
= \; & \frac{1}{c}\boldsymbol{p}(\boldsymbol{z}(t) \mid \boldsymbol{x}(t))\boldsymbol{p}(x(t) \mid \boldsymbol{x}(t-1))\boldsymbol{p}(\boldsymbol{X}^{t-1} \mid \boldsymbol{Z}^{t-1}) && (4.10)
\end{aligned}
$$

where $c = \boldsymbol{p}(\boldsymbol{z}(t) \mid \boldsymbol{Z}^{t-1})$. The derivation of eqn. (4.10) is possible due to two facts. First, $\boldsymbol{p}(\boldsymbol{z}(t) \mid \boldsymbol{X}^t, \boldsymbol{Z}^{t-1}) = \boldsymbol{p}(\boldsymbol{z}(t) \mid \boldsymbol{x}(t))$ because the measurement depends only on the

Table 4.1: Viterbi algorithm

Input: observation sequence $\boldsymbol{Z}^t$

state prior probability $\boldsymbol{p}_0$

Output: the most probable state sequence $\boldsymbol{X}^t$

1. Initialize $V_{1,j}$

   $V_{1,j} = \boldsymbol{p}(\boldsymbol{z}(1) \mid j) \cdot \boldsymbol{p}_0(j)$

2. Compute $V_{t,j}$ from $V_{t-1,i}$, transition matrix $\boldsymbol{A}(t-1)$ and $\boldsymbol{z}(t)$

   $V_{t,j} = \boldsymbol{p}(\boldsymbol{z}(t) \mid j) \max_i\{a_{ij}(t-1) \cdot V_{t-1,i}\}$

3. Compute the t-th column of matrix $\boldsymbol{Ptr}$. The value of $\boldsymbol{Ptr}(j,t)$ is state i that used to compute $V_{t,j}$.

4. Retrieve the most likely state sequence

   $\boldsymbol{x}_t = \arg\max_{j \in \boldsymbol{X}}(V_{t,j})$

   $\boldsymbol{x}_{t-1} = \boldsymbol{Ptr}(\boldsymbol{x}_t, t)$

current state. Second, $\boldsymbol{p}(\boldsymbol{X}^t \mid \boldsymbol{Z}^{t-1}) = \boldsymbol{p}(\boldsymbol{x}(t) \mid \boldsymbol{x}(t-1))\boldsymbol{p}(\boldsymbol{X}^{t-1} \mid \boldsymbol{Z}^{t-1})$ because the state is a Markov process.

Let $V_{t,j}$ be the probability of the most probable state sequence that has j as its final state, given the first t observation sets. From eqn. (4.10), we have

$$\begin{aligned} V_{t,j} &= \max_{\boldsymbol{X}^{t-1}}\{\boldsymbol{p}(\boldsymbol{z}(t) \mid j)\boldsymbol{p}(j \mid \boldsymbol{x}(t-1))\boldsymbol{p}(\boldsymbol{X}^{t-1} \mid \boldsymbol{Z}^{t-1})\} \\ &= \boldsymbol{p}(\boldsymbol{z}(t) \mid j) \max_i\{\boldsymbol{p}(j \mid \boldsymbol{x}(t-1) = i) \cdot V_{t-1,i}\} \\ &= \boldsymbol{p}(\boldsymbol{z}(t) \mid j) \max_i\{a_{ij}(t-1) \cdot V_{t-1,i}\}, \end{aligned} \tag{4.11}$$

where $a_{ij}(t-1) = \boldsymbol{p}(j \mid \boldsymbol{x}(t-1) = i)$ represents the probability of the transition of state from i at time $t-1$ to j at time t.

Given that the state space is discrete with maximum number of states N, this

state transition can be represented by the matrix

$$\boldsymbol{A}(t) = [a_{ij}(t)] \in \mathbb{R}^{N \times N}. \tag{4.12}$$

Note that since the state at each time step t must go someplace at $t+1$, we have the constraint that $\sum_{l=1}^{N} a_{il}(t) = 1$. In this chapter, we ignore the case that the LED leaves the field of view; however, this case can be accomodated by augmenting the matrix with an additional row and column.

Eqn. (4.11) shows the scheme of computing $V_{t,j}$ recursively. It is initialized by equation $V_{1,j} = \boldsymbol{p}(\boldsymbol{z}(1) \mid j) \cdot \boldsymbol{p}_0(j)$ where $\boldsymbol{p}_0(j)$ is the prior probability of state j. To recover the state path, define the pointer matrix

$$\boldsymbol{Ptr} = [\boldsymbol{Ptr}(j,t)] \in \mathbb{R}^{N \times K}. \tag{4.13}$$

Its total number of rows and columns represent the state space and the time steps, respectively. The value of an entry in $\boldsymbol{Ptr}$ at j-th row and t-th column represents the the state value at time step $t-1$ that belongs to the most probable state sequence ended in state j at time step t. This can be shown using the following equation

$$\boldsymbol{Ptr}(j,t) = \arg\max_{i} \{a_{ij}(t-1) \cdot V_{t-1,i}\}. \tag{4.14}$$

After constructing this matrix iteratively until time step K according to eqn. (4.11) and (4.14) Then the most probable state path through time step t can be retrieved by

$$\boldsymbol{x}_t = \arg\max_{j \in \boldsymbol{X}}(V_{t,j}) \quad \text{and} \quad \boldsymbol{x}_{t-1} = \boldsymbol{Ptr}(\boldsymbol{x}_t, t). \tag{4.15}$$

4.3.2 Probabilistic Model

The measurement set $\boldsymbol{Z}(t)$ contains two types of information: the total number of pixels m_t that are above threshold and their positions $\boldsymbol{\mathcal{Z}}(t)$. That is:

$$\boldsymbol{Z}(t) = \{\boldsymbol{\mathcal{Z}}(t), m_t\}.$$

The probability of a measurement set $\boldsymbol{Z}(t)$ conditioned on the state $\boldsymbol{x}(t) = j$ can be decomposed as

$$\begin{aligned} & \boldsymbol{p}\left(\boldsymbol{Z}(t) \mid \boldsymbol{x}(t) = j\right) \\ = \; & \boldsymbol{p}\left(\boldsymbol{\mathcal{Z}}(t), m_t \mid \boldsymbol{x}(t) = j\right) \\ = \; & \boldsymbol{p}\left(\boldsymbol{\mathcal{Z}}(t) \mid m_t, \boldsymbol{x}(t) = j\right) \boldsymbol{p}\left(m_t \mid \boldsymbol{x}(t) = j\right). \end{aligned} \tag{4.16}$$

The second term in eqn. (4.16) is the probability of there being m_t detections in $Z(t)$, which does not depend on the LED pixel position j. Due to this fact, this term will not affect the results achieved by the Viterbi algorithm, so only the first term need to be considered when evaluating the measurement probability.

To compute the first term in eqn. (4.16), let the symbol $\theta_i(t)$ represent the hypothesis that $z_i(t)$ is the measurement of the LED for $i = 1, \cdots, m_t$ and the symbol $\theta_0(t)$ represent the hypothesis that the LED is not detected. Using the total probability theorem, the first term in the right hand side of eqn. (4.16) is decomposed as

$$\boldsymbol{p}\left(\boldsymbol{\mathcal{Z}}(t) \mid m_t, j\right) = \sum_{i=0}^{m_t} p\left(\boldsymbol{\mathcal{Z}}(t), \theta_i(t) \mid m_t, j\right). \tag{4.17}$$

For each term in the right hand side of eqn. (4.17), when $i \neq 0$ and $j \neq z_i(t)$ this (physically) means that the LED is projected on pixel j, but that the measured LED position is not pixel j; due to this contradiction, this probability has value 0. Eqn. (4.17) can therefore be simplified to

$$\boldsymbol{p}\left(\boldsymbol{\mathcal{Z}}(t) \mid m_t, j\right) = \begin{cases} \boldsymbol{p}(\boldsymbol{\mathcal{Z}}(t), \theta_0(t) \mid m_t, j), & j \notin \boldsymbol{Z}(t) \\ \sum\limits_{l \in \{0,i\}} \boldsymbol{p}(\boldsymbol{\mathcal{Z}}(t), \theta_l(t) \mid m_t, j), & j = z_i(t) \end{cases} \tag{4.18}$$

The notation $j \notin \boldsymbol{Z}(t)$ represents that there is no element in the measurement set $\boldsymbol{Z}(t)$ equaling j.

The right hand side of eqn. (4.18) is further decomposed as

$$\boldsymbol{p}\left(\boldsymbol{\mathcal{Z}}(t), \theta_i(t) \mid m_t, j\right) = \boldsymbol{p}\left(\boldsymbol{\mathcal{Z}}(t) \mid \theta_i(t), m_t, j\right) \boldsymbol{p}\left(\theta_i(t) \mid m_t, j\right). \tag{4.19}$$

The first term is the probability of the measurement position distribution given the hypothesis $\theta_i(t)$, state and number of measurements. It is reasonable to assume the positions of measurements due to clutter or noise in set $\boldsymbol{Z}(t)$ are independently and uniformly distributed in the detection space. When $i = 0$ in the first term of eqn. (4.19), none of the measurements in set $\boldsymbol{\mathcal{Z}}(t)$ is originated from the LED, so it is computed according to

$$\boldsymbol{p}\left(\boldsymbol{\mathcal{Z}}(t) \mid \theta_0(t), m_t, j\right) = \prod_{j=1}^{m_t} \frac{1}{N} = \frac{1}{N^{m_t}}. \tag{4.20}$$

When $i \neq 0$, one measurement say it $z_i(t)$ in the set $\boldsymbol{\mathcal{Z}}(t)$ is originated from the LED and the others are due to the noise, so it is decomposed as

$$\boldsymbol{p}\left(\boldsymbol{\mathcal{Z}}(t) \mid \theta_i(t), m_t, j\right) = \left(\prod_{j \neq i} \frac{1}{N}\right) \cdot \boldsymbol{p}\left(z_i(t) \mid \theta_i(t), m_t, j\right) = \frac{1}{N^{m_t - 1}}. \tag{4.21}$$

Combine these two cases together, the first term in eqn. (4.19) is computed by

$$\boldsymbol{p}\left(\boldsymbol{\mathcal{Z}}(t) \mid \theta_i(t), m_t, j\right) = \begin{cases} \frac{1}{N^{m_t}}, & i = 0 \\ \frac{1}{N^{m_t - 1}}, & i \neq 0, j = z_i(t) \end{cases} \tag{4.22}$$

The second term in eqn. (4.19) is the prior hypothesis probability given the number of measurements and LED projection position are known. When $i = 0$, this means the LED is not detected, so the LED must be "off". When $i \neq 0$, this means the LED is detected, so the LED must be "on". The second term can be analyzed as

$$\begin{aligned} \boldsymbol{p}\left(\theta_i(t) \mid m_t, j\right) &= \frac{\boldsymbol{p}\left(\theta_i(t), m_t \mid j\right)}{\boldsymbol{p}\left(m_t \mid j\right)} \\ &= \begin{cases} \frac{1}{c'}(1 - P_{on})\boldsymbol{\mu}_F(m_t), & i = 0 \\ \frac{1}{c'}\frac{1}{m_t} P_{on} \boldsymbol{\mu}_F(m_t - 1). & i \neq 0 \end{cases} \end{aligned} \tag{4.23}$$

where normalized factor $c' = \boldsymbol{p}(m_t \mid j)$ is independent of the state j, P_{on} is the probability that the LED is on, and $\boldsymbol{\mu}_F(m)$ is the PMF of the number of clutter measurements. Here we assume μ_F is a diffuse pmf [11], that is, all number of clutter measurements are equally likely.

Substituting eqn. (4.17)-(4.23) into eqn. (4.16), after combining all constants into c' and significant simplification, the final expression is

$$\boldsymbol{p}(\boldsymbol{Z}(t) \mid j) = \begin{cases} \frac{1}{c'}\frac{1-P_{on}}{N}, & j \notin \boldsymbol{Z}(t) \\ \frac{1}{c'}\left(\frac{1-P_{on}}{N} + \frac{P_{on}}{m_t}\right). & j \in \boldsymbol{Z}(t) \end{cases} \tag{4.24}$$

4.3.3 Effect of LED Image Width

Due to optical distortion and the non-zero LED size, both the LED and clutter measurements may occupy more than one pixel. Actual LED projections will be composed of several connected pixels whose intensity values may be greater than the threshold. The elements in the measurement set $\boldsymbol{Z}(t)$ are no longer independent, since some of them may belong to a same measurement. The assumptions made to derive eqns. (4.22) and (4.23) are no longer valid. For example, the measurement set shown in Fig. 4.2 is

$$\boldsymbol{Z}(t) = \{44 \ldots 48, 231 \ldots 235, 241 \ldots 243\}, \tag{4.25}$$

where there are three measurement clusters that contain a total of 13 pixels. The measurements in each cluster are not independent. It is important to consider how the violation of these assumptions affects the performance of the algorithm.

In Sec. 4.2.3, the state is defined as the true LED projection location on the linear array, which causes no ambiguity when the LED occupies only one pixel. When considering the nonzero LED intensity measurement width, the state is redefined to be the center position of the LED projection on the linear array. This definition will not

contradict with the one in Sec. 4.2.3 when the optics are perfect. This new definition does not influence the state transition matrix, to be defined in eqn. (4.29).

The measurement set $\boldsymbol{Z}(t)$ still has the definition as in eqn. (4.5). Its elements are the pixel locations that have above threshold intensity values. Suppose that the m_t pixel locations in $\boldsymbol{Z}(t)$ form m'_t clusters with the jth cluster containing $n_{t,j}$ pixels, so that $\sum_{j=1}^{m'_t} n_{t,j} = m_t$. For a cluster composed of several pixels, each pixel of the cluster can be viewed as the center of the LED projection. The definition of hypothesis $\theta_i(t)$ in Sec. 4.3.2 is extended to be $z_i(t)$ is the center of the LED projection.

In eqn. (4.22), since the total number of measurements is m'_t, there will be only m'_t independent terms. Eqn. (4.22) is modified as

$$\boldsymbol{p}\left(\boldsymbol{\mathcal{Z}}(t) \mid \theta_i(t), m_t, j\right) = \begin{cases} \frac{1}{N^{m'_t}}, & i = 0 \\ \frac{1}{N^{m'_t - 1}}, & i \neq 0, j = z_i(t). \end{cases} \tag{4.26}$$

Similarly, eqn. (4.23) is modified as

$$\boldsymbol{p}\left(\theta_i(t) \mid m_t, j\right) = \begin{cases} \frac{1}{c''}(1 - P_{on})\boldsymbol{\mu}_F(m'_t), & i = 0, \\ \frac{1}{c''}\frac{1}{m_t}P_{on}\boldsymbol{\mu}_F(m'_t - 1), & i \neq 0. \end{cases} \tag{4.27}$$

Notice that the term $1/m_t$ in the eqn. (4.27) is not replaced by $1/m'_t$ when $i \neq 0$ The reason is there are still m_t (not m'_t) candidates for the center of an LED that is "on".

Combining all the constants into c'', eqn. (4.24) becomes

$$\boldsymbol{p}\left(\boldsymbol{Z}(t) \mid j\right) = \begin{cases} \frac{1}{c''}\frac{1-P_{on}}{N}, & j \notin \boldsymbol{Z}(t), \\ \frac{1}{c''}\left(\frac{1-P_{on}}{N} + \frac{P_{on}}{m_t}\right), & j \in \boldsymbol{Z}(t). \end{cases} \tag{4.28}$$

Notice that the only difference between eqn. (4.28) and eqn. (4.24) is the constant term, which does not influence the most likely path final result. Table 4.2 summarizes the LED path recovery algorithm.

Table 4.2: LED path recovery algorithm

Input: linear array measurements for $t \in [0, K]$

state prior probability $\boldsymbol{p}_0 = \mathcal{N}(\boldsymbol{x}_0, \boldsymbol{S}_0)$

Output: the most probable state sequence $\boldsymbol{X}^T$

1. Find the detection range defined by the state prior $\boldsymbol{p}_0$

   $\left[\lfloor \boldsymbol{x}_0 - \rho * \sqrt{\boldsymbol{S}_0} \rfloor, \lceil \boldsymbol{x}_0 + \rho * \sqrt{\boldsymbol{S}_0} \rceil\right]$

2. Extract the measurement sets defined in eqn. (4.5) for $t \in [0, K]$

3. Initialize $V_{1,j}$ in eqn. (4.11)

   $V_{1,j} = \boldsymbol{p}(\boldsymbol{z}(1) \mid j) \cdot \boldsymbol{p}_0(j)$

4. Compute $V_{t,j}$ using the transition matrix $\boldsymbol{A}(t-1)$ and measurement model defined in eqns. (4.29) and (4.16)

   $V_{t,j} = \boldsymbol{p}(\boldsymbol{z}(t) \mid j) \max_i \{a_{ij}(t-1) \cdot V_{t-1,i}\}$

5. Assign values to the elements in the t-th column of matrix $\boldsymbol{Ptr}$. The value of $\boldsymbol{Ptr}(j,t)$ is the state i that used to compute $V_{t,j}$.

   $\boldsymbol{Ptr}(j,t) = \arg\max_i \{a_{ij}(t-1) \cdot V_{t-1,i}\}$

6. Retrieve the most likely state sequence

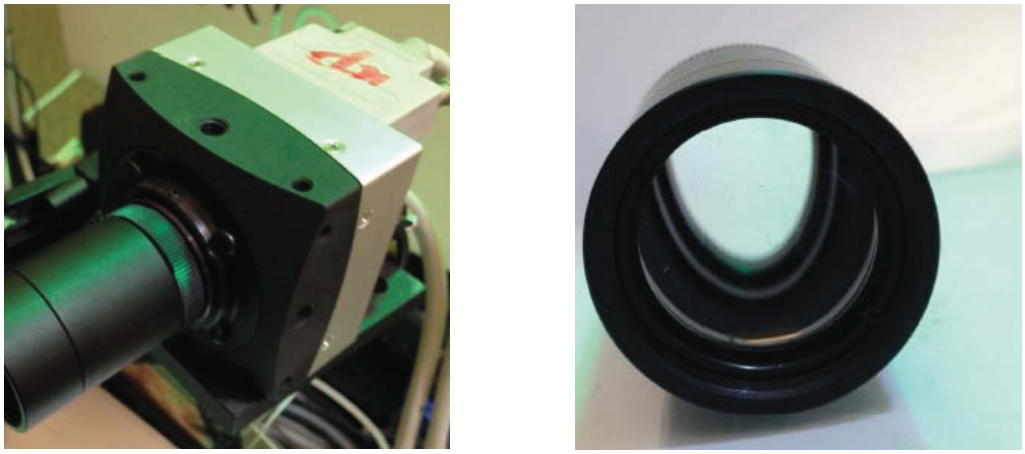
   $\boldsymbol{x}_t = \arg\max_{j \in \boldsymbol{X}}(V_{t,j})$

   $\boldsymbol{x}_{t-1} = \boldsymbol{Ptr}(\boldsymbol{x}_t, t)$

4.4 Results

4.4.1 Experimental Setup

The experiments reported in this section produce linear array measurements by reading a single row of pixel data from an EoSens CL camera modified to with a cylindrical lens, see Fig. 4.4. The maximum frame rate when sampling one row is $120KHz$. In the following experiments, the LED's data rate is modulated as $800Hz$, and the linear array frame is read at $1600Hz$.



(a) EoSens CL camera.

(b) Cylindrical lens.

Figure 4.4: EoSens CL camera and cylindrical lens.

4.4.2 State Transition Model

The state transition model $\boldsymbol{p}(\boldsymbol{x}(t) = j \mid \boldsymbol{x}(t-1) = i)$ in eqn. (4.11) depends on the motion $\boldsymbol{u}(t-1)$, rover state $\boldsymbol{\chi}(t-1)$, LED position $\boldsymbol{F}_l$, and LED projected state $\boldsymbol{x}(t-1)$. The full nonlinear analysis would be very complicated, resulting in very hard implementation. The assumption in eqn. (4.3) that the frame rate is high relative to the rover's motion bandwidth can be trnslated into a model of the state transition by

assuming that the LED state has probability $\alpha \in (0,1)$ of staying in its current state and probability $(1-\alpha)$ of moving to an adjacent state in each sample instant. Ignoring the issue of LED's leaving the field-of-view, this means that the state transition matrix of eqn. (4.12) is tri-diagonal:

$$\boldsymbol{A} = \begin{bmatrix} \alpha & 1-\alpha & 0 & \cdots & 0 & 0 \\ \frac{1-\alpha}{2} & \alpha & \frac{1-\alpha}{2} & \cdots & 0 & 0 \\ 0 & \frac{1-\alpha}{2} & \alpha & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \alpha & \frac{1-\alpha}{2} \\ 0 & 0 & 0 & \cdots & 1-\alpha & \alpha \end{bmatrix}. \tag{4.29}$$

This matrix representation also assumes that the LED has identical probability moving to either side. The state transition process defined by matrix $\boldsymbol{A}$ is illustrated by Fig. 4.5. The algorithm extends directly to state transition matrices that represent more general motion models. In this implementation, the state transition probability α is set as $1/3$.

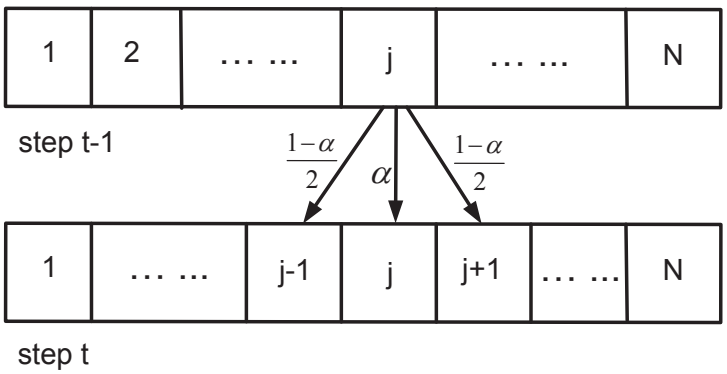


Figure 4.5: State transition process defined in matrix $\boldsymbol{A}$ (4.29).

4.4.3 Stationary

The left image of Fig. 4.6 shows a series of stationary linear array measurements as a single image. The linear array of greyscale pixel readings from any one time instant is represented as a column of the image. This figure is formed by stacking a sequence of linear array measurements together each as one column. The time sequence of array measurements forms a matrix, shown as an image, with time increasing to the right along the horizontal axis. Because the linear array is stationary, the communicating LED always projects to essentially the same pixel array elements. The number of columns in the image matrix is 300, corresponding to a time interval of $0.1875s$. The right image in Fig. 4.6 shows the binary image after thresholding. Notice the on-off alternation of the LED and the noise in the figure.

