ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

INDOOR SELF LOCALIZATION FOR MOBILE ROBOTS IN 2D ENVIRONMENT

M.Sc. THESIS

Hatice ERDOĞAN

Department of Control and Automation Engineering

Control and Automation Engineering Programme

JANUARY 2015

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

INDOOR SELF LOCALIZATION FOR MOBILE ROBOTS IN 2D ENVIRONMENT

M.Sc. THESIS

Hatice ERDOĞAN
(504111140)

Department of Control and Automation Engineering

Control and Automation Engineering Programme

Thesis Advisor: Prof. Dr. Hakan TEMELTAŞ

JANUARY 2015

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

İÇ ORTAMLARDA 2 BOYUTLU MOBİL ROBOT LOKALİZASYONU

YÜKSEK LİSANS TEZİ

Hatice ERDOĞAN
(504111140)

Kontrol ve Otomasyon Mühendisliği Anabilim Dalı

Kontrol ve Otomasyon Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Hakan TEMELTAŞ

OCAK 2015

Hatice Erdoğan, a **M.Sc.** student of ITU **Graduate School of Science Engineering and Technology,** student ID **504111140**, successfully defended the **thesis** entitled "**INDOOR SELF LOCALIZATION FOR MOBILE ROBOTS IN 2D ENVIORONMENT**", which she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Prof. Dr. Hakan TEMELTAŞ** ....
İstanbul Technical University

Jury Members : **Yrd.Doç.Dr. Gülay Öke Günel** ....
İstanbul Technical University

Yrd.Doç.Dr. Özgür Turay Kaymakçı ....
Yıldız Technical University

Date of Submission : 15 December 2014
Date of Defense : 22 January 2015

To my family,

FOREWORD

This thesis was written for my Master degree in Control and Automation Engineering Department at Istanbul Technical University. The aim of the thesis is to obtain the answer for a mobile robot "Where am I?"

I would like to thank my supervisor, Professor Hakan TEMELTAŞ, for his great help, guidance and encouragements during the development of this thesis.

I also want to thank my friends, Özgür ÖZKAN and Özen ÖZKAYA, for their willingness, practical supports, visions, and help.

Finally, I appreciate and love my family who support me every moment and being forever with me.

December 2014

Hatice ERDOĞAN
(Systems Engineer)

TABLE OF CONTENTS

Page

ABBREVIATIONS

2D	**:** 2 Dimensional
LMS 200	**:** Laser Measurement System 200
PC	**:** Personal computer
DC	**:** Direct current
D&A	**:** Distance and Angle
d_R	**:** Distance of reference points
d	**:** Distance
θ	**:** Rotation angle
T	**:** Translational

LIST OF TABLES

Page

LIST OF FIGURES

Page

INDOOR SELF LOCALIZATION FOR MOBILE ROBOTS IN 2D ENVIRONMENT

SUMMARY

Localization of an object is an important issue in order to achive navigation successfully. In this study, localization is used for a mobile robot navigation for indoor environment.

A mobile robot is used in a wide range and also it has lots of missions. The mobile robot has to know its current poisition in order to perform its assignments such as path following, reaching target positon, etc. So, localization is a paramount case for a mobile robot in order to carry out its assignments.

Localization with odometry technique is the most known in this area. Odometry is based on integration in order to find position parameters. So, this integration basis causes error parameter for each calculation. For each scan, the error paramater is added the previous error value and error is getting bigger with each environment. When the error increases, the difference between current position parameters and calculated position parameters also increase. Because of this, localization does not give reliable solution. In order to overcome this disadvantageous method, the necessity of new techniques occurs. One of the alternatives is scanning environment with a laser. To accomplish localization for mobile robots with laser scanning, position coordinates and rotation angle parameters are correlated during the process without any error parameters.

In this study, localization technique for a mobile robot is performed for 2D indoor environment. In order to carry out this study, laser sensor is used to scan environment. When laser sensor scans environment and collect data, these data have to be transferred to the computer to process and apply localization procedure. The communation and transformation between laser sensor and computer is provided by serial interface RS422. In the computer, MATLAB programme is used to process data.