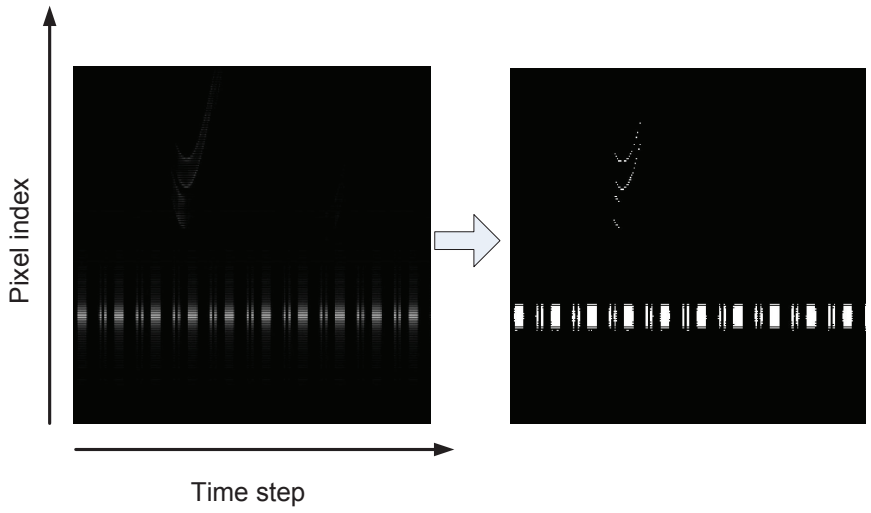


Figure 4.6: Stationary platform sequence of raw (left) and thresholded (right) linear array measurement data represented as an image.

The Viterbi algorithm in Table 4.2 is implemented using the thresholded data to calculate the probability of the most likely LED path ending at each pixel. The results $V_{t,j}$ are shown in the colored map at left image of Fig. 4.7, where brightness

(e.g., black is unlikely, yellow is likely) increases for higher values of $V_{t,j}$. In the last (rightmost) scan, the pixel with largest probability is chosen as the end position of the most likely LED path. The most likely path is recovered backwards through time using eqn. (4.15). The most likely path is graphed as the green line in the right image of Fig. 4.7. The fact that this path passes through the LED measurement path at each time step demonstrates the accuracy of extracted path.

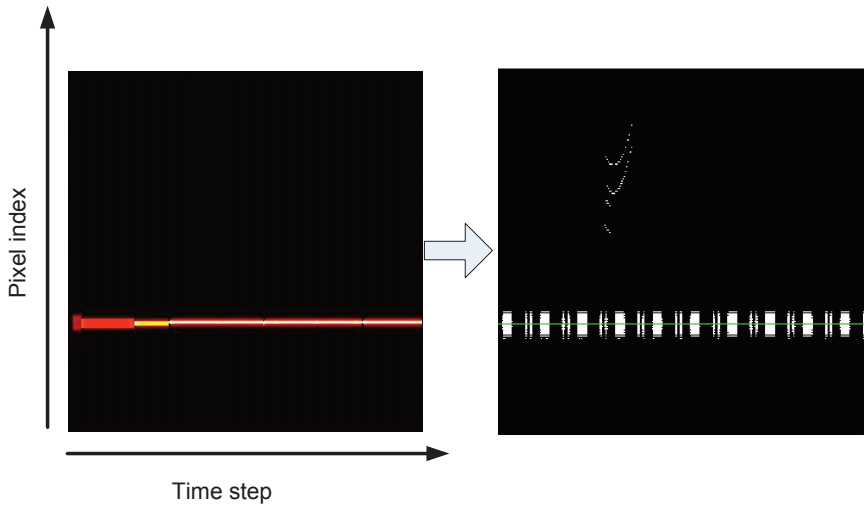


Figure 4.7: Stationary probability image and recovered LED path (green line).

4.4.4 Moving

When the linear array is moving, the LED projection changes its pixel location from one time to the next as shown in the upper left image of Fig. 4.8. The same pixel thresholding process is shown by the two upper images of Fig. 4.8. The results $V_{t,j}$ and the recovered LED path are shown by the two bottom images. Even thought the LED's location in the linear is not fixed during the data recovery process, the true LED path is accurately recovered.

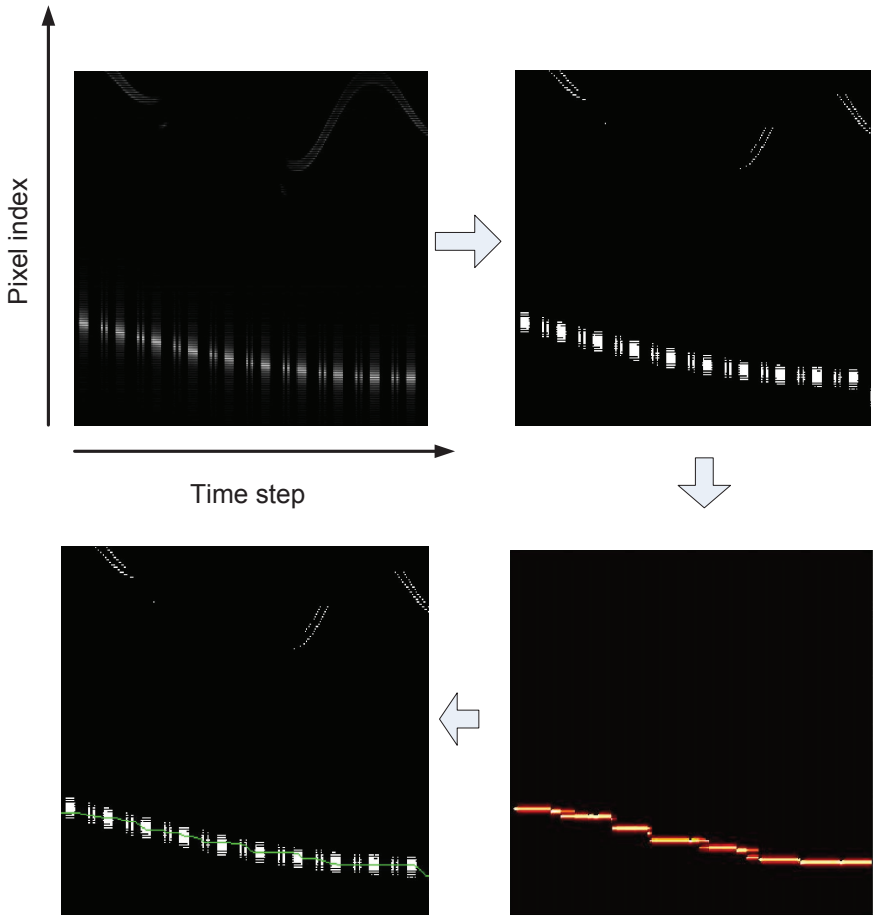


Figure 4.8: Moving platform experimental results.

The LED signal is modulated to send a data sequence. The data sequence was extracted without error after the path was recovered. The following Fig. 4.9 illustrates the recovered data after the LED path is found.

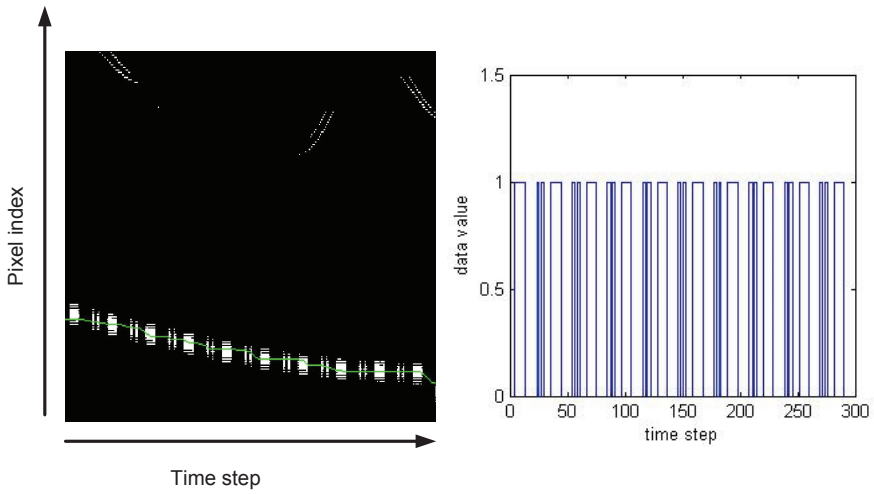


Figure 4.9: The LED data based on the recovered LED path.

Chapter 5

LED Data Recovery in a Moving Sensor

5.1 Introduction

Chapter 4 introduces a Viterbi-based algorithm to recover the LED data in the linear array when the linear array is stationary or moving with a bandwidth that is low relative to the imaging rate. This algorithm evaluates the probability of LED projections in a sequence of linear array measurements, and keeps the most probable projection sequence ended in each pixel. Based on these kept projection sequences, the most probable LED projection positions in the linear array sequence can be recovered backwards. A problem of this algorithm is that it is hard to be applied to the two-dimensional photo-detector such as camera. The enumeration and connection between the pixels in the two-dimensional image is much more complicated than that in the linear array, so the state transition matrix for the two-dimensional measurements may not be diagonal any more and hard to define. Another problem of this Viterbi-based algorithm is the moving bandwidth requirement. Even though the LED switching frequency can

be modulated up to hundreds of megahertz, sometimes we still prefer to utilize a low rate sensor (i.e., 30Hz) such as a webcam or a cellphone camera to receive the data using undersampling methods [68]. In such a case, the assumption for the algorithm in Chapter 4 is no longer valid. Due to the above facts, more general and robust algorithm should be developed to recover the LED data in the photo-detector.

To solve this problem, we follow the idea in Chapter 4 to consider the measurements in a sequence of images jointly. The measurements of the same object (LED) at multiple time instances should be matched together to recover the data sequence. For image measurements, the features in different images can be tried matched by comparing their descriptors. Multiple existing corner detectors (Moravec [55], Forstner [24], Harris [30], Shi-Tomasi [72], and FAST [70]) or blob detectors (SIFT [52], SURF [12], and CENSURE [2]) can be used to match the projections of the same feature in different two-dimensional images. These detectors keep the description of each feature and these descriptions can be compared to determine whether multiple features are originated from the same object. However, there is still probabilities that incorrect pairs are matched or multiple measurements in the same image are matched to the same feature. This incorrect matches happen more frequently when measurements in a relative long sequence of images (hundreds of images) are to be matched instead of only two images, like the case of recovering LED data. In typical vision aided navigation systems, usually only measurements in two consecutive image are to be matched. If multiple measurements in another image are matched to one measurement in the current image, these measurements would be ignored to prevent incorrect matches. If we follow the same strategy to extract the LED data, there would be a low probability to obtain a correct data sequence, so that both the navigation and communication will become unreliable. Moreover, the existing feature descriptors can not be applied to the linear array mea-

surement, so that they can not be used to match the measurements in different linear array scans. All the analysis above indicate that accurately and reliably recovering the LED data sequence merely by the image processing techniques is not enough.

5.1.1 Multiple Hypothesis Data Association

Multiple detections existing within the search area results in a multiple hypothesis data association problem [21]. This data association problem could be solved by choosing the measurement that is closest to the predicted measurement position in Mahalanobis sense, see [53]. Among other issues, this method does not work in this application, as it fails in the presence of noise or clutter when the LED is off. Another method that accounts for the null hypothesis (the LED is off) can be implemented using the probabilistic data association filter, as in [10]. Though this method works in theory, there is a very low probability to obtain a correct data sequence when it is long.

More sophistic methods can be developed using MHT methods, see [67, 9, 47]. MHT uses a deferred-decision approach in which they maintain the complete set of possible data associations within a sliding window, putting off hard decisions as long as possible, see [26]. Decisions are made by evaluating the probabilities of each sequence of data association hypotheses. Since the number of possible sequences increases exponentially, it is expensive to keep all of them. It is also inefficient to compute the probabilities of all possible sequences and then discarding most of them. To solve these problems, an efficient Hypothesis Oriented MHT (HOMHT) was firstly proposed in [17]. The basic idea of this method is to keep only the q best sequences, discarding the other sequences that have low probability.

All the above data recovery techniques require the information from the navigation system. The motion measurements provide additional constraints for matching

objects at different time instances. Accurate navigation enables both efficiency of which LED ID's to expect for and where each LED is expected to be in the image. For both the camera and linear array, the LED's unique ID would facilitate the task of feature association, which is a limiting factor to the use of more generic non-LED features. Measurement confirmation is straightforward using the extracted and expected ID. From all the analysis above, we can again conclude that the communication and navigation applications are synergistic.

5.2 Problem Formulation

As discussed in Sec. 4.2, given a time interval $\lambda \in [t_s, t]$, the projected position of the l-th LED $z_l(\lambda)$ defines a trajectory across the photo-detector. Detection of the projected LED position depends on the LED "on" or "off" status, as well as environmental conditions and interference from other light sources. The accuracy of each LED's recovered on-off status is highly dependent on the accuracy of the data association at each time step. Therefore, the data recovery and trajectory estimation problems are coupled. Improvement in the solution of either problem enhances the solution of the other.

The purpose of this chapter is to develop an algorithm that simultaneously estimates the most likely data association sequences and rover state over each time interval $\lambda \in [t_s, t]$. Without loss of generality, we define $t_s = 0$, and the pixel array is sampled at times $t_k = kT \in [t_s, t]$ for $k = 0, 1, \cdots, K$. After the data association step, and the validity of the LED data and therefore the association is established, the validated sequence of measurements $\tilde{z}_l(t_k)$ will be referred to *aiding measurements* and used to correct the estimated state trajectory.

5.2.1 Predicted Measurement Region

Given the state estimate at time step k, it is straightforward to compute both the predicted measurement position of the l-th LED $\hat{\boldsymbol{z}}_l(k)$ and its error covariance $\boldsymbol{S}_l(k)$ according to eqn. (3.9) and (3.12). The quantities $\hat{\boldsymbol{z}}_l(k)$ and $\boldsymbol{S}_l(k)$ define a prior distribution for the LED trajectory that can focus the algorithm. A region in the measurement space for the l-th LED can be defined as

$$\begin{aligned} V_{l,\gamma}(k) &\triangleq \{\boldsymbol{z} : (\boldsymbol{z} - \hat{\boldsymbol{z}}_l(k))^\top \boldsymbol{S}_l^{-1}(k)(\boldsymbol{z} - \hat{\boldsymbol{z}}_l(k)) \leq \gamma\} \\ &= \{\boldsymbol{z} : r_l(k)^\top \boldsymbol{S}_l^{-1}(k)\boldsymbol{r}_l(k) \leq \gamma\}, \end{aligned} \tag{5.1}$$

where $\boldsymbol{r}_l(k) = \boldsymbol{z}(k) - \hat{\boldsymbol{z}}_l(k)$ is the residual and γ is a parameter that determines the probability that the real measurement falls in $V_{l,\gamma}(k)$. For the camera measurement, the region $V_{l,\gamma}(k)$ represents the interior of an ellipse at each time k. For the linear array measurement, the region $V_{l,\gamma}(k)$ represents the interior of a segment at each time k, respectively. In the following, without loss of generality, we only consider a single LED in its predicted region $V_{l,\gamma}$; therefore, we drop the l subscript. The parameter γ is selected using a χ^2-distribution table.

5.2.2 Data Association Hypothesis

Let $m_k \geq 0$ denote the number of detected LED like features in the predicted region $V_\gamma(k)$ at time k, where $z_j(k)$ for $j \in [0, m_k]$ denotes projected location of the j-th measurement hypothesis. Define the measurement set at time step k as

$$\boldsymbol{Z}(k) \triangleq \{z_j(k)\}_{j=1}^{m_k}. \tag{5.2}$$

Note that at most one of them originated from the LED. The others are false detections due to noise, interference, or clutter. Since each or none of the measurements in this

set could be originated from the LED, the number of data association hypotheses is $(m_k + 1)$.

Suppose there are total of K time steps, the state estimates $\{\hat{\boldsymbol{x}}(k)\}_{k=1}^{K}$ and their error covariance matrices $\{\boldsymbol{P}_k\}_{k=1}^{K}$ are calculated using eqns. (3.2) and (3.7). Then the predicted region at each time step is calculated by eqns. (3.9, 3.12, 5.1), and the measurements that fall into V_γ are extracted. Define

$$\boldsymbol{Z}^k \triangleq \{\boldsymbol{Z}(l)\}_{l=1}^{k} and \boldsymbol{U}^k \triangleq \{\boldsymbol{u}(l)\}_{l=1}^{k} \tag{5.3}$$

as the set of measurements and inputs up to time step k, respectively. Since there are a total of K time steps, the number of joint data association hypotheses is

$$L_K = \prod_{k=1}^{K} (m_k + 1). \tag{5.4}$$

Each data association sequence defines a LED trajectory across the detector, which in turn decodes a sequence of detection and nondetection events into a bit sequence containing an ID, checksum, data, etc. The total number of time steps K in the algorithm is a parameter selected by the designer. It choice depends on the length (number of bits) of each LED data package and the sampling rate of photo-detector. If the length of LED data package is l_d and the sampling rate is n_s times of the data rate, the total number of time steps K is at least $(n_s\, l_d)$. Only after the LED status is recovered at each time instant in the sequence of K measurements, can the LED data be extracted. Fig. 4.1(b) a sequence of linear array measurements. Both the variation of the LED projected position due to rover motion and the on-off status of the LED due to the data sequence can be clearly seen.

Let $\theta^{k,\ell}$ represent a specific list of joint data association hypotheses up to time step k. The superscript $\ell \in [1, L_k]$ is the numeration index for the hypothesis sequence.

Each hypothesis $\theta^{k,\ell}$ has the form

$$\theta^{k,\ell} = \{j_1, j_2, \cdots, j_i, \cdots, j_k\}, \tag{5.5}$$

where $j_i \in [0, m_i]$. When $\theta^{k,\ell}(i) = j > 0$, it represents that at time step i the j-th measurement originated from the LED. When $\theta^{k,\ell}(i) = 0$ it represents that at time step i none of the measurements has originated from the LED. We say the joint hypothesis $\theta^{k,\ell}$ is an extension of $\theta^{k-1,s}$, if

$$\theta^{k,\ell} = \{\theta^{k-1,s}, j_k\},$$

where $\theta^{k,\ell}(k) = j_k \in [0, m_k]$. In this nomenclature, $\theta^{k-1,s}$ is the *parent hypothesis* of $\theta^{k,\ell}$. Herein, the indices ℓ and s in the superscript of θ are not necessarily identical, since the two hypotheses are enumerated independently at different time steps. Note that each joint data association hypothesis corresponds to a potentially distinct posterior vehicle trajectory.

5.2.3 Technical Problem Statement

Given the measurement set $\boldsymbol{Z}^K$, we would like to find the most probable joint data association hypotheses (i.e. q-best hypotheses $\{\theta^{K,\ell}\}_{\ell=1}^{q}$). After these hypotheses are found, their corresponding ID sequences are straightforward to recover. By comparing each of the recovered ID's with the predicted ID, the most probable data association hypothesis with correct ID is found and used both for data extraction and the navigation state update. The advantage of finding the q-best hypotheses instead of only the most probable one is that having several candidate hypotheses yields a higher probability to recover the correct ID.

The problem is stated as: Given measurements $\boldsymbol{Z}^K$ and navigation system inputs $\boldsymbol{U}^K$, find the q-best joint data association hypotheses $\{\theta^{K,\ell}\}_{\ell=1}^{q}$ among the total

number of L_K hypotheses, where q-best means the q hypotheses that maximize the hypothesis posterior probability.

5.3 MHT-based LED Data Recovery Method

This section evaluates the posterior probability mass function (PMF) of a joint data association hypothesis up to time step k. The analysis follows the method in [67] while adding details to clarify the measurement quantity information. Here we make an assumption that the measurements due to clutter are uniformly distributed in the predicted region V_γ.

5.3.1 Hypothesis Probability

The joint probability of an association hypothesis up to time step $k \in [1, K]$ can be decomposed as

$$\begin{aligned} & p\left(\theta^{k,\ell} \mid \boldsymbol{Z}^k, \boldsymbol{U}^{k-1}\right) \\ = \; & \frac{1}{c} p(\boldsymbol{Z}(k) \mid \theta^{k,\ell}, \boldsymbol{Z}^{k-1}, \boldsymbol{U}^{k-1})\, p(\theta^{k,\ell} \mid \boldsymbol{Z}^{k-1}, \boldsymbol{U}^{k-1}) \end{aligned} \tag{5.6}$$

where the normalization factor $c = p\left(\boldsymbol{Z}(k) \mid \boldsymbol{Z}^{k-1}, \boldsymbol{U}^{k-1}\right)$ is independent of the data association hypothesis $\theta^{k,\ell}$. The decomposition on the right hand side of eqn. (5.6) is based on Bayes Rule. The last term on the right hand side can be further simplified by application of the general multiplication rule of conditional probability
(i.e., $p(A, B) = p(A \mid B)p(B)$):

$$\begin{aligned} & p(\theta^{k,\ell} \mid \boldsymbol{Z}^{k-1}, \boldsymbol{U}^{k-1}) \\ = \; & p(\theta^{k,\ell}(k), \theta^{k-1,s} \mid \boldsymbol{Z}^{k-1}, \boldsymbol{U}^{k-1}) \\ = \; & p(\theta^{k,\ell}(k) \mid \theta^{k-1,s}, \boldsymbol{Z}^{k-1}, \boldsymbol{U}^{k-1}) p(\theta^{k-1,s} \mid \boldsymbol{Z}^{k-1}, \boldsymbol{U}^{k-1}). \end{aligned}$$

Combining the last equation above with eqn. (5.6) yields the final desired result:

$$p\left(\theta^{k,\ell} \mid \boldsymbol{Z}^k, \boldsymbol{U}^{k-1}\right) = \frac{1}{c} p(\boldsymbol{Z}(k) \mid \theta^{k,\ell}, \boldsymbol{Z}^{k-1}, \boldsymbol{U}^{k-1}) \tag{5.7}$$

$$p(\theta^{k,\ell}(k) \mid \theta^{k-1,s}, \boldsymbol{Z}^{k-1}, \boldsymbol{U}^{k-1}) p\left(\theta^{k-1,s} \mid \boldsymbol{Z}^{k-1}, \boldsymbol{U}^{k-2}\right)$$

When analyzing this joint probability, we have to take notice of the measurement quantity information $m^k = \{m_i\}_{i=1}^k$ which is not explicitly represented in eqn. (5.7). In [8], this quantity information m^k is considered to be part of the measurement set $\boldsymbol{Z}^k$, while in [67] it is viewed to be in the hypothesis set $\theta^{k,\ell}$. The results given by the two approaches are identical except for a scale factor $p\left(m^k\right)$ that is independent of $\theta^{k,\ell}$, which can be seen from

$$p\left(\theta^{k,\ell}, m^k \mid \boldsymbol{Z}^k, \boldsymbol{U}^{k-1}\right) \quad = \quad p\left(\theta^{k,\ell} \mid \boldsymbol{Z}^k, m^k, \boldsymbol{U}^{k-1}\right) p\left(m^k\right).$$

We use the later approach in the following analysis, so that $\boldsymbol{Z}^k$ and $\boldsymbol{Z}(k)$ will only contain the measurement position information.

The first term in the right hand side of eqn. (5.7) is the joint prior probability of the current position measurements given the data association hypotheses, measurements and inputs of all the previous time steps. The measurements in $Z(k)$ are independent because they come from different sources, potentially one from the LED and the others from clutter, etc. Because the different measurements in set $Z(k)$ are independent, this term decomposes as

$$\begin{aligned} & p(Z(k) \mid \theta^{k,\ell}, Z^{k-1}, U^{k-1}) \\ = & \prod_{j=1}^{m_k} p(z_j(k) \mid \theta^{k,\ell}, Z^{k-1}, U^{k-1}) = \prod_{j=1}^{m_k} f(j). \end{aligned} \tag{5.8}$$

Note that the measurement quantity m_k is considerred to be known when evaluating the probability of measurement set $Z(k)$ in eqn. (5.8). The measurements due to clutter or false alarms are assumed to be uniformly distributed in the predicted region, while the

one from the LED is assumed to be corrupted by Gaussian noise. Therefore, we have

$$f(j) \quad = \quad \begin{cases} \frac{1}{V_k} & \text{for clutter,} \\ \mathcal{N}(z_j(k); \hat{z}^s(k), S^s(k)) & \text{for the LED,} \end{cases} \tag{5.9}$$

where V_k is the volume (area) of the predicted region $V_{\gamma(k)}$, and $\mathcal{N}(z_j(k); \hat{z}^s(k), S^s(k))$ is the Gaussian distribution evaluated at $z_j(k)$ with expectation $\hat{z}^s(k)$ and covariance $S^s(k)$.

Because $\theta^{k,\ell}$ is an extension of $\theta^{k,\ell-1}$ as defined by eqn. (5.2.2), the conditioning variables allow the distributions in eqn. (5.8) to be written as

$$\prod_{j=1}^{m_k} p(z_j(k) \mid \theta^{k-1,s}, j_k, Z^{k-1}, U^{k-1}).$$

The variable j_k and the index j determine the form of f_j as defined in eqn. (5.9). The variables $\theta^{k-1,s}, Z^{k-1}, U^{k-1}$ are the prior information used to compute $\hat{\boldsymbol{x}}^s(k-1)$, $\hat{z}^s(k)$ and $S^s(k)$, where the superscript s indicates its corresponding hypothesis $\theta^{k-1,s}$, which are computed based on eqns. (3.2)-(3.12) with measurement updates implemented by an optimal estimator (e.g., Kalman filter). More details of computing $\hat{z}^s(k)$ and $S^s(k)$ for a specific example are discussed in Sec. 5.4.

Substituting eqn. (5.9) into (5.8), the first term of (5.7) is

$$\begin{aligned} & p(Z(k) \mid \theta^{k,\ell}, Z^{k-1}, U^{k-1}) \\ = & \begin{cases} \frac{1}{V_k^{m_k}} & \text{for } \theta^{k,\ell}(k) = 0 \\ \frac{\mathcal{N}(z_j(k); \hat{z}^s(k), S^s(k))}{V_k^{m_k - 1}} & \text{for } \theta^{k,\ell}(k) \neq 0 \end{cases} \end{aligned} \tag{5.10}$$

The second term of eqn. (5.7) is the prior probability of the current association

hypothesis $\theta^{k,\ell}(k)$ conditioned on $\theta^{k-1,s}$, Z^{k-1}, and U^{k-1}. This term is computed as

$$
\begin{aligned}
& p(\theta^{k,\ell}(k) \mid \theta^{k-1,s}, Z^{k-1}, U^{k-1}) \\
= & \begin{cases} (1 - P_{on})\mu_{\lambda V_k}(m_k) & \text{for } \theta^{k,\ell}(k) = 0, \\ \\ \frac{1}{m_k} P_{on}\mu_{\lambda V_k}(m_k - 1) & \text{for } \theta^{k,\ell}(k) = 1 \cdots m_k, \end{cases}
\end{aligned} \tag{5.11}
$$

where P_{on} is the probability that the LED is on, and μ_F is the PMF of the number of false measurements due to clutter or noise. The existence of term $\mu_{\lambda V_k}$ is due to the measurement quantity information m_k in $\theta^{k,\ell}(k)$. When $\theta^{k,\ell}(k) = 0$, all the measurements in the predicted region are not from the LED, so the LED is off. We do not consider the case of the missing detections. When $\theta^{k,\ell}(k) \neq 0$, the hypothesis is that the LED is on and that it is associated with a specific measurement in the predicted region. Because the distribution is not conditioned on the current position measurement $Z(k)$, when evaluating this prior probability, each measurement has equal probability (i.e., $1/m_k$) to be the correct one.

The number of measurements due to clutter and false alarms is modeled as a Poisson process [67, 9], so its distribution function μ_F at time step k is

$$\mu_{\lambda V_k}(\phi) = \exp^{-\lambda V_k} \frac{(\lambda V_k)^{\phi}}{\phi!}. \tag{5.12}$$

The parameter λ represents the spatial density of noise measurements. The term λV_k is the expected number of noise measurements in the predicted region (i.e., $E\langle m_k \rangle = \lambda V_k$). In some applications, the parameter λ is unknown. A common solution [9] is to use the true number of measurements m_k to replace its expected value, λV_k, in eqn. (5.12). This results in the nonparametric model of $\boldsymbol{\mu}_{m_k}$. Note that in this special case it can be shown that $\mu_{m_k}(m_k) = \mu_{m_k}(m_k - 1)$.

Let $\beta^{k,\ell} = \boldsymbol{p}(\theta^{k,\ell} \mid \boldsymbol{Z}^k, \boldsymbol{U}^{k-1})$ denote the posterior PMF of $\theta^{k,\ell}$ in eqn. (5.7).

Substituting eqn. (5.10) - (5.12) into eqn. (5.7) and combining the constants into c', the probability is

$$\beta^{k,\ell} = \begin{cases} \frac{1}{c'}\lambda(1-P_{on})\beta^{k-1,s} & \text{for } \theta(k)=0 \\ \\ \frac{1}{c'}\mathcal{N}^{k-1,s}(\boldsymbol{z}_j(k))P_{on}\beta^{k-1,s} & \text{for } \theta(k)=j\neq 0, \end{cases} \tag{5.13}$$

where $\mathcal{N}^{k-1,s}(\boldsymbol{z}_j(k)) = \mathcal{N}(\boldsymbol{z}_j(k); \hat{\boldsymbol{z}}^s(k), \boldsymbol{S}^s(k))$, and $c' = c \cdot \boldsymbol{e}^{\lambda V_k} \cdot m_k! \cdot \lambda^{1-m_k}$. If using the nonparametric model of $\boldsymbol{\mu}_F$, the probability is

$$\beta^{k,\ell} = \begin{cases} \frac{1}{c''}\frac{m_k}{V_k}(1-P_{on})\beta^{k-1,s} & \text{for } \theta(k)=0 \\ \\ \frac{1}{c''}\mathcal{N}^{k-1,s}(\boldsymbol{z}_j(k))P_{on}\beta^{k-1,s} & \text{for } \theta(k)=j\neq 0, \end{cases} \tag{5.14}$$

where $c'' = c \cdot V_k^{m_k-1} \cdot m_k$. Eqn. (5.13) or (5.14) shows how to compute the probability of joint data association incrementally.

From eqn. (5.13), the marginal likelihood function for the hypothesis that the j-th measurement at time step k has originated from the LED is

$$\mathcal{L}_{s,j}(k) = \mathcal{N}^{k-1,s}(\boldsymbol{z}_j(k))P_{on}, \tag{5.15}$$

and the marginal likelihood function for the hypothesis that no measurement at time step k is from the LED is

$$\mathcal{L}_{s,0}(k) = \lambda(1-P_{on}). \tag{5.16}$$

Therefore, the probability that hypothesis $\theta(k)=j$ is the extension of joint hypothesis $\theta^{k-1,s}$ is

$$\boldsymbol{p}(\theta(k)=j \mid \boldsymbol{Z}^k, \theta^{k-1,s}, \boldsymbol{U}^{k-1}) = \frac{\mathcal{L}_{s,j}(k)}{\sum_{i=0}^{m_k}\mathcal{L}_{s,i}(k)}. \tag{5.17}$$

5.3.2 Hypothesis for Multiple LEDs

When there are N_k ($N_k > 1$) LEDs predicted in the image or linear array at time step k, the data association hypothesis at this single step $\theta(k)$ becomes a N_k tuple. Assuming the number of measurements at the i-th LED's predicted region is $m_{i,k}$, the number of hypothesis at a single step will be

$$\ell_k = \prod_{i=1}^{N_k} (m_{i,k} + 1)\,. \tag{5.18}$$

For total of K time steps, the number of joint data association hypotheses L_K becomes

$$L_K = \prod_{k=1}^{K} \ell_k. \tag{5.19}$$

Comparing eqn. (5.19) with (5.4), the total number of hypothesis is greatly increased for multiple LEDs measurements. The measurement set $\boldsymbol{Z}(k)$ at step k contains N_k subsets, and the i-th subset is denoted by symbol $\boldsymbol{Z}_i(k)$. Their relations can be explained by the following equations:

$$\boldsymbol{Z}(k) \triangleq \{\boldsymbol{ID}_i(k), \boldsymbol{Z}_i(k)\}_{i=1}^{N_k} \tag{5.20}$$

$$\boldsymbol{Z}_i(k) \triangleq \{\boldsymbol{z}_{i,j}(k)\}_{j=1}^{m_{i,k}}, \tag{5.21}$$

where $\boldsymbol{ID}_i(k)$ represents the ID index of subset $\boldsymbol{Z}_i(k)$. The definition in eqn. (5.21) is similar to that in eqn. (5.2) where the subscript i is omitted due to single LED.

After redefining the hypothesis and measurement set, the probability of the hypothesis could be analyzed identical to eqn. (5.7). However, the calculation of the first and second terms in eqn. (5.7) are different due to multiple LEDs. Different from

eqn. (5.8), the first term is decomposed as

$$\begin{aligned}
\boldsymbol{p}(\boldsymbol{Z}(k) \mid \theta^{k,\ell}, \boldsymbol{Z}^{k-1}, \boldsymbol{U}^{k-1}) &= \prod_{i=1}^{N_k} \boldsymbol{p}(\boldsymbol{Z}_i(k) \mid \theta^{k,\ell}, \boldsymbol{Z}^{k-1}, \boldsymbol{U}^{k-1}) \\
&= \prod_{i=1}^{N_k} \prod_{j=1}^{m_{i,k}} \boldsymbol{p}(\boldsymbol{z}_{i,j}(k) \mid \theta^{k,\ell}, \boldsymbol{Z}^{k-1}, \boldsymbol{U}^{k-1}) \\
&= \prod_{i=1}^{N_k} \prod_{j=1}^{m_{i,k}} \boldsymbol{f}(i,j). \qquad (5.22)
\end{aligned}$$

Similar to eqn. (5.9), the function $\boldsymbol{f}(i,j)$ is calculated by

$$\boldsymbol{f}(i,j) = \begin{cases} \frac{1}{V_{i,k}} & \text{for clutter,} \\ \mathcal{N}(\boldsymbol{z}_{i,j}(k); \hat{\boldsymbol{z}}_i^s(k), \boldsymbol{S}_i^s(k)) & \text{for } i\text{-th the LED,} \end{cases} \qquad (5.23)$$

where $V_{i,k}$ is the volume of the i-th LED's predicted region at time step k, and $\hat{\boldsymbol{z}}_i^s(k)$ and $\boldsymbol{S}_i^s(k)$) are the i-th LED's predicted position and error covariance matrix. Substituting (5.23) into (5.22), the first term is computed by

$$\begin{aligned}
&\boldsymbol{p}(\boldsymbol{Z}(k) \mid \theta^{k,\ell}, \boldsymbol{Z}^{k-1}, \boldsymbol{U}^{k-1}) \\
= \; &\frac{V_{i_1,k} \cdots V_{i_r,k}}{V_{1,k}^{m_{1,k}} \cdots V_{N_k,k}^{m_{N_k,k}}} \prod_{h=1}^{r} \mathcal{N}(\boldsymbol{z}_{\theta_{i_h}(k)}(k); \hat{\boldsymbol{z}}_{i_h}^s(k), \boldsymbol{S}_{i_h}^s(k)) \qquad (5.24)
\end{aligned}$$

where $\theta_{i_h}(k) \neq 0$ for $h = 1 \ldots r$. The second term in eqn. (5.7) is computed as

$$\begin{aligned}
&\boldsymbol{p}(\theta(k) \mid \theta^{k-1,s}, \boldsymbol{Z}^{k-1}, \boldsymbol{U}^{k-1}) \\
= \; &\frac{1}{m_{i_1,k} \cdots m_{i_r,k}} P_{on}^r (1 - P_{on})^{N_k - r} \boldsymbol{\mu}_F(m_k - r). \qquad (5.25)
\end{aligned}$$

5.3.3 q-best Hypotheses

From eqn. (5.4), the total number of possible hypotheses grows exponentilly with the time step K, while only one of them is correct. Computing the probability of all possible hypotheses and then discarding most of them is inefficient. To over come this problem, it is preferable to only compute those hypotheses having relatively high probability. This is reasonable since the correct hypothesis (LED path) should be always

among the most probable hypotheses. An efficient method to implement this approach was first introduced in [17]. The basic idea of this method is, at each time step, to only keep the q-best hypotheses, discarding the hypotheses that have lower probability. This method employs Murty's algorithm [57] to find the j-th best hypothesis solution.

Given the q-best hypotheses $\{\theta^{k-1,i}\}_{i=1}^{q}$ and their corresponding probabilities $\{\beta^{k-1,i}\}_{i=1}^{q}$ up to the former step $k-1$, and measurement set $Z(k)$ at current time step k, the new q-best hypotheses $\{\theta^{k,i}\}_{i=1}^{q}$ up to the current time step will be generated. Define $\theta^{k-1,i}$ to be the parent hypothesis of $\theta^{k,j}$, if the latter is the extension of the former. Choosing the single hypothesis j_k at time step k, $\theta^{k,*} = \{\theta^{k-1,i}, j_k\}$ is one possible extension of $\theta^{k-1,i}$, where '*' is replaced by an integer to enumerate the extensions. The best extension of $\theta^{k-1,i}$ is

$$\theta^{k,i_1} = \{\theta^{k-1,i}, j_k\} \text{ where } j_k = \arg\max_j \mathcal{L}_{i,j}(k). \tag{5.26}$$

The notation i_1 in the superscript of θ^{k,i_1} means that this hypothesis is the most probable extension of $\theta^{k-1,i}$. The best extension corresponds to the hypothesis having maximum marginal likelihood.

The following text describes the implementation details of the approach in [17]. To generate the q-best joint hypotheses up to current time step k, the first step is for each hypothesis $\theta^{k-1,i}$ to generate its best extension according to eqns. (5.15–5.16) and (5.26). These new joint hypotheses are ordered according to their probabilities and stored in the ordered list HYP-LIST. Their probabilities are calculated according to eqn. (5.13) and stored in PROB-LIST. The second step is for each hypothesis in HYP-LIST to use its parent hypothesis to generate the j-th ($j = 2, 3, \cdots$) best extension. If the probability of this extension is higher than the lowest probability in PROB-LIST, add it to HYP-LIST and the corresponding probability to PROB-LIST, and delete the

hypothesis with lowest probability from HYP-LIST and its corresponding probability from PROB-LIST. If the probability of this extension is lower than the lowest probability in PROB-LIST, stop generating new extensions by its parent hypothesis. After these processes conclude, the algorithm has produced the q-best hypotheses up to current time step. The algorithm is described in Table 5.1.

Note the factor $\mathcal{N}^{k-1,s}(z_j(k))$ in eqn. (5.13) is the Gaussian distribution evaluated at $z_j(k)$ with expectation $\hat{z}^s(k)$ and covariance $S^s(k)$. The process of computing the two parameters $\hat{z}^s(k)$ and $S^s(k)$ is discussed in Section 5.3.4.

5.3.4 Hypothesis: Computed Quantities

As we mentioned in Sec. 5.2, the rover state estimation and the LED path recovery are coupled. When computing the probability of each LED path hypothesis from eqn. (5.13), the algorithm also computes various other useful items as illustrated in (5.27):

$$\left\{\begin{array}{l} \hat{x}^{s+}(k-1) \\ \\ P^{s+}_{k-1} \end{array}\right. \Rightarrow \left\{\begin{array}{l} \hat{x}^{s-}(k) \\ \\ P^{s-}_{k} \end{array}\right. \Rightarrow \left\{\begin{array}{l} \hat{z}^{s}(k) \\ \\ S^{s}(k) \end{array}\right. \tag{5.27}$$

At time step $k-1$, we have the posterior state estimate $\hat{x}^{s+}(k-1)$ and error covariance matrix P^{s+}_{k-1} of each hypothesis $\theta^{k-1,s}$ where $s = 1, \cdots, q$. The '+' in the superscript indicates a posterior estimate. The '$-$' in the superscript indicates a prior estimate. The first arrow in (5.27) represents the state and covariance temporal propagation processes generating the prior quantities at time k for hypethesis s. The state and error covariance are temporally propagated using eqns. (3.2) and (3.6). The second arrow in (5.27) represents the (prior) measurement prediction process using eqns. (3.9) and (3.12).

Table 5.1: q-best hypotheses algorithm

Input: q-best hypotheses up to former time step $\{\theta^{k-1,i}\}_{i=1}^{q}$

the corresponding probabilities $\{\beta^{k-1,i}\}_{i=1}^{q}$

the corresponding state estimate $\{\hat{\boldsymbol{x}}^{i}\}_{i=1}^{q}$

current measurement set $Z(k)$

Output: q-best hypotheses up to current time step $\{\theta^{k,i}\}_{i=1}^{q}$

and their corresponding probabilities $\{\beta^{k,i}\}_{i=1}^{q}$

1. Initialize HYP-LIST and PROB-LIST

   HYP-LIST$\triangleq \{\theta^{k,i_1}\}_{i=1}^{q}$

   $\theta^{k,i_1} \triangleq \{\theta_{i,1}(k), \theta^{k-1,i}\}$

   $\theta_{i,1}(k) = \arg\max_{\theta(k)} p(\theta(k) \mid Z^k, \theta^{k-1,s}, U^{k-1})$

2. Sort the hypotheses in HYP-LIST according to their probabilities

3. for $i = 1 : q$

   If the j-th best new hypothesis generated by $\theta^{k-1,i}$ is still in HYP-LIST, then generate its $(j+1)$-th best new hypothesis.

   If the probability of the new hypothesis is higher than the lowest probability in PROB-LIST, then add it into the list and delete the hypothesis with the lowest probability.

   If not, break.

end

Here these processes are repeated for each hypothesis in set $\{\theta^{k-1,s}\}_{s=1}^{q}$.

The probability of new hypothesis $\{\theta^{k-1,s}, j\}$ can be evaluated after computing $\mathcal{N}^{k-1,s}(z_j(k))$. Using the algorithm presented in Sec. 5.3.3, the new q-best hypotheses up to current time step k are selected. Given each new hypothesis, for which $j \neq 0$, its posterior state estimate is updated according to the standard EKF measurement update process:

$$\hat{x}^{+}(k) = \hat{x}^{-}(k) + P_k^{-} H_k^{\top} S^{-1}(z_j(k) - \hat{z}(k)) \tag{5.28}$$

$$P_k^{+} = P_k^{-} - P_k^{-} H_k^{\top} S^{-1} H_k P_k^{-} \tag{5.29}$$

If $j = 0$, there is no measurement due to LED at current time step k; therefore, the posterior state estimate will be identical to the prior.

5.4 Vehicle Trajectory Recovery

This section derives the specific details of the presented approach for a land vehicle example that will be fully developed with application results presented in the next section.

5.4.1 Motion Sensor Model

For a land vehicle moving on a 2D plane (i.e., building floor), the rover navigation state vector can be defined as $\boldsymbol{x} = \begin{bmatrix} n & e & \psi \end{bmatrix}^{\top}$, where $\{n, e\}$ represent the 2D position coordinates in the navigation frame, and ψ represents the rover's yaw angle (heading). The (Dubbin's vehicle [20]) kinematic model is

$$\dot{n} = \cos(\psi)u, \quad \dot{e} = \sin(\psi)u, \quad \dot{\psi} = \omega. \tag{5.30}$$

The linear velocity u and angular rate ω can be computed as

$$\hat{u} = \tfrac{1}{2}(\hat{R}_L\dot{\phi}_L + \hat{R}_R\dot{\phi}_R), \quad \hat{\omega} = \tfrac{1}{L}(\hat{R}_L\dot{\phi}_L - \hat{R}_R\dot{\phi}_R), \tag{5.31}$$

where $\dot{\phi}_L$ and $\dot{\phi}_R$ denote the radian angular rates of rotation of each wheel and $\hat{R}$ are the wheel radii estimates. The quanitities $\dot{\phi}_L$ and $\dot{\phi}_R$ can be accurately computed based on encoder measurements so that eqn. (5.30) can be accurately computed in discrete-time, see Chapter 9 in [22].