The laser sensor scans the environment with 180° scanning angle and 1° resoluion. Thus, at the end of one scan 181 data point is transferred to MATLAB. First of all, these 181 points' are extracted to features with split and merge algorithm. So, less number of features are obtained instead of points. This split and merge algorithm is applied for each scan data. At the result of split and merge algorithm, there are feature maps for each scan.

Features have to be associated between each consecutive two feature maps in order to obtain common fetaures. These common features provide calculating changes on the position in the next step, data relation. Data association is carried out by distance and angle algorithm. In this method, not for all features, only two chosen features for each maps are associated and the result is obtained via these features. At the end of this step, there is one couple random feature for the first map and another couple

feature in the second map which correspond to the features in the first map. In this way, associated features are obtained.

In the final step, it is known that which feature in the first map match which feature in the second map. Therefore, the changes between these associated features gives the changes for the position in the same time. These changes are deal with translational on x coordinate, translational on y coordinate and rotation. In this step first of all, it is assumed that the start point cartesian coordinates is known for the first map as a reference point (0,0) and also the rotation angle is 0. Thus, for the first consecutive scans, the difference is obtained according to this reference point assumption and calculated new position values. Then, for the second consecutive scan, the new position parameters is calculated with the previous position values.

Finally, in order to obtain rotation and translational paramaters between two featıure sets ‘MATLAB Estimate Geometric Transform’ function is used. This function matches point pairs. Thus, current position can be calculated with these parameters with konwledge of translation from the reference point and also its rotation according to the reference point.

İÇ ORTAMLARDA 2 BOYUTLU MOBİL ROBOT LOKALİZASYONU

ÖZET

Yön bulmayı başarılı bir şekilde gerçekleştirmek için bir objenin yerini belirleme (lokalizasyonu) önemli bir konudur. Bu tez çalışmasında lolalizasyon konusu mobil robotlar üzerinden ele alınmıştır. İç ortamlarda mobil robotlar için 2 boyutlu lokalizayon bilgisinin elde edilmesi çalışmaları gerçekleştirilmiştir.

Günümüzde mobil robotların kullanımı çok geniş bir alana yayılmış durumdadır. Mobil robotlar engellerden sakınarak hedefe ulaşma, belirli bir yol planını takip etme gibi görevleri yerine getirdiği alanlarda kullanılmaktadır. Mobil robotların kendilerine yüklenen bu görevleri gerçekleştirebilmeleri için nerede olduklarını bilmeleri gerekmektedir. Bu nedenle lokalizasyon hesaplamaları ile bir mobil robotun sürekli olarak 'Ben neredeyim?' sorusuna cevap aranmaktadır. Bu nedenlerden dolayı bir mobil robotun etkili bir şekilde kullanılabilmesi için lokalizasyon verileri öenmli bir yere sahip olmaktadır.

Lokalizasyon ile yer bulma birçok yöntem ile gerçekleştirilebilmektedir. Bu yöntemlerden en bilineni odometre ile lokalizasyon hesaplamaktır. Ancak odometre yönteminde mobil robot tekerlerinden alınan dönme bilgisi integral hesaplamalarında kullanarak mobil robotun pozisyon bilgisi hesaplandığından bu yönteme alternatif olacak yöntemler ihtiyacı ortaya çıkmıştır. Çünkü integral hesaplamalarında yer alan hata katsayısı her yeni hesaplamada bir önceki hata değerine eklenerek artmaktadır. Bu nedenle tarama sayısı arttıkça hata katsayısı da artacağından zamanla gerçek pozisyon bilgisi ve hesaplanan pozisyon bilgisi arasındaki fark artmaktadır. Giderek lokalizayon hesaplamasının gerçekliğini kaybetmesinden dolayı yeni yöntemlere ihtiyaç artmıştır. Bu ihtiyaca bir kaşılık lazer sensör ile çevrenin taranarak lokalizayon hesaplamalarının yapılmasıdır. Bu yöntem ile mobil robotlarda lokalizasyon herhangi bir hata paramatresi içermeden koordinat noktaları ve açı değeri ile ölçümlendirilmektedir.