The wheel radii errors are modeled as Gauss-Markov processes:

$$\delta\dot{R}_L = -\lambda_L\delta R_L + \omega_L, \quad \delta\dot{R}_R = -\lambda_R\delta R_R + \omega_R, \tag{5.32}$$

where the choice of time constant $1/\lambda_L$ and PSD of ω_L are discussed in Sec. 9.2 in [22]. The augmented state vector is $\boldsymbol{x} = [n, e, \psi, R_L, R_R]^\top$. With the error state vector defined as

$$\delta\boldsymbol{x} = [\delta n, \delta e, \delta\psi, \delta R_L, \delta R_R]^\top, \tag{5.33}$$

the error model is

$$\begin{aligned}
\delta\dot{\boldsymbol{x}} &= \boldsymbol{F}\delta\boldsymbol{x} + \boldsymbol{G}\boldsymbol{\nu} \quad \text{where} \qquad (5.34)\\
\boldsymbol{F} &= \begin{bmatrix}
0 & 0 & -\sin(\psi)u & \frac{\tilde{\dot{\phi}}_L \cos(\psi)}{2} & \frac{\tilde{\dot{\phi}}_R \cos(\psi)}{2} \\
0 & 0 & \cos(\psi)u & \frac{\tilde{\dot{\phi}}_L \sin(\psi)}{2} & \frac{\tilde{\dot{\phi}}_R \sin(\psi)}{2} \\
0 & 0 & 0 & \frac{1}{L}\tilde{\dot{\phi}}_L & -\frac{1}{L}\tilde{\dot{\phi}}_R \\
0 & 0 & 0 & -\lambda_L & 0 \\
0 & 0 & 0 & 0 & -\lambda_R
\end{bmatrix} \\
\boldsymbol{G} &= \begin{bmatrix}
\frac{R_L \cos(\psi)}{2} & \frac{R_R \cos(\psi)}{2} & 0 & 0 \\
\frac{R_L \sin(\psi)}{2} & \frac{R_R \sin(\psi)}{2} & 0 & 0 \\
\frac{1}{L}R_L & -\frac{1}{L}R_R & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix} \\
\boldsymbol{\nu} &= \begin{bmatrix} \omega_{\dot{\phi}_L} & \omega_{\dot{\phi}_R} & \omega_L & \omega_R \end{bmatrix}^\top . \qquad (5.35)
\end{aligned}$$

The derivation of matrices $\boldsymbol{F}$ and $\boldsymbol{G}$ with encoder measurements can be found in Appendix B. The encoder measurement noises are $\omega_{\dot{\phi}_L}$ and $\omega_{\dot{\phi}_R}$.

When the rover moves freely in 3D space, its motion information can be measured by inertial sensors such as IMU (inertial measurement unit). The IMU measures the rover's acceleration $\boldsymbol{a}$ and angular velocity $\boldsymbol{\omega}$ in three directions of the body frame. The state vector can be defined as $\boldsymbol{x} = \left[{}^n\boldsymbol{p}^\top, {}^n\boldsymbol{v}^\top, {}^b_n\bar{q}^\top, \boldsymbol{b}_a^\top, \boldsymbol{b}_g^\top\right]^\top$ to represent the position, velocity, rotation, and biases of acceleration and angular rate. The kinematic model using the measurements from IMU can be described by

$$\begin{aligned}
&{}^n\dot{\boldsymbol{p}} = {}^n\boldsymbol{v}, \quad {}^n\dot{\boldsymbol{v}} = {}^n\boldsymbol{a}, \quad {}^b_n\dot{\bar{q}} = \tfrac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}){}^b_n\bar{q} \\
&\dot{\boldsymbol{b}}_a = \boldsymbol{\omega}_{b_a}, \quad \dot{\boldsymbol{b}}_g = \boldsymbol{\omega}_{b_g}
\end{aligned}$$

The IMU measurements are modeled as

$$\tilde{\boldsymbol{a}} = \boldsymbol{R}_{\bar{q}}({}^{n}\boldsymbol{a} - {}^{n}\boldsymbol{g} + 2\lfloor\boldsymbol{\omega}_n\times\rfloor{}^{n}\boldsymbol{v} + \lfloor\boldsymbol{\omega}_n\times\rfloor^2{}^{n}\boldsymbol{p}) + \boldsymbol{b}_a + \boldsymbol{\omega}_a \tag{5.36}$$

$$\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega} + \boldsymbol{R}({}^{b}_{n}\bar{q})\boldsymbol{\omega}_n + \boldsymbol{b}_g + \boldsymbol{\omega}_g, \tag{5.37}$$

where $\boldsymbol{R}(\cdot)$ denotes the rotation matrix, and $\boldsymbol{\omega_n}$ is the rotation vector of the navigation frame with respect to the Earth centered inertial frame (ECI), and ${}^{\boldsymbol{n}}\boldsymbol{g}$ is the gravity represented in the navigation frame, and $\boldsymbol{\omega_a}$ and $\boldsymbol{\omega_g}$ are the measurement noise.

The state estimate propagates according to

$${}^{n}\dot{\hat{\boldsymbol{p}}} = {}^{n}\hat{\boldsymbol{v}} \tag{5.38}$$

$${}^{n}\dot{\hat{\boldsymbol{v}}} = \boldsymbol{R}_{\hat{\bar{q}}}^{\top}\hat{\boldsymbol{a}} - 2\lfloor\boldsymbol{\omega}_n\times\rfloor{}^{n}\hat{\boldsymbol{v}} - \lfloor\boldsymbol{\omega}_n\times\rfloor^2{}^{n}\hat{\boldsymbol{p}} + {}^{n}\boldsymbol{g} \tag{5.39}$$

$${}^{b}_{n}\dot{\hat{\bar{q}}} = \tfrac{1}{2}\boldsymbol{\Omega}(\hat{\boldsymbol{\omega}}){}^{b}_{n}\hat{\bar{q}}, \quad \dot{\hat{\boldsymbol{b}}}_a = 0, \quad \dot{\hat{\boldsymbol{b}}}_g = 0 \tag{5.40}$$

where $\hat{\boldsymbol{a}} = \tilde{\boldsymbol{a}} - \hat{\boldsymbol{b}}_{\boldsymbol{a}}$ and $\hat{\boldsymbol{\omega}} = \tilde{\boldsymbol{\omega}} - \boldsymbol{R}({}^{b}_{\boldsymbol{n}}\hat{\bar{\boldsymbol{q}}})\boldsymbol{\omega_n} - \hat{\boldsymbol{b}}_{\boldsymbol{g}}$. The error state is defined as $\boldsymbol{x} = \left[\delta{}^{n}\boldsymbol{p}^{\top}, \delta{}^{n}\boldsymbol{v}^{\top}, \delta\boldsymbol{\theta}^{\top}, \delta\boldsymbol{b}_a^{\top}, \delta\boldsymbol{b}_g^{\top}\right]^{\top}$, where $\delta\boldsymbol{\theta}$ represent the rotation from the computed frame $\hat{b}$ to the true frame b. Its model has the same structure as (3.3) with the following matrix format.

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3}\,\boldsymbol{0}_{3\times3} \\ -\lfloor\boldsymbol{\omega}_n\times\rfloor^2 & -2\lfloor\boldsymbol{\omega}_n\times\rfloor & -\boldsymbol{R}_{\hat{\bar{q}}}^{\top}\lfloor\hat{\boldsymbol{a}}\times\rfloor & -\boldsymbol{R}_{\hat{\bar{q}}}^{\top}\boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & -\lfloor(\hat{\boldsymbol{\omega}} + \boldsymbol{R}_{\hat{\bar{q}}}\boldsymbol{\omega}_n)\times\rfloor & \boldsymbol{0}_{3\times3}\ -\boldsymbol{I}_3 \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3}\,\boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3}\,\boldsymbol{0}_{3\times3} \end{bmatrix} \tag{5.41}$$

$$G = \begin{bmatrix} \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ -\boldsymbol{R}_{\hat{\bar{q}}}^{\top} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & -\boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 \end{bmatrix}, \ \boldsymbol{\nu} = \begin{bmatrix} \boldsymbol{\omega}_a \\ \boldsymbol{\omega}_g \\ \boldsymbol{\omega}_{b_a} \\ \boldsymbol{\omega}_{b_g} \end{bmatrix}.$$

The derivation of matrices $\boldsymbol{F}$ and $\boldsymbol{G}$ with IMU measurements can be found in Appendix C.

5.4.2 Photo-Detector Model

When using a camera to measure the LEDs, each LED projection in the image is a bright blob. The camera measurement model is described by eqn. (3.21) and (3.22) in Chapter 3. The camera measurement error model is defined in Sec. 3.2.3 of Chapter 3 with linearized measurement matrix derived in eqn. (3.31). The camera measurement error model is

$$\begin{aligned} \delta \boldsymbol{z} &= \boldsymbol{H}\delta\boldsymbol{x} + \boldsymbol{n} \\ &= -\boldsymbol{J}\,{}^{c}_{n}\hat{\boldsymbol{R}}\,\delta^{n}\boldsymbol{p} + \boldsymbol{J}\,{}^{c}_{b}\boldsymbol{R}\,\lfloor {}^{b}\hat{\boldsymbol{p}}_{bL}\times\rfloor\,\delta\boldsymbol{\theta} + \boldsymbol{n} \end{aligned} \tag{5.42}$$

where

$$\boldsymbol{J} = \frac{1}{{}^{c}\hat{z}}\begin{bmatrix} 1 & 0 & -\frac{{}^{c}\hat{x}}{{}^{c}\hat{z}} \\ 0 & 1 & -\frac{{}^{c}\hat{y}}{{}^{c}\hat{z}} \end{bmatrix}, \tag{5.43}$$

where $\delta^{n}\boldsymbol{p}$ is the position error vector of which the only nonzero components are δn and δe. The symbol $\delta\boldsymbol{\theta}$ represents the attitude error vector for which the only nonzero term is $\delta\psi$. Therefore, with the IMU error state as defined in eqn. (5.33), the camera aiding $\boldsymbol{H}$ matrix is

$$\boldsymbol{H} = \boldsymbol{J}\begin{bmatrix} -{}^{c}_{n}\hat{\boldsymbol{R}}(:,1:2) & {}^{c}_{b}\boldsymbol{R}\boldsymbol{J}_{\psi}\,{}^{b}\hat{\boldsymbol{p}}_{bL} & \boldsymbol{0}_{3\times 2} \end{bmatrix} \tag{5.44}$$

$$\boldsymbol{J}_{\psi} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{5.45}$$

The linear array measurement contains only half of the information in camera measurement, effectively only providing the $\tilde{u}$ measurement defined in (5.44). Therefore,

its measurement model is

$$\tilde{\boldsymbol{z}} = {}^{c}x/{}^{c}z + \boldsymbol{n}, \tag{5.46}$$

where ${}^{c}x$ and ${}^{c}z$ have the same meaning with that in (3.21).

5.5 Results

In the experiment, we use a rover equipped with wheel encoders and a camera to test and demonstrate the algorithm. Encoder based navigation is discussed in Chapter 9 of [22]. The rover implements a trajectory tracking controller which is a (nonadaptive) version of the command filtered backstepping approach described in the example section of [19]. The rover moves in a $5m \times 5m$ area with eleven LEDs mounted on the four walls at known locations. The position of the rover in the navigation frame is represented by the vector ${}^{n}\boldsymbol{p} = (n, e, d)$ where $d = 0$. Encoders attached to each rear wheel measure the wheel rotation, which allows computation of the rover speed u and angular rate w. The origin of the body frame is the center of the axle connecting the two rear wheels. The camera's position ${}^{b}\boldsymbol{p}_{bl}$ and pose ${}^{c}_{b}\boldsymbol{R}$ relative to the body frame are calibrated off-line. The parameter γ in eqn. (5.1) is selected such that the probability that the residual falls within $V_{l,\gamma}$ is 0.997.

5.5.1 Stationary

This section considers the case that the rover is stationary while the camera or linear array is detecting the LEDs. The rover is put close to position ${}^{n}\boldsymbol{p} = (-0.5, 0, 0)$ with rotation angle $\psi = 0^{o}$, so the prior estimate of the rover state is $\hat{\boldsymbol{x}}_0 = [-0.5, 0, 0]^{\top}$.

5.5.1.1 Camera

Since the inputs from encoders are zero, the estimate of the vehicle state remains unchanged prior to incorporating the camera measurements. At any time instant, for the current vehicle pose, only some of the LEDs will be within the field-of-view of the camera. An example of the predicted LED positions with error covariance ellipse projected onto the image plane is illustrate in Fig. 5.1. The resolution of the image is 640×512. We can see that only two LEDs are predicted to be in the image for the current rover state.

A time sequence of camera measurements is illustrated in Fig. 5.2. These measurements are extracted by thresholding the pixel intensities within each LED's predicted region. The green stars are the predicted LED positions at each time step based on the rover state estimate. The magenta crosses are the detected measurements. We can see from this figure that multiple measurement candidates exist in both of the predicted LED regions.

At the k-th measurement time step, the detected potential LED projection locations that fall into the detection region are enumerated and stored in the measurement set $Z(k)$ defined in eqn (5.3) of Section 5.2.2. The detection region is the minimum rectangle containing the predicted ellipse. The upper left image of Fig. 5.3 shows the detected and predicted measurements along with the prior uncertainty ellipse at time $t = 0$. Since there are two detected measurements in the detection region of LED 1 and no measurement in the detection region for LED 0, accounting for the null hypotheses, three data association hypotheses will be generated. Using each data association hypothesis to update the state estimate and then repredict the LED projection

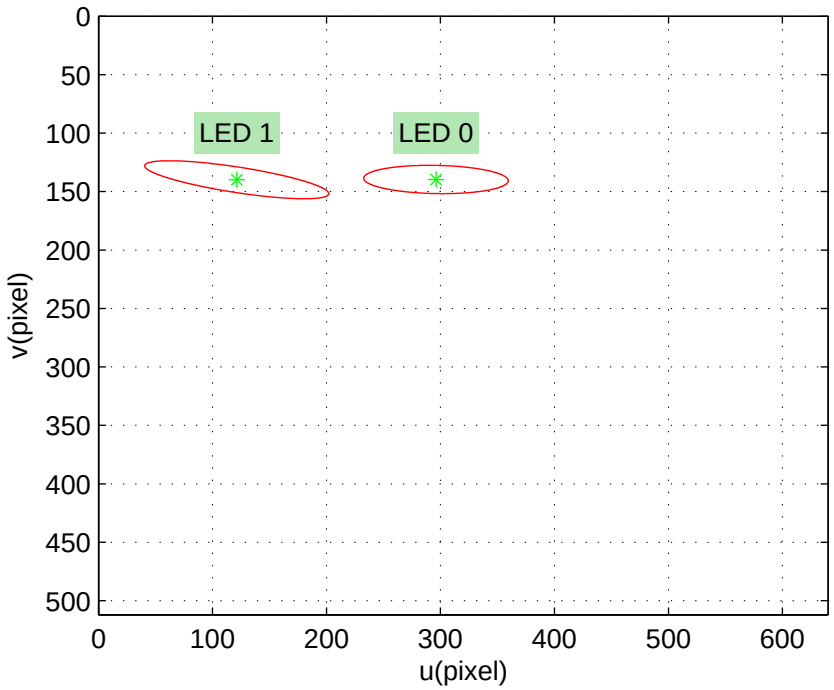
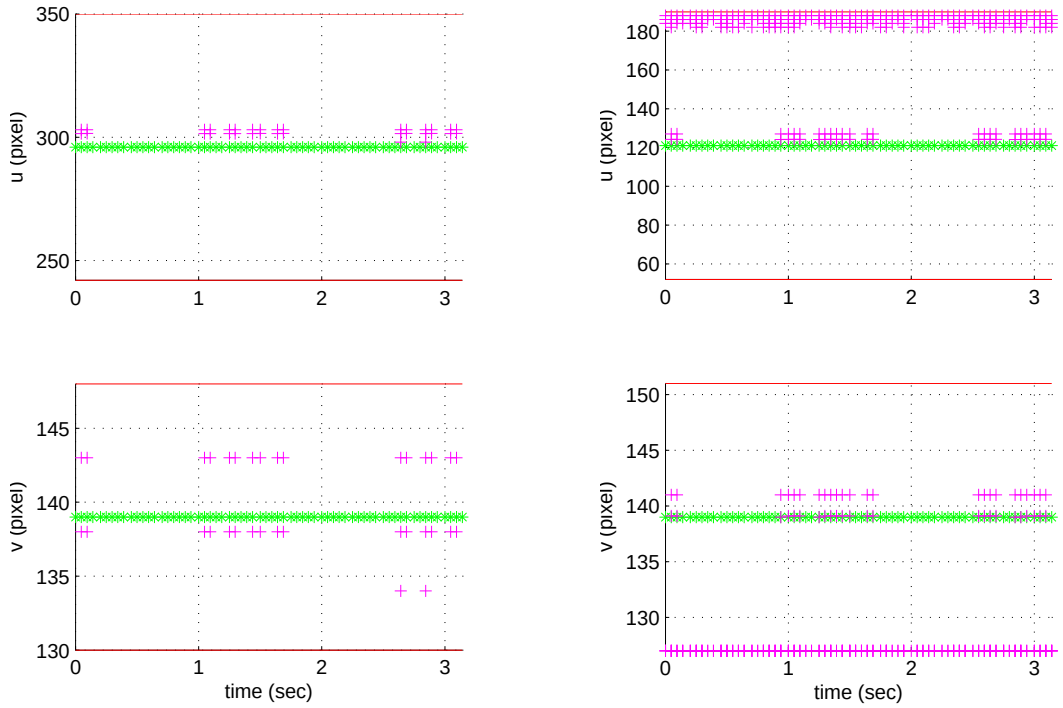


Figure 5.1: Predicted LEDs' positions in the image plane based on the prior information of the rover state: Predicted positions (green stars), 3-σ error ellipse regions (red)

location based on each hypothesis, the results are shown in the upper right image of Fig. 5.3. The blue "$*$" symbol represents the predicted LED projection location after updating the state by a hypothesis, and the corresponding blue ellipse represents shows its uncertainty. Note that the two null hypotheses have left the state estimate and its uncertainty unchanged, so they are identical to the prior and each other. The single measurement hypothesis has corrected the state according to its hypothesis and reduced the uncertainty. The bottom right image of Fig. 5.3 shows the predicted LED projection locations at the second measurement time $t = 0.05s$ with two detected measurements for LED 0 and four detected measurements for LED 1. At each time step, only the first $q = 10$ most probable hypotheses are kept and shown in the bottom left image by their corresponding posterior predicted LED locations and error ellipses.

The final navigation state estimation results for each hypothesis are shown in



(a) LED 0

(b) LED 1

Figure 5.2: Camera measurements of LED 0 and LED 1 in the first few seconds: Predicted LED positions (green stars), 3-σ error regions (red), LED measurements (magenta "+").

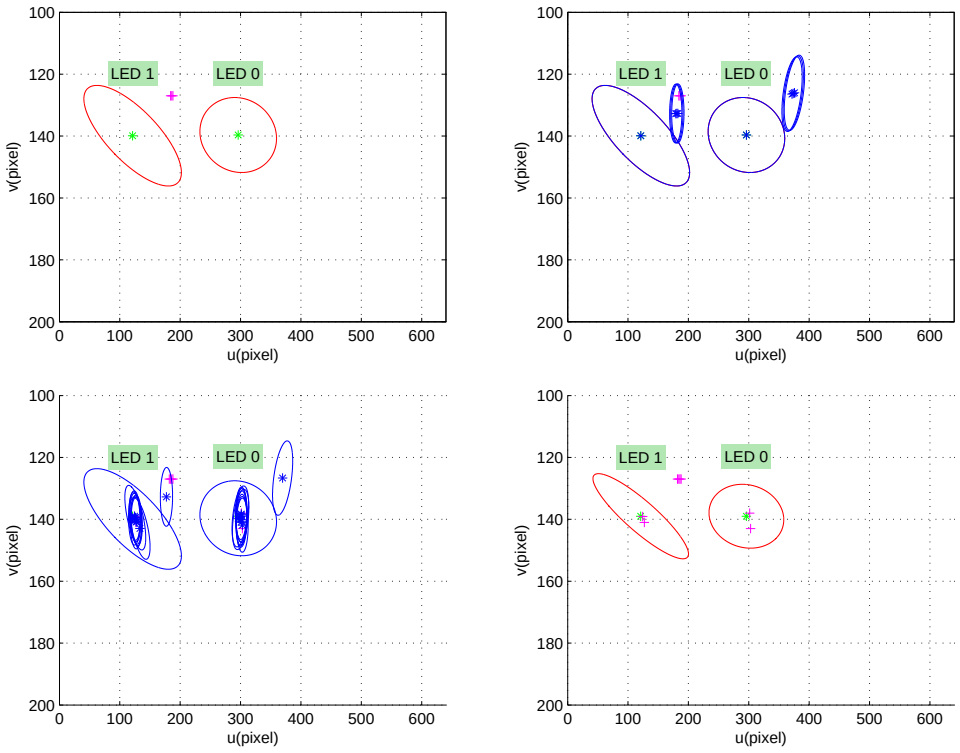


Figure 5.3: The measurements and hypotheses at the first two steps.

Fig. 5.4. The estimation results for the $q = 10$ different hypotheses are shown in different colors. Fig. 5.5 is revised version of Fig. 5.2 where we have changed the coloring of the measurements that have been selected as the most probable hypothesis sequence from magenta to blue.

The $q = 10$ data sequences shown below are recovered for LED 0 whose true ID is $\{00000000\}$. When sending the ID, each LED will add a 4-bit header $\{1010\}$ in the front and another 4-bit checksum in the back. For the correct data sequence, the checksum is $\{1010\}$. In this experiment, the camera works at a frame rate twice of the LED data rate, which means that each bit is recovered from two consecutive sampled images. The symbols "$*$" in the sequences indicates that this bit could not be determined due to different recovered statuses in the corresponding two consecutive steps. The numbers to the right of each sequence are the normalized probabilities. From these sequences, we will find the second one is the most probable correct data sequence.

$$\{*00000000010101010000000000101010\}, p = 0.24,$$

$$\{10000000001010101000000000101010\}, p = 0.16,$$

$$\{10000000001010101000000000101010\}, p = 0.13,$$

$$\{*00000000010101010000000000101010\}, p = 0.08,$$

$$\{00000000001010101000000000101010\}, p = 0.07,$$

$$\{*000000000 * 01010100000000000101010\}, p = 0.06,$$

$$\{*0000000001010 * 01000000000101010\}, p = 0.06,$$

$$\{*000000000101010 * 000000000101010\}, p = 0.06,$$

$$\{*00000000010101010000000010 * 010\}, p = 0.06,$$

$$\{*000000000101010101000000000 * 01010\}, p = 0.06,$$

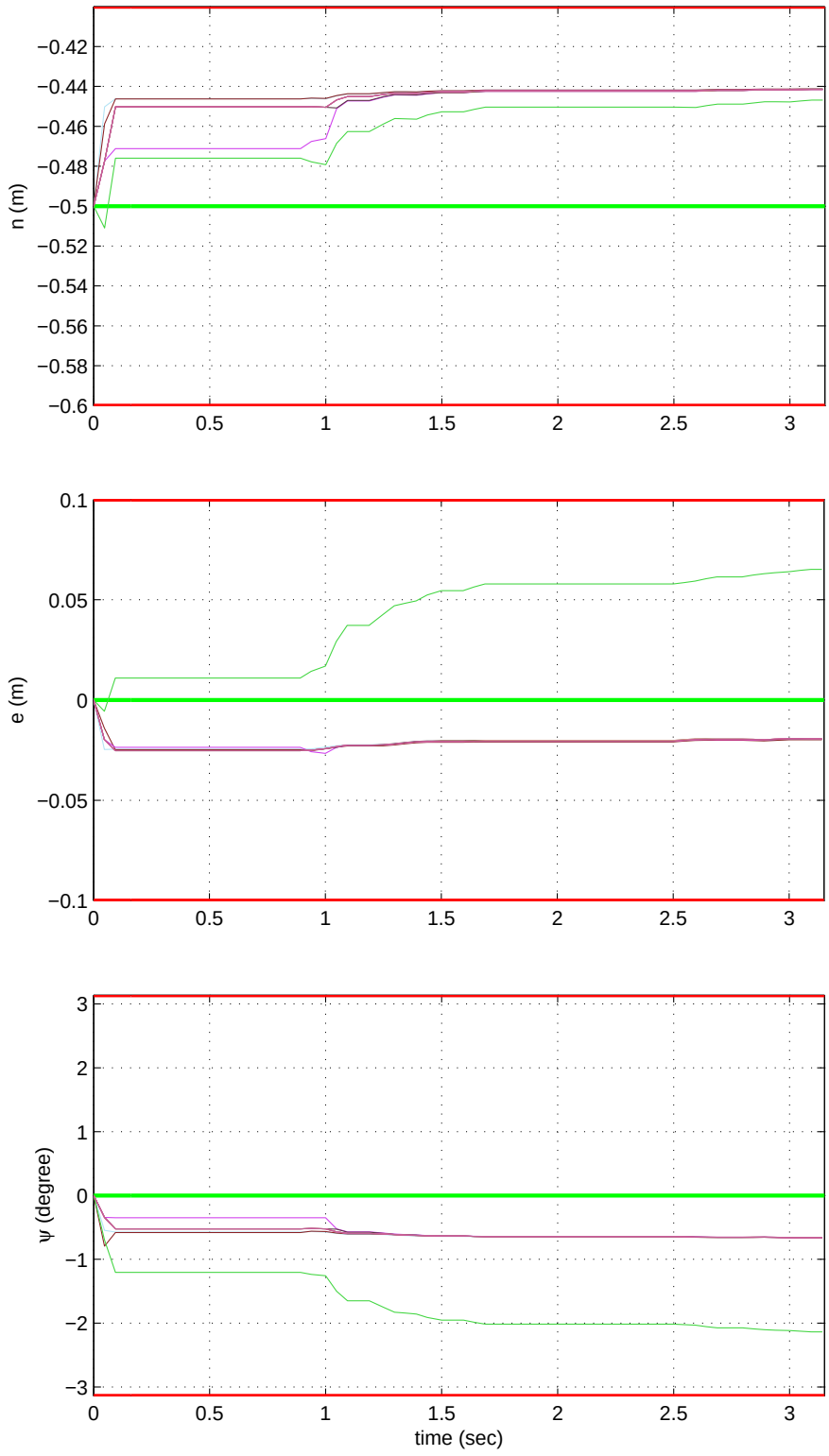
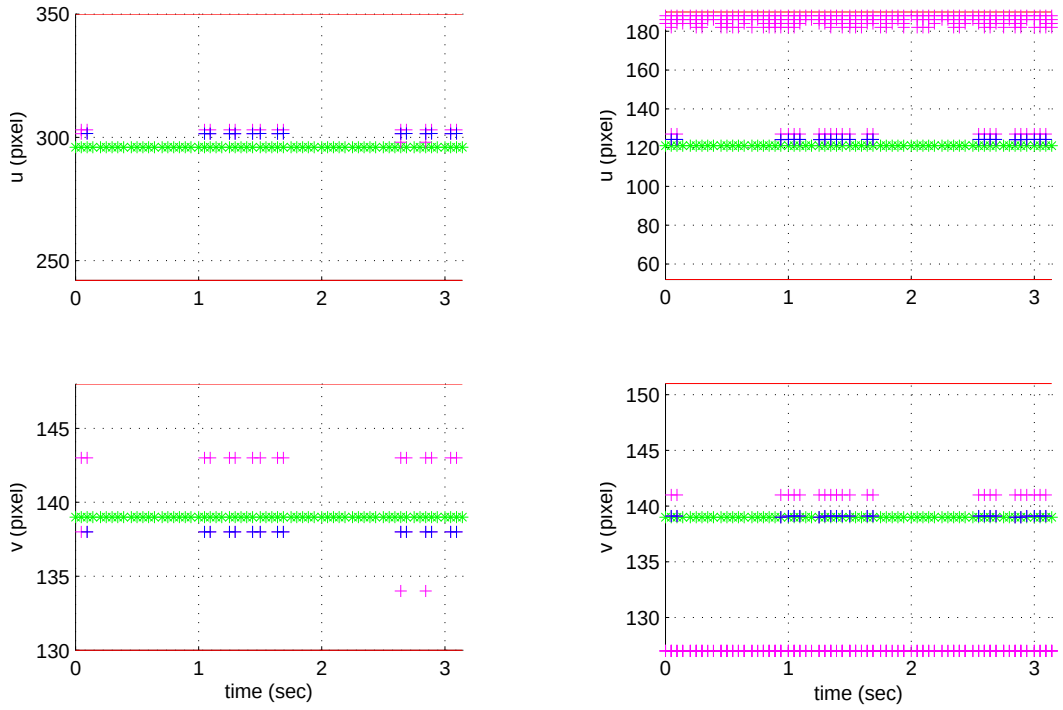


Figure 5.4: Estimation results associated with each hypothesis sequence.



(a) LED 0

(b) LED 1

Figure 5.5: Camera measurements with the most probable selection of the measurements at each time step.

5.5.1.2 Linear Array

The predicted LED position and error covariance in the linear array are illustrate in Fig. 5.6. The green stars are the predicted LED positions and the red parentheses are its uncertainty interval.

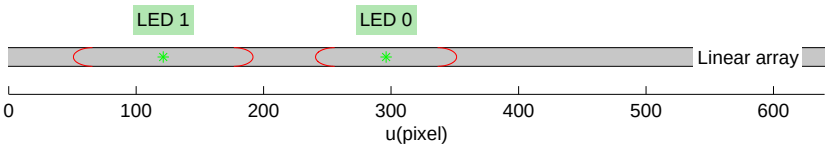


Figure 5.6: Predicted LEDs' positions and their uncertainty intervals in the linear array.

Similar to the process of camera measurements in Fig. 5.3, the process to generate new hypotheses by linear array measurements is illustrated in Fig. 5.7.

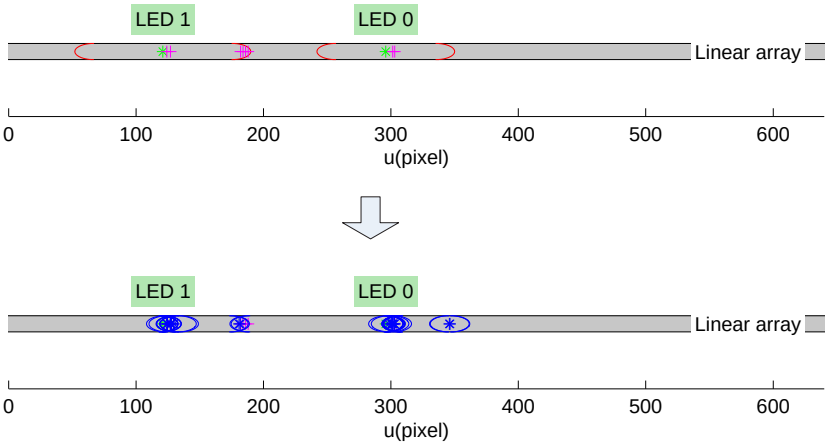


Figure 5.7: The linear array measurements and the new hypotheses.

The estimation result of each hypothesis using linear array measurements is shown in Fig. 5.8. The estimation result of each hypothesis is more disperse than the result of camera due to less informative measurements.

Fig. 5.9 shows the linear array measurements at each time step and the ones

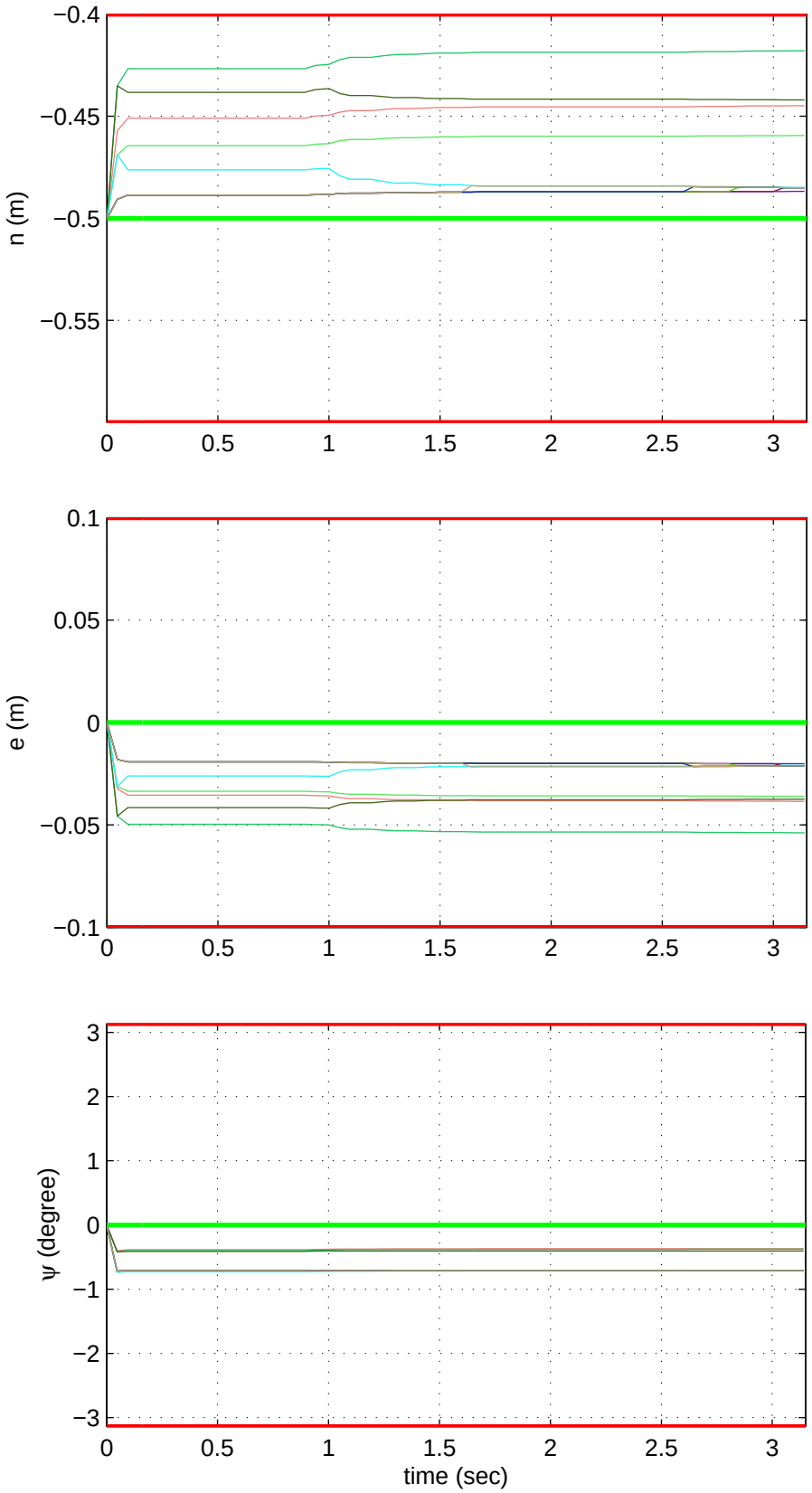
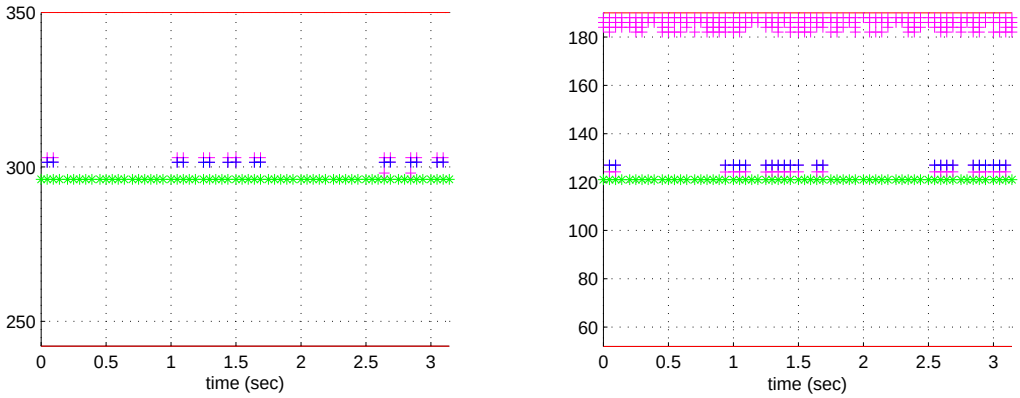


Figure 5.8: Estimation results associated with each hypothesis sequence using linear array measurements.

(blue "+") selected by the most probable hypothesis.



(a) LED 0

(b) LED 1

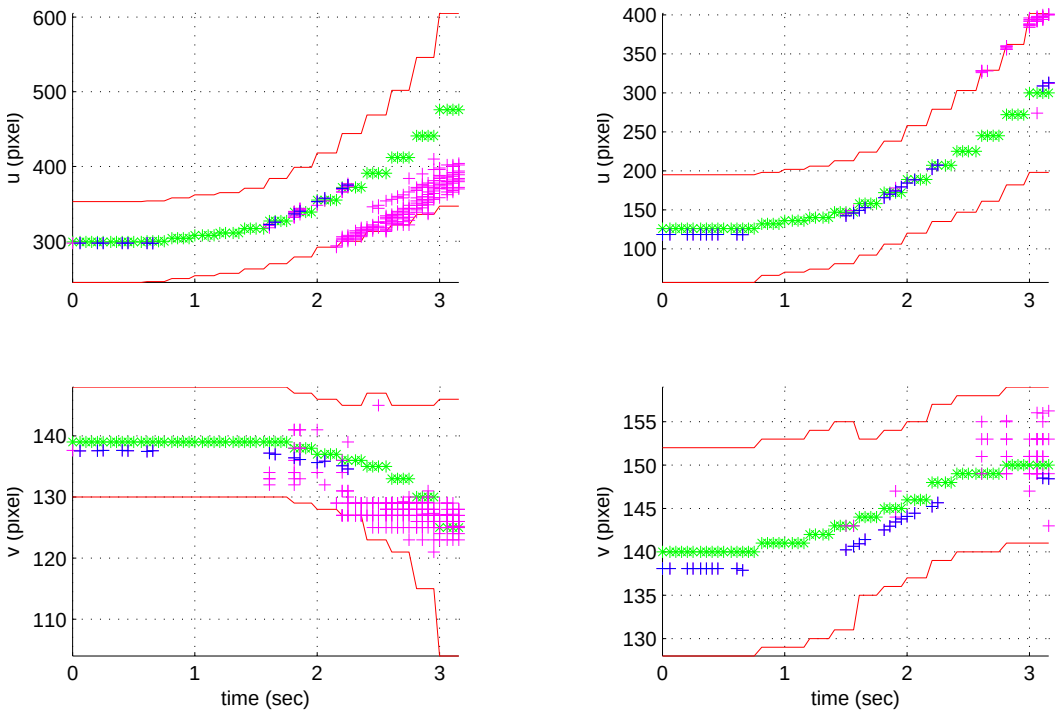
Figure 5.9: Linear array measurements with the most probable selection of the measurements at each time step.

5.5.2 Moving

5.5.2.1 Camera

When the vehicle is moving, the outputs from motion sensors (i.e. encoders) help to maintain an accurate state estimate for navigation and which enables accurate prediction of the LED projection locations in the image. Since the encoder measurements do not perfectly reproduce the rover speed and angular rate, the state error covariance of the prior state as well as the residual covariance both grow with time. An example set of actual and predicted LED location measurements at each time step are illustrated in Fig. 5.10. After applying the data recovery algorithm, the most probable data sequence with the correct ID is illustrated at blue "+". The process of generating the new hypotheses at one time step is illustrated in Fig. 5.12. The left and right images illustrate the predicted LED positions and their error covariances based on the prior and posterior state estimate, respectively. In this figure, all the measurements are not selected by

these most probable hypotheses. The prior and posterior state estimates associate with each hypothesis are shown in Fig. 5.11.



(a) LED 0 (b) LED 1

Figure 5.10: Coordinates of LED 0 and LED 1 in the image plane: Predicted LED positions (green stars), 3-σ error regions (red), LED measurements (blue "+"), noise and clutter measurements (magenta "+").

The following are the recovered data sequences of LED 0 for each hypothesis when the rover is moving. The second one is the most probable and correct data sequence. Note that several data sequences are identical, which doesn't not mean their hypothesis sequences are same since different choices may result in the same recovered

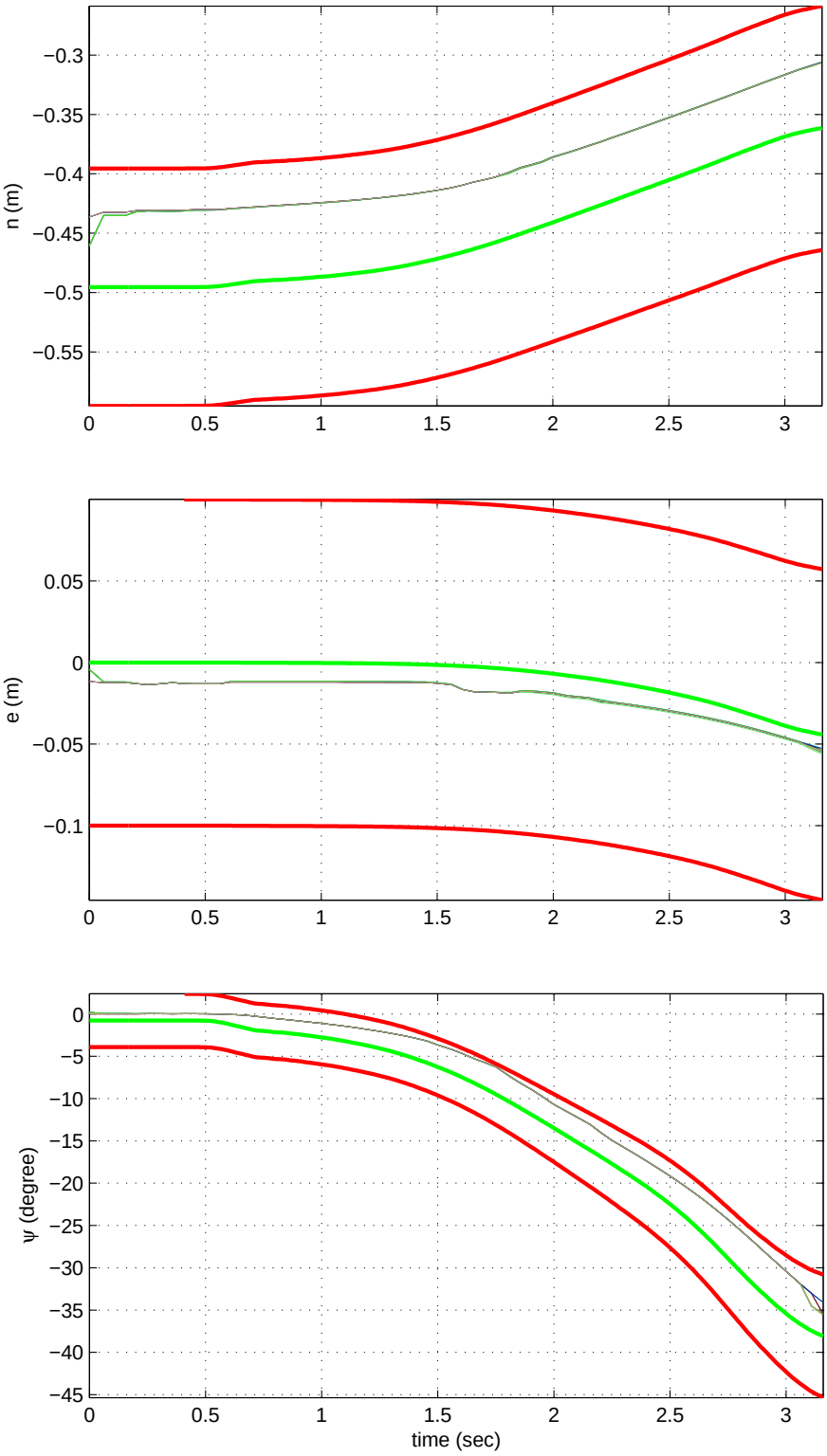
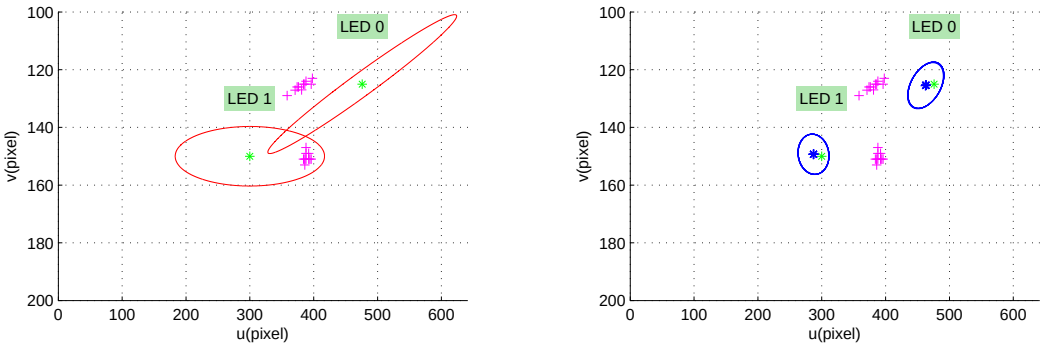


Figure 5.11: Estimation results associated with each hypothesis: State estimates only by motion sensors (green), standard deviation (red), and posterior state estimates of each hypothesis (other colors).



(a) Prior prediction.

(b) Posterior prediction.

Figure 5.12: Prior prediction (left) and posterior prediction of each hypothesis (right).

data.

$$\{*01010100000000010101010000000000\}, p = 0.23,$$

$$\{10101010000000001010101000000000\}, p = 0.21,$$

$$\{*01010100000000010 * 0101000000000\}, p = 0.12,$$

$$\{*0101010000000000101010 * 000000000\}, p = 0.10,$$

$$\{*01010100000000010101010000000000\}, p = 0.06,$$

$$\{*01010100000000010101010000000000\}, p = 0.06,$$

$$\{10101010000000001010101000000000\}, p = 0.06,$$

$$\{*01010100000000010101010000000000\}, p = 0.05,$$

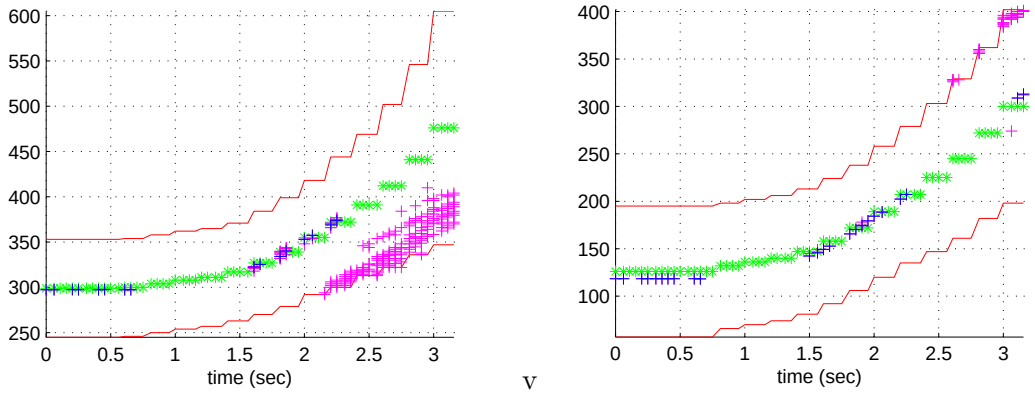
$$\{10101010000000001010101000000000\}, p = 0.05,$$

$$\{10101010000000001010101000000000\}, p = 0.05,$$

5.5.2.2 Linear Array

The estimation result when the rover is moving is shown in Fig. 5.14. Compare with Fig. 5.8, the estimation results are very close, which also proved accuracy of

estimating 2D state using linear array measurements. Fig. 5.13 shows the linear array measurements at each time step and the ones (blue "+") selected by the most probable hypothesis.



(a) LED 0

(b) LED 1

Figure 5.13: Linear array measurements with the most probable selection of the measurements at each time step when the rover is moving.

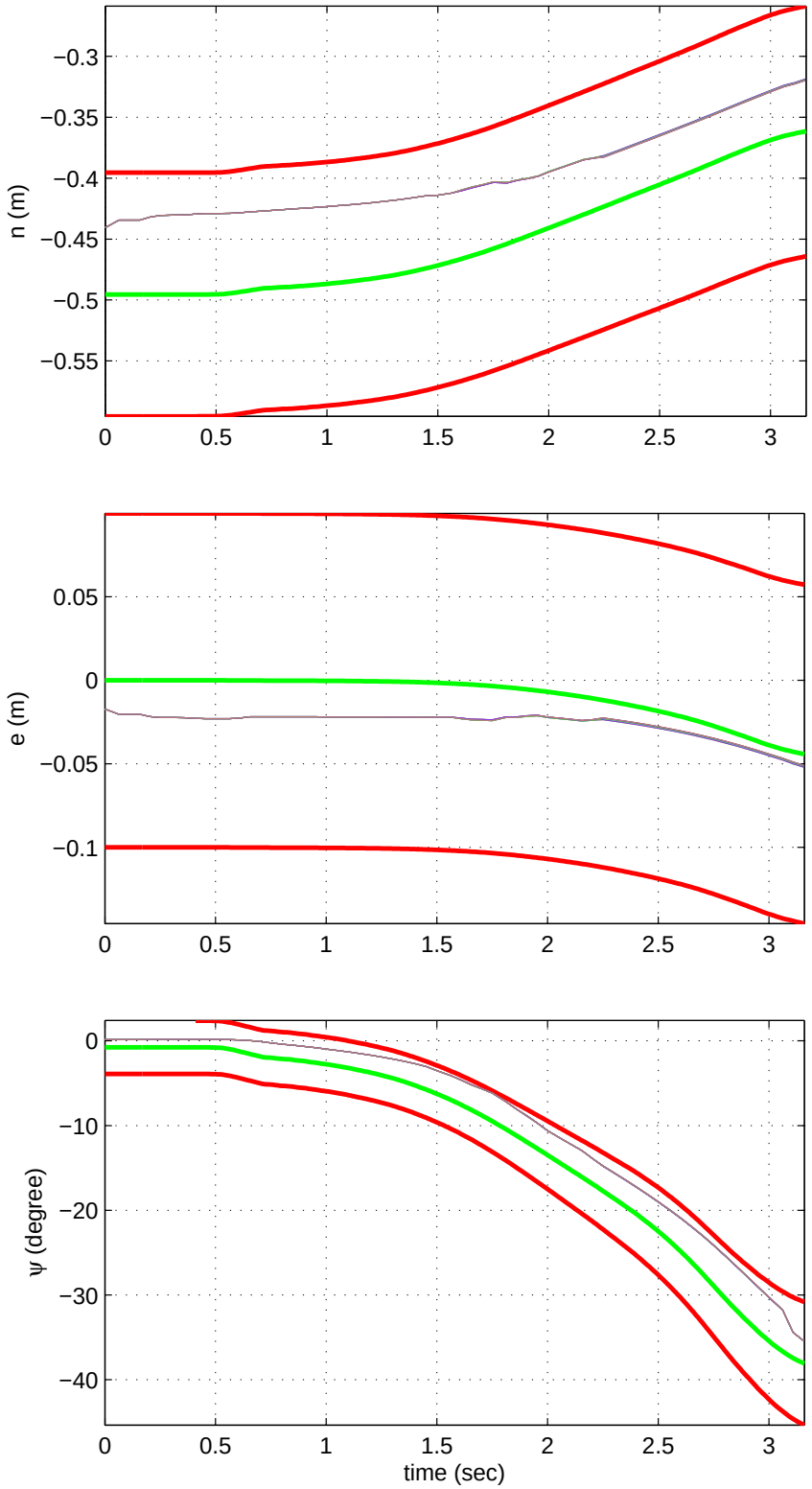


Figure 5.14: Estimation results associated with each hypothesis sequence using linear array measurements when the rover is moving.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Many personnel, equipment, and vehicular applications would have enhanced performance given more accurate knowledge of their location. LED light and signaling is in its infancy, offering unique dual use opportunities. Due to their high switching rates, which enables communication of unique ID's, compared with other vision-based navigation method, the detection and feature association sub-problems are more easily and reliably solved for LED features.

This dissertation has discussed various aspects of the communication and navigation using LED features. For the camera sensor, we demonstrate that the vehicle confined to move in the 2D-plane can be initialized by measuring at least two LEDs, as long as the vector joining the two LED's is not parallel to the world frame D-axis. With the initialization algorithm, no prior knowledge is required about the vehicle state and the EKF can be robustly and accurately initialized. When this rover is equipped with wheel encoders, the observability analysis shows that at least two LEDs are enough to ensure the observable of the navigation state.

To overcome the problem of low frame rate of normal camera, a new sensor is proposed that could simultaneously receive signals from LED's at a high rate and enable accurate navigation. This sensor combines a linear array, a convex-cylindrical lens and the shutter together to form a one-dimension photo-detector. The convex-cylindrical lens will focus the light that passes through onto a line instead of a point, so that the linear array preserves the sensitivity to angle of arrival relative to a single axis. Its one-dimension sensitivity increases the difficulty of the extrinsic parameter calibration and state initialization. An offline calibration method is proposed in this dissertation. The state initialization method is also analyzed. We also analyzed the observability of the navigation system and proved that at least two LED's with different x_f and y_f must be measured to have full observability of the vehicle state.

When using the photo-detector array (camera or linear array) to communicate with blinking LEDs, it is necessary to track the LED status in the sensor to extract the data. Two methods have been developed in this dissertation to solve the problem of accurately and efficiently extracting a data sequence. The first one is applicable to the situation when the data communication rate is high relative to the bandwidth of the moving platform so that the LED pixel projection changes little from one frame to the next. This approach is based on the Viterbi algorithm. The presentation included analysis and discussion of the necessary probability models, transition matrix, and measurement model. We also proved that the LED projection image size does not influence the final result. The results in Sec. 4.4 show the performance of this algorithm.

When the rover motion bandwidth is significant relative to the frame rate, the position of LED in the linear array can change significantly from one frame to the next. The assumption for the Viterbi-based algorithm is no longer valid. Another problem of this algorithm is that it is hard to be applied to the two-dimensional photo-detector such

as camera. The enumeration and connection between the pixels in the two-dimensional image is much more complicated than that in the linear array, so the state transition matrix for the two-dimensional measurements may not be diagonal anymore and hard to define. A second method is developed to overcome the above problems. This method is based on multiple hypothesis tracking, which tries to find the most probable candidate measurements at each time step instead of recover the LED positions in the sensor. The presentation included analysis and discussion of the hypothesis probability model and measurement model, and the experimental results both for camera and linear array measurements.

6.2 Future work

LED-based navigation and optical communication offer future synergistic performance improvement opportunities. In Chapter 3, we proposed a new sensor by combining a linear array and a cylindrical lens. DSP (digital signal processing) hardware will also be designed to sampling and process the data collected by the linear array to further improve the sampling and processing efficiency. The observability of the navigation equipped with a linear array and wheel encoders are analyzed when the rover moves in the 2D plane. Future research will consider the case that the rover can move freely in the 3D space. The wheel encoders could be replaced by the inertial measurement unit (IMU) to measure the acceleration and rotation. Lots of more issues should be considered for this linear array-IMU system including its observability and initialization.

Chapter 4 has focused on processing of a single thresholded LED. Future work will consider the approaches suitable for either multiple LED projections or multiple simultaneous hypotheses. Algorithms incorporating the measured intensity could also

be of interest. Finally, knowledge of the most likely LED image path provides useful information for correcting the rover trajectory over the corresponding time window.

Chapter 5 gives a more accurate data recovery algorithm based on MHT. The situation when a measurement can fall into multiple LEDs' uncertainty ellipse is not considered. This is illustrated in Fig. 6.1 for linear array measurements. Future work will consider the approach that associates the measurements to multiple LEDs' jointly. When multiple LED's data associations are jointly considered, each LED's data association hypothesis at a time step not only depends on its own former association hypotheses, but also other LEDs' association hypotheses. This could greatly increase the complexity of the assignment of the measurements to the LED's. This could be addressed by efficient assignment algorithms such as Auction [13] and JVC algorithm [37].

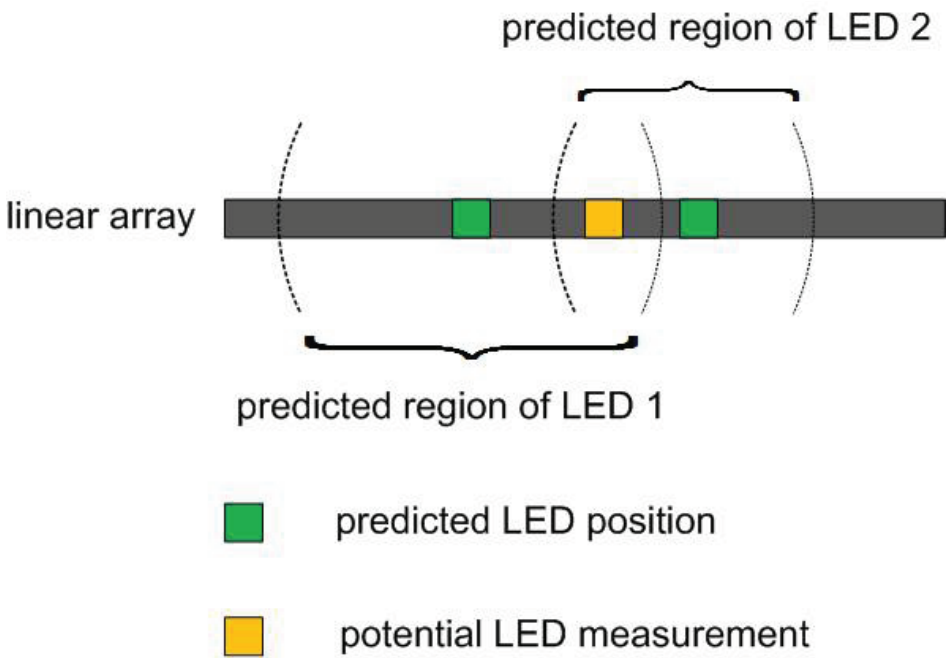


Figure 6.1: Linear array with one candidate measurement (yellow) falls into both the predicted region of two LEDs.

Another problem that should be considered in the future is the missing detection of LEDs. The proposed LED data recovery algorithm in this paper focuses on

solving the problem of LED false detections. As we mentioned in Sec. 5.3, the case of missing detection of LEDs is not considered. However, due to various reasons, both the camera and linear array may not detect the LEDs in the predicted region when the LED is "on". When the missing detection happens but not considered properly, inaccurate data will be recovered. Future work should also consider the LED passing in and out of the field of view. Chapter 4 and 5 both use the prior state estimate to calculate the measurement uncertainty region. This uncertainty region is considered to be fully contained in the image or linear array, which is not always true in practice. If this happens, some noise measurements or the LED measurement itself may be lost by merely processing the part of the predicted region that falls into the image or linear array. This is even more complicated than the case of missing detection.

Both the LED data recovery algorithms introduced in this dissertation requires extracting the whole ID data sequence firstly in order to judge them later based on the expected ID sequence later. In practice, it is not necessary to obtain the whole length of ID to determine whether it is valid. For example, assume that the frame rate is two time of the LED data rate, then the recovered status sequences are wrong if their "on" or "off" do not appear in pairs. This fact could greatly help delete the incorrect hypothesis sequences, so that computation will not be wasted on these incorrect ones. Future work should consider how to take advantage of these hypothesis pruning methods.

Besides the algorithms introduced in this dissertation, random sample consensus (RANSAC) is another choice for solving this data recovery algorithm. The advantage of using RANSAC is that more accurate data recovery as well as the rover state estimate may be obtained. The reason is that RANSAC estimates the parameters (rover states) simultaneously by all the measurements in the consensus set, which is similar to the update process of smoothing algorithms. The set of observations is $\boldsymbol{Z}^K$. The consensus

set should be modified to contain at most one measurement at each time step for each LED. Instead of fitting a trajectory that has as many as inliers, the objective should be modified to search for the most probable trajectory.

Data transmission using multiple colors [31] provides other methods to separate the LEDs from clutter or noise. LEDs can change their colors easily by modulating the driving signal. When the data is sent by LED's colored light, extracting the LED data will be very different from the algorithms in this dissertation. Data transmission using patterns should also be considered in the future work. Higher data rate can be achieved when transmitting patterns, but more complicated image processing technique is also required to accurately extract the pattern in the image.

Bibliography

[1] Mostafa Z Afgani, Harald Haas, Hany Elgala, and Dietmar Knipp. Visible light communication using ofdm. In *Testbeds and Research Infrastructures for the Development of Networks and Communities, 2006. TRIDENTCOM 2006. 2nd International Conference on*, pages 6–pp. IEEE, 2006.

[2] Motilal Agrawal, Kurt Konolige, and Morten Rufus Blas. Censure: Center surround extremas for realtime feature detection and matching. In *Computer Vision–ECCV 2008*, pages 102–115. Springer, 2008.

[3] A. Ansar and K. Daniilidis. Linear pose estimation from points or lines. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(5):578–589, 2003.

[4] S. Arai, S. Mase, T. Yamazato, T. Yendo, T. Fujii, M. Tanimoto, and Y. Kimur. Feasible study of road-to-vehicle communication system using LED array and high-speed camera. *15th World Congress on IT*, 2008.

[5] Shintaro Arai, Shohei Mase, Takaya Yamazato, Tomohiro Endo, Toshiaki Fujii, Masayuki Tanimoto, Kiyosumi Kidono, Yoshikatsu Kimura, and Yoshiki Ninomiya. Experimental on hierarchical transmission scheme for visible light communication using led traffic light and high-speed camera. In *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*, pages 2174–2178. IEEE, 2007.

[6] Ashwin Ashok, Marco Gruteser, Narayan Mandayam, Jayant Silva, Michael Varga, and Kristin Dana. Challenge: mobile optical networks through visual mimo. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, pages 105–112. ACM, 2010.

[7] Bo Bai, Gang Chen, Zhengyuan Xu, and Yangyu Fan. Visible light positioning based on led traffic light and photodiode. In *Vehicular Technology Conference (VTC Fall), 2011 IEEE*, pages 1–5. IEEE, 2011.

[8] Yaakov Bar-Shalom. *Tracking and data association*. Academic Press Professional, Inc., 1987.

[9] Yaakov Bar-Shalom, Sam S Blackman, and Robert J Fitzgerald. The dimensionless score function for multiple hypothesis decision in tracking. In *Systems, Man and Cybernetics, IEEE International Conference on*, 2005.

[10] Yaakov Bar-Shalom, Fred Daum, and Jim Huang. The probabilistic data association filter. *Control Systems Magazine, IEEE*, 29(6):82–100, 2009.

[11] Yaakov Bar-Shalom, X Rong Li, and Thiagalingam Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[12] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Computer Vision–ECCV 2006*, pages 404–417. Springer, 2006.

[13] Dimitri P Bertsekas. The auction algorithm: A distributed relaxation method for the assignment problem. *Annals of operations research*, 1988.

[14] Y.-Y. Bouguet. Camera calibration toolbox for Matlab. Online, http://www.vision.caltech.edu/bouguetj/calib_doc/index.html, 2010.

[15] R. Brunelli. *Template Matching Techniques in Computer Vision: Theory and Practice*. John Wiley and Sons, New York, 2009.

[16] Giorgio Corbellini, Stefan Schmid, Stefan Mangold, Thomas R Gross, and Armen Mkrtchyan. Demo: Led-to-led visible light communication for mobile applications. 2012.

[17] Ingemar J. Cox and Sunita L. Hingorani. An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(2):138–150, 1996.

[18] Kaiyun Cui, Gang Chen, Zhengyuan Xu, and Richard D Roberts. Line-of-sight visible light communication system design and demonstration. In *Communication Systems Networks and Digital Signal Processing (CSNDSP), 2010 7th International Symposium on*, pages 621–625. IEEE, 2010.

[19] W. Dong, J. A. Farrell, M. M. Polycarpou, V. Djapic, and M. Sharma. Command filtered adaptive backstepping. *IEEE Trans. on Control Systems Technology*, 20(3):566–580, 2012.

[20] L.E. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79(3):497516, 1957.

[21] Jos Elfring, Rob Janssen, and Rene van de Molengraft. Data association and tracking: A literature survey. Technical report, RoboEarth, 2010.

[22] J. A. Farrell. *Aided Navigation: GPS with High Rate Sensors*. McGraw-Hill, 2008.

[23] G. David Forney. The Viterbi algorithm. *Proc. IEEE*, 61(3):268–278, 1973.

[24] Wolfgang Förstner. A feature based correspondence algorithm for image matching. *International Archives of Photogrammetry and Remote Sensing*, 26(3):150–166, 1986.

[25] A. Frank, P. Smyth, and A. Ihler. A graphical model representation of the track-oriented multiple hypothesis tracker. In *Statistical Signal Processing Workshop (SSP), IEEE*, pages 768–771, 2012.

[26] A. Frank, P. Smyth, and A. Ihler. A graphical model representation of the track-oriented multiple hypothesis tracker. In *Statistical Signal Processing Workshop, IEEE*, pages 768–771, 2012.

[27] A. Golding and N Lesh. Indoor navigation using a diverse set of cheap wearable sensors. *The Third Int. Sym. on Wearable Computer*, 1999.

[28] J Grubor, OC Jamett, JW Walewski, S Randel, and K-D Langer. High-speed wireless indoor communication via visible light. *ITG-Fachbericht-Breitbandversorgung in Deutschland-Vielfalt für alle?*, 2007.

[29] Swook Hann, Jung-Hun Kim, Soo-Yong Jung, and Chang-Soo Park. White led ceiling lights positioning systems for optical wireless indoor applications. In *Proc. ECOC*, pages 1–3, 2010.

[30] Christopher G Harris and JM Pike. 3d positional integration from image sequences. *Image and Vision Computing*, 6(2):87–90, 1988.

[31] Schnichiro Haruyama. Visible light communications: Recent activities in japan. *Smart Spaces: A Smart Lighting ERC Industry Academia Day at BU Photonics Center, Boston University (Feb. 8, 2011)(49 pages)*, 2011.

[32] Shinichiro Haruyama. Advances in visible light communication technologies. In *European Conference and Exhibition on Optical Communication*. Optical Society of America, 2012.

[33] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997.

[34] R. Hermann and A. Krener. Nonlinear controllability and observability. *Automatic Control, IEEE Transactions on*, 22(5):728 – 740, Oct 1977.

[35] Steve Hewitson, Hung Kyu Lee, and Jinling Wang. Localizability analysis for gps/galileo receiver autonomous integrity monitoring. *Journal of Navigation*, 57(02):245–259, 2004.

[36] Steve Hewitson and Jinling Wang. Extended receiver autonomous integrity monitoring (e raim) for gnss/ins integration. *Journal of Surveying Engineering*, 136(1):13–22, 2010.

[37] Roy Jonker and Anton Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 1987.

[38] Mohsen Kavehrad. Broadband room service by light. *Scientific American*, 2007.

[39] Talha Ahmed Khan. Visible light communications using wavelength division multiplexing. Bachelor's thesis, Electrical Engineering Department of the University of Engineering and Technology Lahore, 2006.

[40] Kamran Kiasaleh. Performance analysis of free-space on-off-keying optical communication systems impaired by turbulence. In *Proc. SPIE*, volume 4635, pages 150–161, 2002.

[41] Hyun-Seung Kim, Deok-Rae Kim, Se-Hoon Yang, Yong-Hwan Son, and Sang-Kook Han. Indoor positioning system based on carrier allocation visible light communication. In *Conference on Lasers and Electro-Optics/Pacific Rim*, page C327. Optical Society of America, 2011.

[42] Hyun-Seung Kim, Deok-Rae Kim, Se-Hoon Yang, Yong-Hwan Son, and Sang-Kook Han. An indoor visible light communication positioning system using a rf carrier allocation technique. *Lightwave Technology, Journal of*, 31(1):134–144, 2013.

[43] Thomas King, Stephan Kopf, Thomas Haenselmann, Christian Lubberger, and Wolfgang Effelsberg. Compass: A probabilistic indoor positioning system based on 802.11 and digital compasses. In *Proceedings of the 1st international workshop on Wireless network testbeds, experimental evaluation & characterization*, pages 34–40. ACM, 2006.

[44] T. Komine and M. Nakagawa. Fundamental analysis for visible-light communication system using LED lights. *IEEE Trans. Consumer Electron*, 50(1), 2004.

[45] Toshihiko Komine, Shinichiro Haruyama, and Masao Nakagawa. A study of shadowing on indoor visible-light wireless communication utilizing plural white led lightings. *Wireless Personal Communications*, 34(1-2):211–225, 2005.

[46] Toshihiko Komine and Masao Nakagawa. Integrated system of white led visible-light communication and power-line communication. *Consumer Electronics, IEEE Transactions on*, 49(1):71–79, 2003.

[47] T Kurien. Issues in the design of practical multitarget tracking algorithms. *Multitarget-Multisensor Tracking: Advanced Applications*, 1990.

[48] H. Le-Minh, L. Zeng, D.C. O'Brien, O. Bouchet, S. Randel, J. Walewski, J.A.R. Borges, K.-D. Langer, J.G. grubor, K. Lee, and E.T Won. Short-range visible light communications. *Wireless World Research Forum*, 2007.

[49] Hao Le Minh, Dominic O'Brien, Grahame Faulkner, Lubin Zeng, Kyungwoo Lee, Daekwang Jung, and YunJe Oh. High-speed visible light communications using multiple-resonant equalization. *Photonics Technology Letters, IEEE*, 20(14):1243–1245, 2008.

[50] Hugh Sing Liu and Grantham Pang. Positioning beacon system using digital camera and leds. *Vehicular Technology, IEEE Transactions on*, 52(2):406–419, 2003.

[51] Jiang Liu, W. Noonpakdee, H. Takano, and S. Shimamoto. Foundational analysis of spatial optical wireless communication utilizing image sensor. In *IEEE Int. Conf. on Imaging Systems and Techniques*, 2011.

[52] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[53] De Maesschalck, D. Jouan-Rimbaud, and D.L. Massart. The Mahalanobis distance. *Chemometrics and Intelligent Laboratory Systems*, 50, 2000.

[54] Nakagawa Masao. Visible light communications consortium (vlcc). `http://www.vlcc.net`.

[55] Hans P Moravec. Obstacle avoidance and navigation in the real world by a seeing robot rover. Technical report, DTIC Document, 1980.

[56] Anastasios I Mourikis and Stergios I Roumeliotis. A multi-state constraint kalman filter for vision-aided inertial navigation. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3565–3572. IEEE, 2007.

[57] Katta G. Murty. An algorithm for ranking all the assignments in order of increasing cost. *Operations Research*, 16:682–687, 1968.

[58] T. Nagura, T. Yamazato, M. Katayama, T. Yendo, T. Fujii, and H. Okada. Improved decoding methods of visible light communication system for ITS using LED array and high-speed camera. In *IEEE 71st Vehicular Technology Conf.*, 2010.

[59] Dominic O'brien, Lubin Zeng, Hoa Le-Minh, Grahame Faulkner, Joachim W Walewski, and Sebastian Randel. Visible light communications: Challenges and possibilities. In *Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on*, pages 1–5. IEEE, 2008.

[60] Jun Ohta, Koji Yamamoto, Takao Hirai, Keiichiro Kagawa, Masahiro Nunoshita, Masashi Yamada, Yasushi Yamasaki, Shozo Sugishita, and Kunihiro Watanabe. An image sensor with an in-pixel demodulation function for detecting the intensity of a modulated light signal. *Electron Devices, IEEE Transactions on*, 50(1):166–172, 2003.

[61] Grantham Pang, Chi ho Chan, Hugh Liu, and Thomas Kwan. Dual use of LEDs: Signaling and communications in ITS. *5th World Congr. Intelligent Transport Systems*, 1998.

[62] Grantham KH Pang and Hugh HS Liu. Led location beacon system based on processing of digital images. *Intelligent Transportation Systems, IEEE Transactions on*, 2(3):135–150, 2001.

[63] Kusha Panta and Jean Armstrong. Indoor localisation using white leds. *Electronics letters*, 48(4):228–230, 2012.

[64] Halpage Chinthaka Nuwandika Premachandra, Tomohiro Yendo, Mehrdad Panahpour Tehrani, Takaya Yamazato, Hiraku Okada, Toshiaki Fujii, and Masayuki Tanimoto. High-speed-camera image processing based led traffic light detection for road-to-vehicle visible light communication. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 793–798. IEEE, 2010.

[65] Mohammad Shaifur Rahman, Md Mejbaul Haque, and Ki-Doo Kim. High precision indoor positioning using lighting led and image sensor. In *Computer and Information Technology (ICCIT), 2011 14th International Conference on*, pages 309–314. IEEE, 2011.

[66] Mohammad Shaifur Rahman, Md Mejbaul Haque, and Ki-Doo Kim. Indoor positioning by led visible light communication and image sensors. *International Journal of Electrical and Computer Engineering (IJECE)*, 1(2):161–170, 2011.

[67] Donald Reid. An algorithm for tracking multiple targets. *Automatic Control, IEEE Transactions on*, 24(6):843–854, 1979.

[68] Rick Robert. Intel labs camera communications (CamCom). Technical report, Intel Lab, 2013.

[69] Richard Roberts, Praveen Gopalakrishnan, and Somya Rathi. Visible light positioning: automotive use case. In *Vehicular Networking Conference (VNC), 2010 IEEE*, pages 309–314. IEEE, 2010.

[70] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006*, pages 430–443. Springer, 2006.

[71] Chinnapat Sertthin, Emiko Tsuji, Masao Nakagawa, Shigeru Kuwano, and Kazuji Watanabe. A switching estimated receiver position scheme for visible light based indoor positioning system. In *Wireless Pervasive Computing, 2009. ISWPC 2009. 4th International Symposium on*, pages 1–5. IEEE, 2009.

[72] Jianbo Shi and Carlo Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 593–600. IEEE, 1994.

[73] D. Sun and J.L. Crassidis. Observability analysis of six-degree-of-freedom configuration determination using vector observations. *Journal of guidance, control, and dynamics*, 25(6):1149–1157, 2002.

[74] Y Tanaka, T Komine, Shinichiro Haruyama, and M Nakagawa. Indoor visible communication utilizing plural white leds as lighting. In *Personal, Indoor and Mobile Radio Communications, 2001 12th IEEE International Symposium on*, volume 2, pages F–81. IEEE, 2001.

[75] Yuichi Tanaka, Toshihiko Komine, Shinichiro Haruyama, and Masao Nakagawa. Indoor visible light data transmission system utilizing white led lights. *IEICE transactions on communications*, 86(8):2440–2454, 2003.

[76] Michael Varga, Ashwin Ashok, Marco Gruteser, Narayan Mandayam, Wenjia Yuan, and Kristin Dana. Demo: visual mimo based led-camera communication applied to automobile safety. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 383–384. ACM, 2011.

[77] Andrew Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Trans.*, 13(2):260–269, 1967.

[78] V.Kulyukin, C. Gharpure, J. Nicholson, and S. Pavithran. RFID in robot-assisted indoor navigation for the visually impaired. *IEEE/RSJ Int. Conf. Intell. Robots Syst*, 2004.

[79] Anh Quoc Vu. *Robust Vehicle State Estimation for Improved Traffic Sensing and Management*. PhD thesis, University of California, Riverside, 2011.

[80] Jelena Vučić, Christoph Kottke, Stefan Nerreter, Klaus-Dieter Langer, and Joachim W Walewski. 513 mbit/s visible light communications link based on dmt-modulation of a white led. *Journal of Lightwave Technology*, 28(24):3512–3518, 2010.

[81] Dong Wang, Huchuan Lu, and Ming-Hsuan Yang. Least soft-threshold squares tracking. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2371–2378. IEEE, 2013.

[82] S-H Yang, E-M Jeong, D-R Kim, H-S Kim, Y-H Son, and S-K Han. Indoor three-dimensional location estimation based on led visible light communication. *Electronics Letters*, 49(1):54–56, 2013.

[83] Masaki Yoshino, Shinichiro Haruyama, and Masao Nakagawa. High-accuracy positioning system using visible led lights and image sensor. In *Radio and Wireless Symposium, 2008 IEEE*, pages 439–442. IEEE, 2008.

[84] Dongfang Zheng, Gang Chen, and Jay A. Farrell. Navigation using linear photodetector arrays. In *IEEE Multi-conference on Systems and Control*, 2013.

[85] Dongfang Zheng, Kaiyun Cui, Bo Bai, Gang Chen, and Jay A. Farrell. Indoor localization based on LEDs. In *IEEE Multi-conference on Systems and Control*, pages 573–578, 2011.

[86] Dongfang Zheng, Rathavut Vanitsthian, Gang Chen, and Jay A. Farrell. LED-based initialization and navigation. In *American Control Conference*, pages 6199–6205, 2013.

[87] Zhou Zhou, Mohsen Kavehrad, and Peng Deng. Indoor positioning algorithm using light-emitting diode visible light communications. *Optical Engineering*, 51(8):085009–1, 2012.

Appendix A

Nonlinear Optimization Methods

The general nonlinear convex optimization problem could be modeled as

$$\hat{\boldsymbol{x}}_{opt} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^n} \boldsymbol{r}(x), \tag{A.1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$, and $\boldsymbol{r}(x) : \mathbb{R}^n \to \mathbb{R}$ is a nonlinear convex function of $\boldsymbol{x}$. Lots of algorithms have been proposed to find $\boldsymbol{x}$ to minimize Eqn. (A.1): Gauss-Newton, Gradient Descent and Powell's Dog-Leg method. The commonality between these methods is that they all compute $\hat{\boldsymbol{x}}$ iteratively based on a initial value $\hat{\boldsymbol{x}}^0$. In iteration l, they try to compute a correction $\delta\boldsymbol{x}^l$ to obtain a better estimation $\hat{\boldsymbol{x}}^{l+1} = \hat{\boldsymbol{x}}^l + \delta\boldsymbol{x}^l$.

In iteration l, the value of objective function $\boldsymbol{r}(x)$ in the neighborhood of $\hat{\boldsymbol{x}}^l$ could be approximated as

$$\boldsymbol{r}(\boldsymbol{x}) \approx \boldsymbol{r}(\hat{\boldsymbol{x}}^l) + \nabla\boldsymbol{r}^\top(\hat{\boldsymbol{x}}^l)\delta\boldsymbol{x}^l, \tag{A.2}$$

where $\delta\boldsymbol{x}^l = \boldsymbol{x} - \hat{\boldsymbol{x}}^l$ and $\nabla\boldsymbol{r}(\hat{\boldsymbol{x}}^l) \in \mathbb{R}^{n\times 1}$ is the gradient vector of $\boldsymbol{r}(\boldsymbol{x})$ at $\hat{\boldsymbol{x}}^l$. In the following text, we will introduce the nonlinear optimization methods of Gradient Descent, Newton and Gauss-Newton.

A.1 Gradient Descent Method

Within the neighborhood of current point $\hat{\boldsymbol{x}}^l$, the objective function decreases fastest opposite to the direction of its gradient $\nabla \boldsymbol{r}$ at $\hat{\boldsymbol{x}}^l$. Based on this fact, the next better point $\hat{\boldsymbol{x}}^{l+1}$ could be computed by

$$\hat{\boldsymbol{x}}^{l+1} = \hat{\boldsymbol{x}}^l - \gamma_l \nabla \boldsymbol{r}(\hat{\boldsymbol{x}}^l), \tag{A.3}$$

where $\gamma_l > 0$ is the length of this step. To choose the step length γ_l that makes the objective function $\boldsymbol{r}(x)$ decrease most, we can take the derivative of $\boldsymbol{r}(\hat{\boldsymbol{x}}^l - \gamma_l \nabla \boldsymbol{r}(\hat{\boldsymbol{x}}^l))$ with respect to γ_l and set it be zero:

$$\nabla \boldsymbol{r}^\top(\hat{\boldsymbol{x}}^{l+1}) \nabla \boldsymbol{r}(\hat{\boldsymbol{x}}^l) = 0. \tag{A.4}$$

Eqn. (A.4) indicates that the step length γ_l is chosen such that the gradients of $\boldsymbol{r}$ at point $\hat{\boldsymbol{x}}^l$ and $\hat{\boldsymbol{x}}^{l+1}$ are perpendicular to each other.

A.2 Newton Method

The Newton method approximates the function $\boldsymbol{r}$ at at the neighborhood of $\hat{\boldsymbol{x}}^l$ as its second order Taylor expansion:

$$\boldsymbol{r}(\boldsymbol{x}) \approx \boldsymbol{r}(\hat{\boldsymbol{x}}^l) + \nabla \boldsymbol{r}^\top(\hat{\boldsymbol{x}}^l) \delta \boldsymbol{x}^l + \frac{1}{2} \delta \boldsymbol{x}^{l\top} \Delta \boldsymbol{r}(\hat{\boldsymbol{x}}^l) \delta \boldsymbol{x}^l, \tag{A.5}$$

where $\Delta \boldsymbol{r}(\hat{\boldsymbol{x}}^l) \in \mathbb{R}^{n \times n}$ is the Hessian matrix of $\boldsymbol{r}$ at $\hat{\boldsymbol{x}}^l$. Taking the derivative of eqn. (A.5) with respect to $\delta \boldsymbol{x}^l$ and setting it to zero, we have

$$\nabla \boldsymbol{r}^\top(\hat{\boldsymbol{x}}^l) + \delta \boldsymbol{x}^{l\top} \Delta \boldsymbol{r}(\hat{\boldsymbol{x}}^l) = 0. \tag{A.6}$$

Using eqn. (A.6), the step vector at iteration l is

$$\delta \boldsymbol{x}^l = -\Delta \boldsymbol{r}(\hat{\boldsymbol{x}}^l)^{-1} \nabla \boldsymbol{r}(\hat{\boldsymbol{x}}^l). \tag{A.7}$$

A.3 Gauss-Newton Method

Gauss-Newton method is a modification of Newton method to solve nonlinear least square (NLS) problems. Nonlinear least square problem is a special case of the nonlinear optimization problem defined in (A.1), where the objective function $\boldsymbol{r}(x)$ is the sum of squares. Nonlinear least square problem could be modeled as

$$\hat{\boldsymbol{x}}_{opt} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^n} \parallel \boldsymbol{v}(x) \parallel_2^2, \tag{A.8}$$

where $\boldsymbol{v}(x) : \mathbb{R}^n \to \mathbb{R}^m$ is a vector function of $\boldsymbol{x}$.

In the neighborhood of $\hat{\boldsymbol{x}}^l$, the function $\boldsymbol{v}(x)$ is approximated as

$$\boldsymbol{v}(x) \approx \boldsymbol{v}(\hat{\boldsymbol{x}}^l) + \boldsymbol{J}(\hat{\boldsymbol{x}}^l)\delta\boldsymbol{x}^l, \tag{A.9}$$

where $\boldsymbol{J}(\hat{\boldsymbol{x}}^l) \in \mathbb{R}^{m\times n}$ is the Jacobian matrix of $\boldsymbol{v}$ at $\hat{\boldsymbol{x}}^l$. To compute the Gauss-Newton step, substitute Eqn. (A.9) into (A.8), take derivative of (A.8) with respect to $\delta\boldsymbol{x}^l$ and set it equal to zero. We have

$$\boldsymbol{J}^\top\boldsymbol{J}\delta\boldsymbol{x}^l_{gn} = -\boldsymbol{J}^\top\boldsymbol{v} \quad \Rightarrow \quad \delta\boldsymbol{x}^l_{gn} = -(\boldsymbol{J}^\top\boldsymbol{J})^{-1}\boldsymbol{J}^\top\boldsymbol{v}, \tag{A.10}$$

where we simplify the notation by using $\boldsymbol{J}$ and $\boldsymbol{v}$ instead of $J(\hat{\boldsymbol{x}}^l)$ and $\boldsymbol{v}(\hat{x}^l)$, respectively. We also use $\delta\boldsymbol{x}^l_{gn}$ to emphasize the solution is Gauss-Newton step.

Appendix B

Deirvation of Encoder Model

B.1 Compute Velocity and Angular Rate From Encoder Measurements

Let $\tilde{e}_L$ and $\tilde{e}_R$ denote the outputs of the left and right encoders. They are discrete and quantized from the continuous valued encoder signal e_L and e_R. Assuming C is the encoder scale factor in units of counts per revolution and $(\phi_L(t), \phi_R(t))$ are the radian angular rotation from time t_0 to t. Then, the continuous encoder signal and angular rotation of the two wheels have the relation that

$$\begin{aligned} e_L(t) &= \frac{C}{2\pi}\phi_L(t) \\ e_R(t) &= \frac{C}{2\pi}\phi_R(t). \end{aligned} \tag{B.1}$$

The true outputs of encoder are the integer remainder resulting from dividing continuous valued encoder signal by scale factor C.

Let R_L and R_R denote the radius of left and right wheels, $s_L(t)$ and $s_R(t)$

denote the length that they have traveled up to time t. Then

$$\begin{aligned} \phi_L(t) &= \frac{1}{R_L} s_L(t) \\ \phi_R(t) &= \frac{1}{R_R} s_R(t). \end{aligned} \tag{B.2}$$

Substitute Eqn. (B.2) into Eqn. (B.1) and take the derivation with respect to t, then we have

$$\begin{aligned} \dot{e}_L(t) &= \frac{C}{2\pi R_L} \dot{s}_L(t) = \frac{C}{2\pi R_L} u_L \\ \dot{e}_R(t) &= \frac{C}{2\pi R_R} \dot{s}_R(t) = \frac{C}{2\pi R_R} u_R, \end{aligned} \tag{B.3}$$

where we use the speed of the two wheels $u_L(t)$ and $u_R(t)$ to replace $\dot{s}_L(t)$ and $\dot{s}_L(t)$, respectively.

Since the vehicle's body frame is centered at the middle of the axle that connects the two wheels and the heading is perpendicular to it, the velocity of the vehicle u and rotation rate ω have the following relations with u_L and u_R.

$$\begin{aligned} u &= \frac{1}{2} (u_L + u_R) \\ \omega &= \frac{1}{L} (u_L - u_R), \end{aligned} \tag{B.4}$$

where L is the length of the axle. From Eqn. (B.3) and Ean. (B.4), the velocity and rotation rate of vehicle and the outputs of encoder are related by

$$\begin{aligned} u &= \frac{\pi}{C} (R_L \dot{e}_L + R_R \dot{e}_R) \\ \omega &= \frac{2\pi}{CL} (R_L \dot{e}_L - R_R \dot{e}_R). \end{aligned} \tag{B.5}$$

B.2 State Transition Matrix

In 2D frame, the kinematic model is

$$\begin{aligned} \dot{n} &= cos(\psi)\frac{\pi}{C}\left(R_L\dot{e}_L + R_R\dot{e}_R\right) \\ \dot{e} &= sin(\psi)\frac{\pi}{C}\left(R_L\dot{e}_L + R_R\dot{e}_R\right) \\ \dot{\psi} &= \frac{2\pi}{CL}\left(R_L\dot{e}_L - R_R\dot{e}_R\right). \end{aligned} \tag{B.6}$$

The error propagation model is

$$\delta\dot{\boldsymbol{x}} = \boldsymbol{F}\delta\boldsymbol{x} + \boldsymbol{G}\nu, \tag{B.7}$$

where $\delta\boldsymbol{x} = \left[\begin{array}{c} \delta\dot{n}, \delta\dot{e}, \delta\dot{\psi}, \delta\dot{R}_L, \delta\dot{R}_R \end{array}\right]^T$, the matrix $\boldsymbol{F}$ is

$$\boldsymbol{F} = \begin{bmatrix} 0 & 0 & -\frac{\pi}{C}\left(\hat{R}_L\dot{\hat{e}}_L + \hat{R}_R\dot{\hat{e}}_R\right)sin(\hat{\psi}) & cos(\hat{\psi})\frac{\pi}{C}\dot{\hat{e}}_L & cos(\hat{\psi})\frac{\pi}{C}\dot{\hat{e}}_R \\ 0 & 0 & \frac{\pi}{C}\left(\hat{R}_L\dot{\hat{e}}_L + \hat{R}_R\dot{\hat{e}}_R\right)cos(\hat{\psi}) & sin(\hat{\psi})\frac{\pi}{C}\dot{\hat{e}}_L & sin(\hat{\psi})\frac{\pi}{C}\dot{\hat{e}}_R \\ 0 & 0 & 0 & \frac{2\pi}{CL}\dot{\hat{e}}_L & -\frac{2\pi}{CL}\dot{\hat{e}}_R \\ 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & 0 & \lambda \end{bmatrix},$$

the matrix **G** is

$$\begin{bmatrix} cos(\hat{\psi})\frac{\pi}{C}\hat{R}_L & cos(\hat{\psi})\frac{\pi}{C}\hat{R}_R & 0 & 0 \\ sin(\hat{\psi})\frac{\pi}{C}\hat{R}_L & sin(\hat{\psi})\frac{\pi}{C}\hat{R}_R & 0 & 0 \\ \frac{2\pi}{CL}\hat{R}_L & -\frac{2\pi}{CL}\hat{R}_R & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{B.8}$$

and $\nu = [\delta\dot{e}_L, \delta\dot{e}_R, \nu_L, \nu_R]^T$. The radius errors are added into the state vector, and they are modeled as scalar Gauss-Markov processes

$$\begin{aligned} \delta\dot{R}_L &= -\lambda\delta R_L + \nu_L \\ \delta\dot{R}_R &= -\lambda\delta R_R + \nu_R, \end{aligned} \tag{B.9}$$

where $\lambda > 0$, $E[\delta R_L(0)] = E[\delta R_L(0)] = 0$. The discrete error propagation model is

$$\delta \boldsymbol{x}_{k+1} = \Phi \delta \boldsymbol{x}_k + \omega_k, \tag{B.10}$$

where T is the time interval, $\Phi = e^{\boldsymbol{F}T}$ and $\omega_k = \int_{kT}^{(k+1)T} e^{\boldsymbol{F}t} \boldsymbol{G} \nu dt$.

Appendix C

Derivation of IMU Error Model

The IMU measures the acceleration $\tilde{\boldsymbol{a}}$ and angular rate $\tilde{\boldsymbol{\omega}}$, and they are modeled as

$$\tilde{\boldsymbol{a}} = \boldsymbol{C}({}^{I}_{G}\bar{\boldsymbol{q}})\left({}^{G}\boldsymbol{a} - {}^{G}\boldsymbol{g} + 2\lfloor\boldsymbol{\omega}_G\times\rfloor{}^{G}\boldsymbol{v}_I + \lfloor\boldsymbol{\omega}_G\times\rfloor^2\,{}^{G}\boldsymbol{p}_I\right) + \boldsymbol{b}_a + \boldsymbol{n}_a \tag{C.1}$$

$$\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega} + \boldsymbol{C}({}^{I}_{G}\bar{\boldsymbol{q}})\boldsymbol{\omega}_G + \boldsymbol{b}_g + \boldsymbol{n}_g \tag{C.2}$$

where ${}^{I}_{G}\bar{\boldsymbol{q}}$ is the quaternion that represents the rotation from global frame to inertial frame, and $\boldsymbol{C}({}^{I}_{G}\bar{\boldsymbol{q}})$ represents its corresponding rotation matrix. The angular rate of Earth Centered Earth Fixed (ECEF) frame rotation with respect to the inertial frame is represented by $\boldsymbol{\omega}_G$.

The Kinematics model is

$${}^{G}\dot{\boldsymbol{p}}_I = {}^{G}\boldsymbol{p}_I,\quad {}^{G}\dot{\boldsymbol{v}}_I = {}^{G}\boldsymbol{a}_I,\quad {}^{I}_{G}\dot{\bar{\boldsymbol{q}}} = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}){}^{I}_{G}\bar{\boldsymbol{q}},\quad \dot{\boldsymbol{b}}_a = \boldsymbol{n}_{\omega_a},\quad \dot{\boldsymbol{b}}_g = \boldsymbol{n}_{\omega_g}, \tag{C.3}$$

where

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} -\lfloor\boldsymbol{\omega}\times\rfloor & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^{\top} & \boldsymbol{0} \end{bmatrix},\quad \lfloor\boldsymbol{\omega}\times\rfloor = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \tag{C.4}$$

The state estimate propagates according to

$$ {}^{G}\dot{\hat{\boldsymbol{p}}}_I = {}^{G}\hat{\boldsymbol{v}}_I, \quad {}^{G}\dot{\hat{\boldsymbol{v}}}_I = {}^{G}\hat{\boldsymbol{a}}_I, \quad {}^{I}_{G}\dot{\hat{\bar{\boldsymbol{q}}}} = \frac{1}{2}\boldsymbol{\Omega}(\hat{\boldsymbol{\omega}}){}^{I}_{G}\hat{\bar{\boldsymbol{q}}}, \quad \dot{\hat{\boldsymbol{b}}}_a = \mathbf{0}, \quad \dot{\hat{\boldsymbol{b}}}_g = \mathbf{0}, \tag{C.5} $$

where

$$ {}^{G}\hat{\boldsymbol{a}}_I = \boldsymbol{C}({}^{G}_{I}\hat{\bar{\boldsymbol{q}}})(\tilde{\boldsymbol{a}} - \hat{\boldsymbol{b}}_a) + {}^{G}\boldsymbol{g} - 2\lfloor\boldsymbol{\omega}_G\times\rfloor{}^{G}\hat{\boldsymbol{v}}_I - \lfloor\boldsymbol{\omega}_G\times\rfloor^2\ {}^{G}\hat{\boldsymbol{p}}_I, \tag{C.6} $$

$$ \hat{\boldsymbol{\omega}} = \tilde{\boldsymbol{\omega}} - \boldsymbol{C}({}^{I}_{G}\hat{\bar{\boldsymbol{q}}})\boldsymbol{\omega}_G - \hat{\boldsymbol{b}}_g \tag{C.7} $$

Define the state error vector as

$$ \delta\boldsymbol{x} = \begin{bmatrix} \delta{}^{G}\boldsymbol{p}_I & \delta{}^{G}\boldsymbol{v}_I & \delta\boldsymbol{\theta} & \delta\boldsymbol{b}_a & \delta\boldsymbol{b}_g \end{bmatrix}^\top \in \mathbb{R}^{15\times 1}, \tag{C.8} $$

where $\delta\boldsymbol{\theta} \in \mathbb{R}^{3\times 1}$ represents the rotation from the estimated inertial frame to the true inertial frame. We also have

$$ {}^{I}_{I}\bar{\boldsymbol{q}} = {}^{I}_{G}\bar{\boldsymbol{q}} \otimes {}^{G}_{I}\hat{\bar{\boldsymbol{q}}} = \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\theta} \\ 1 \end{bmatrix}, \ \boldsymbol{C}({}^{I}_{I}\bar{\boldsymbol{q}}) = \boldsymbol{I} + \lfloor\delta\boldsymbol{\theta}\times\rfloor \tag{C.9} $$

The state error model is obtained from eqn. (C.5) and (C.3):

$$ \delta{}^{G}\dot{\boldsymbol{p}}_I = {}^{G}\dot{\boldsymbol{p}}_I - {}^{G}\dot{\hat{\boldsymbol{p}}}_I = {}^{G}\boldsymbol{v}_I - {}^{G}\hat{\boldsymbol{v}}_I = \delta{}^{G}\boldsymbol{v}_I \tag{C.10} $$

$$ \begin{aligned} \delta{}^{G}\dot{\boldsymbol{v}}_I &= {}^{G}\dot{\boldsymbol{v}}_I - {}^{G}\dot{\hat{\boldsymbol{v}}}_I = {}^{G}\boldsymbol{a}_I - {}^{G}\hat{\boldsymbol{a}}_I \\ &= \boldsymbol{C}({}^{G}_{I}\bar{\boldsymbol{q}})\,(\tilde{\boldsymbol{a}} - \boldsymbol{b}_a - \boldsymbol{n}_a) + {}^{G}\boldsymbol{g} - 2\lfloor\boldsymbol{\omega}_G\times\rfloor{}^{G}\boldsymbol{v}_I - \lfloor\boldsymbol{\omega}_G\times\rfloor^2\ {}^{G}\boldsymbol{p}_I \\ &\quad - \left[\boldsymbol{C}({}^{G}_{I}\hat{\bar{\boldsymbol{q}}})(\tilde{\boldsymbol{a}} - \hat{\boldsymbol{b}}_a) + {}^{G}\boldsymbol{g} - 2\lfloor\boldsymbol{\omega}_G\times\rfloor{}^{G}\hat{\boldsymbol{v}}_I - \lfloor\boldsymbol{\omega}_G\times\rfloor^2\ {}^{G}\hat{\boldsymbol{p}}_I\right] \\ &= \boldsymbol{C}({}^{G}_{I}\hat{\bar{\boldsymbol{q}}})\boldsymbol{C}({}^{I}_{I}\bar{\boldsymbol{q}})\,(\tilde{\boldsymbol{a}} - \boldsymbol{b}_a - \boldsymbol{n}_a) - \boldsymbol{C}({}^{G}_{I}\hat{\bar{\boldsymbol{q}}})(\tilde{\boldsymbol{a}} - \hat{\boldsymbol{b}}_a) \\ &\quad -2\lfloor\boldsymbol{\omega}_G\times\rfloor\delta{}^{G}\boldsymbol{v}_I - \lfloor\boldsymbol{\omega}_G\times\rfloor^2\delta{}^{G}\boldsymbol{p}_I \\ &= \boldsymbol{C}({}^{G}_{I}\hat{\bar{\boldsymbol{q}}})(\boldsymbol{I} + \lfloor\delta\boldsymbol{\theta}\times\rfloor)\,(\tilde{\boldsymbol{a}} - \boldsymbol{b}_a - \boldsymbol{n}_a) - \boldsymbol{C}({}^{G}_{I}\hat{\bar{\boldsymbol{q}}})(\tilde{\boldsymbol{a}} - \hat{\boldsymbol{b}}_a) \\ &\quad -2\lfloor\boldsymbol{\omega}_G\times\rfloor\delta{}^{G}\boldsymbol{v}_I - \lfloor\boldsymbol{\omega}_G\times\rfloor^2\delta{}^{G}\boldsymbol{p}_I \end{aligned} $$

$$
\begin{aligned}
&= -\boldsymbol{C}({}^{G\hat{}}_{I}\hat{\bar{q}})\delta\boldsymbol{b}_a - \boldsymbol{C}({}^{G\hat{}}_{I}\hat{\bar{q}})\boldsymbol{n}_a - \boldsymbol{C}({}^{G\hat{}}_{I}\hat{\bar{q}})\lfloor\delta\boldsymbol{\theta}\times\rfloor(\hat{\boldsymbol{b}}_a + \delta\boldsymbol{b}_a) \\
&\quad +\boldsymbol{C}({}^{G\hat{}}_{I}\hat{\bar{q}})\lfloor\delta\boldsymbol{\theta}\times\rfloor\tilde{\boldsymbol{a}} - 2\lfloor\boldsymbol{\omega}_G\times\rfloor\delta^G\boldsymbol{v}_I - \lfloor\boldsymbol{\omega}_G\times\rfloor^2\delta^G\boldsymbol{p}_I \\
&= -\lfloor\boldsymbol{\omega}_G\times\rfloor^2\delta^G\boldsymbol{p}_I - 2\lfloor\boldsymbol{\omega}_G\times\rfloor\delta^G\boldsymbol{v}_I - \boldsymbol{C}({}^{G\hat{}}_{I}\hat{\bar{q}})\lfloor(\tilde{\boldsymbol{a}} - \hat{\boldsymbol{b}}_a)\times\rfloor\delta\boldsymbol{\theta} \\
&\quad -\boldsymbol{C}({}^{G\hat{}}_{I}\hat{\bar{q}})\delta\boldsymbol{b}_a - \boldsymbol{C}({}^{G\hat{}}_{I}\hat{\bar{q}})\boldsymbol{n}_a. \qquad \text{(C.11)}
\end{aligned}
$$

To compute the rotation error model, we firstly evaluate the derivative of quaternion. From eqn. (C.9), we have

$$
\begin{aligned}
{}^{I\hat{}}_{I}\dot{\hat{\bar{q}}} &= {}^{I}_{G}\dot{\bar{q}} \otimes {}^{G\hat{}}_{I}\hat{\bar{q}} + {}^{I}_{G}\bar{q} \otimes {}^{G\hat{}}_{I}\dot{\hat{\bar{q}}} \\
&= \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}){}^{I}_{G}\bar{q} \otimes {}^{G\hat{}}_{I}\hat{\bar{q}} + {}^{I}_{G}\bar{q} \otimes {}^{G\hat{}}_{I}\dot{\hat{\bar{q}}} \otimes \left(-\frac{1}{2}\begin{bmatrix} \hat{\boldsymbol{\omega}} \\ 0 \end{bmatrix} \right) \\
&= \frac{1}{2}\begin{bmatrix} \boldsymbol{\omega} \\ 0 \end{bmatrix} \otimes {}^{I\hat{}}_{I}\hat{\bar{q}} - \frac{1}{2}{}^{I\hat{}}_{I}\hat{\bar{q}} \otimes \begin{bmatrix} \hat{\boldsymbol{\omega}} \\ 0 \end{bmatrix} \\
&= \frac{1}{2}\begin{bmatrix} -\lfloor\boldsymbol{\omega}\times\rfloor & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^\top & \mathbf{0} \end{bmatrix} {}^{I\hat{}}_{I}\hat{\bar{q}} - \frac{1}{2}\begin{bmatrix} \lfloor\hat{\boldsymbol{\omega}}\times\rfloor & \hat{\boldsymbol{\omega}} \\ -\hat{\boldsymbol{\omega}}^\top & \mathbf{0} \end{bmatrix} {}^{I\hat{}}_{I}\hat{\bar{q}} \\
&= \frac{1}{2}\begin{bmatrix} -\lfloor\boldsymbol{\omega}\times\rfloor - \lfloor\hat{\boldsymbol{\omega}}\times\rfloor & \boldsymbol{\omega} - \hat{\boldsymbol{\omega}} \\ -(\boldsymbol{\omega} - \hat{\boldsymbol{\omega}})^\top & \mathbf{0} \end{bmatrix} {}^{I\hat{}}_{I}\hat{\bar{q}} \\
&= \frac{1}{2}\begin{bmatrix} -\lfloor\boldsymbol{\omega}\times\rfloor - \lfloor\hat{\boldsymbol{\omega}}\times\rfloor & \boldsymbol{\omega} - \hat{\boldsymbol{\omega}} \\ -(\boldsymbol{\omega} - \hat{\boldsymbol{\omega}})^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\theta} \\ 1 \end{bmatrix}. \qquad \text{(C.12)}
\end{aligned}
$$

That is

$$
\begin{bmatrix} \frac{1}{2}\delta\dot{\boldsymbol{\theta}} \\ 0 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} -\lfloor\boldsymbol{\omega}\times\rfloor - \lfloor\hat{\boldsymbol{\omega}}\times\rfloor & \boldsymbol{\omega} - \hat{\boldsymbol{\omega}} \\ -(\boldsymbol{\omega} - \hat{\boldsymbol{\omega}})^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\theta} \\ 1 \end{bmatrix}, \qquad \text{(C.13)}
$$

so the rotation error evolves over time according to

$$\begin{aligned}
\delta\dot{\boldsymbol{\theta}} &= \frac{1}{2}\left(-\lfloor\boldsymbol{\omega}\times\rfloor - \lfloor\hat{\boldsymbol{\omega}}\times\rfloor\right)\delta\boldsymbol{\theta} + \boldsymbol{\omega} - \hat{\boldsymbol{\omega}} \\
&= \frac{1}{2}\left(-\lfloor(\boldsymbol{\omega} - \hat{\boldsymbol{\omega}} + \hat{\boldsymbol{\omega}})\times\rfloor - \lfloor\hat{\boldsymbol{\omega}}\times\rfloor\right)\delta\boldsymbol{\theta} + \tilde{\boldsymbol{\omega}} - \boldsymbol{C}(\overset{I}{G}\bar{q})\boldsymbol{\omega}_G - \boldsymbol{b}_g - \boldsymbol{n}_g \\
&\quad - \left(\tilde{\boldsymbol{\omega}} - \boldsymbol{C}(\overset{I}{G}\hat{\bar{q}})\boldsymbol{\omega}_G - \hat{\boldsymbol{b}}_g\right) \\
&= -\left(\lfloor\hat{\boldsymbol{\omega}}\times\rfloor + \lfloor\boldsymbol{C}(\overset{I}{G}\hat{\bar{q}})\boldsymbol{\omega}_G\times\rfloor\right)\delta\boldsymbol{\theta} - \delta\boldsymbol{b}_g - \boldsymbol{n}_g
\end{aligned} \tag{C.14}$$

The acceleration and angular rate measurement bias errors evolve over time according to

$$\delta\dot{\boldsymbol{b}}_a = \dot{\boldsymbol{b}}_a - \dot{\hat{\boldsymbol{b}}}_a = \boldsymbol{n}_{\omega_a} \tag{C.15}$$

$$\delta\dot{\boldsymbol{b}}_g = \dot{\boldsymbol{b}}_g - \dot{\hat{\boldsymbol{b}}}_g = \boldsymbol{n}_{\omega_g}. \tag{C.16}$$

The IMU state error can be written in the form

$$\delta\dot{\boldsymbol{x}} = \boldsymbol{F}\delta\boldsymbol{x} + \boldsymbol{G}\boldsymbol{n}, \tag{C.17}$$

where from the above analysis, the matrices and vector have the forms:

$$\boldsymbol{F} = \begin{bmatrix}
\boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\
-\lfloor\boldsymbol{\omega}_G\times\rfloor^2 & -2\lfloor\boldsymbol{\omega}_G\times\rfloor & -\boldsymbol{C}(\overset{G}{I}\hat{\bar{q}})\lfloor(\tilde{\boldsymbol{a}} - \hat{\boldsymbol{b}}_a)\times\rfloor & -\boldsymbol{C}(\overset{G}{I}\hat{\bar{q}}) & \boldsymbol{0}_{3\times3} \\
\boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \lfloor(\hat{\boldsymbol{\omega}} + \boldsymbol{C}(\overset{I}{G}\hat{\bar{q}})\boldsymbol{\omega}_G)\times\rfloor & \boldsymbol{0}_{3\times3} & -\boldsymbol{I}_{3\times3} \\
\boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\
\boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3}
\end{bmatrix} \tag{C.18}$$

$$\boldsymbol{G} = \begin{bmatrix}
\boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\
-\boldsymbol{C}(\overset{G}{I}\hat{\bar{q}}) & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\
\boldsymbol{0}_{3\times3} & -\boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\
\boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} \\
\boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3}
\end{bmatrix}, \quad \boldsymbol{n} = \begin{bmatrix}
\boldsymbol{n}_a \\ \boldsymbol{n}_g \\ \boldsymbol{n}_{b_a} \\ \boldsymbol{n}_{b_g}
\end{bmatrix}. \tag{C.19}$$

Appendix D

Camera Linearized Measurement Matrix

The camera measurement is modeled as

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{n} \begin{bmatrix} \frac{^{c}x}{^{c}z} \\ \frac{^{c}y}{^{c}z} \end{bmatrix} + \boldsymbol{n}, \quad {}^{c}\boldsymbol{p}_{cf} = \begin{bmatrix} {}^{c}x \\ {}^{c}y \\ {}^{c}z \end{bmatrix} \tag{D.1}$$

where ${}^{c}\boldsymbol{p}_{cf}$ is modeled as

$${}^{c}\boldsymbol{p}_{cf} = {}^{c}_{I}\boldsymbol{R}\left({}^{I}\boldsymbol{p}_{If} - {}^{I}\boldsymbol{p}_{Ic}\right) = {}^{c}_{I}\boldsymbol{R}\left({}^{I}_{G}\boldsymbol{R}\left({}^{G}\boldsymbol{p}_{f} - {}^{G}\boldsymbol{p}_{I}\right) - {}^{I}\boldsymbol{p}_{Ic}\right). \tag{D.2}$$

The camera measurement residual error is modeled as

$$\delta\boldsymbol{z} = \boldsymbol{H}\delta\boldsymbol{x} + \boldsymbol{R}. \tag{D.3}$$

The linearized measurement matrix can be computed by

$$\boldsymbol{H} = \boldsymbol{H}_1\boldsymbol{H}_2, \tag{D.4}$$

where

$$\boldsymbol{H}_1 = \frac{\partial \boldsymbol{h}}{\partial {}^c\boldsymbol{p}_{cf}} = \begin{bmatrix} \frac{1}{c_z} & 0 & -\frac{c_x}{c_z{}^2} \\ 0 & \frac{1}{c_z} & -\frac{c_y}{c_z{}^2} \end{bmatrix}. \tag{D.5}$$

To compute $\boldsymbol{H}_2$, we first evaluate the ${}^c\boldsymbol{p}_{cf}$ error:

$$\begin{aligned}
\delta {}^c\boldsymbol{p}_{cf} &= {}^c\boldsymbol{p}_{cf} - {}^c\hat{\boldsymbol{p}}_{cf} \\
&= {}^c_I\boldsymbol{R}\left[{}^I_G\boldsymbol{R}\left({}^G\boldsymbol{p}_f - {}^G\boldsymbol{p}_I\right) - {}^I\boldsymbol{p}_{Ic}\right] - {}^c_I\boldsymbol{R}\left[{}^I_G\hat{\boldsymbol{R}}\left({}^G\boldsymbol{p}_f - {}^G\hat{\boldsymbol{p}}_I\right) - {}^I\boldsymbol{p}_{Ic}\right] \\
&= {}^c_I\boldsymbol{R}\left[{}^I_I\boldsymbol{R}\,{}^I_G\hat{\boldsymbol{R}}\left({}^G\boldsymbol{p}_f - {}^G\boldsymbol{p}_I\right) - {}^I_G\hat{\boldsymbol{R}}\left({}^G\boldsymbol{p}_f - {}^G\hat{\boldsymbol{p}}_I\right)\right] \\
&= {}^c_I\boldsymbol{R}\left[(\boldsymbol{I} - \lfloor\delta\boldsymbol{\theta}\times\rfloor)\,{}^I_G\hat{\boldsymbol{R}}\left({}^G\boldsymbol{p}_f - {}^G\boldsymbol{p}_I\right) - {}^I_G\hat{\boldsymbol{R}}\left({}^G\boldsymbol{p}_f - {}^G\hat{\boldsymbol{p}}_I\right)\right] \\
&= {}^c_I\boldsymbol{R}\left[-{}^I_G\hat{\boldsymbol{R}}\,\delta {}^G\boldsymbol{p}_I - \lfloor\delta\boldsymbol{\theta}\times\rfloor {}^I_G\hat{\boldsymbol{R}}\left({}^G\boldsymbol{p}_f - {}^G\hat{\boldsymbol{p}}_I\right)\right] \\
&= -{}^c_G\hat{\boldsymbol{R}}\,\delta {}^G\boldsymbol{p}_I + {}^c_I\boldsymbol{R}\lfloor {}^I_G\hat{\boldsymbol{R}}\left({}^G\boldsymbol{p}_f - {}^G\hat{\boldsymbol{p}}_I\right)\times\rfloor\delta\boldsymbol{\theta}.
\end{aligned} \tag{D.6}$$

The matrix $\boldsymbol{H}_2$ is

$$\boldsymbol{H}_2 = \begin{bmatrix} -{}^c_G\hat{\boldsymbol{R}} & \boldsymbol{0}_{3\times3} & {}^c_I\boldsymbol{R}\lfloor {}^I_G\hat{\boldsymbol{R}}\left({}^G\boldsymbol{p}_f - {}^G\hat{\boldsymbol{p}}_I\right)\times\rfloor & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \end{bmatrix}. \tag{D.7}$$