Bu tez çalışmasında da mobil robotlar için iç ortamlarda 2 boyutlu lokalizyon tekniği ile ilgili çalışmalar yapılmıştır. Bu çalışma için çevreyi tarayan bir adet lazer sensör kullanılmıştır. Lazer sensör 0˚ ile 180˚ arasında saat yönünün tersinde tarama yaparak 8191 cm'ye kadar ölçüm alabilmektedir. Ölçüm çözünürlüğü 0.25˚, 0.5˚ ve 1˚ olarak seçilebilmektedir. Bu çalışmada lazer sensör ile 1˚'lik çözünürlükte tarama yapılmıştır.

Lazer sensörde toplanan tarama verilerinin işlenip sonuca ulaşılabilmesi için bu verilerin bilgisayara aktarılması gerekmektedir. Bu veri aktarımı için bilgisayar ve lazer sensör arasındaki iletişim seri haberleşme yoluyla sağlanmıştır. Seri haberleşme üzerinden bilgisayara aktarılan veriler MATLAB programı üzerinden işlenmiştir. Lazer sensörün sahip olduğu komut yapıları MATLAB üzerinden seri haberleşme ile lazer sensöe gönderilerek lazer sensörün istenilen ölçüm ayarları yapılmış sensörden istenilen komutun cevapları alınmıştır.

MATLAB ile lazer sensör bağlantısı kurulup çevre taramaları yapıldıktan sonra elde eidlen çevre verileri MATLAB veritabına kaydedilmiştir. Bu çalışma boyunca kaydedilen bu veriler üzerinden işlemler gerçekleştirilmiştir. Tüm çalışma boyunca farklı pozisyonlarda art arda yapılmış 3 tarama verisi üzerinden lokalizasyon adımları gerçekleştirilmiştir.

Bu tez çalışmasında lokalizasyon; öznitelik çıkarma, öznitelikleri eşleştirme ve öznitelikleri ilişkilendirme olmak üzere 3 adımda tamamlanmıştır.

Lokalizasyon için ilk adım öznitelik çıkarımı ile başlamaktadır. Lazer sensörün 180˚lik bir çevreyi 1˚lik çözünürlükle taraması sonucu taranan çevreye ait 181 nokta lazer sensörden MATLAB’e aktarılmaktadır. 181 nokta üzerinden hesaplamalar yapmak yerine noktalar taranan çevre bilgisi kaybedilmeyecek şekilde daha az sayıdaki öznitelikler aracılığıyla tanımlanmıştır. Öznitelikler taranan çevredeki köşelere karşılık gelmektedir. Böylece 181 nokta yerine 181 noktanın içerdiği köşeler üzerinden hesaplamalar yapılmıştır. Köşeleri belirleyerek öznitelikleri çıkarmak için Bölme ve Birleştirme (Split&Merge) yöntemi kullanılmıştır. Bu yöntem ile köşeleri belirlemek, adından da anlaşılacağı gibi 2 aşamada gerçekleştirilmektedir. Bölme aşamasında elde olan veri kümesine bölünecek nokta kümesi kalmayana kadar belirlenmiş bölme adımları uygulanarak köşeler bulunmaktadır. İkinci aşama olan birleştirme bölümünde ise bölme aşamasında bulunmuş olan köşelerden yanlış olanlar elenmektedir. Bölme ve Birleştirme yönteminin sonunda her bir çevre taraması için nopktalar yerine köşelerin ifade edildiği öznitelik haritaları elde edilmiştir.

Lokalizasyon için ikinci adım ilk adımda her bir tarama verisi için elde edilen özniteliklerin ardışıl iki tarama arasında eşleştirilmesini sağlamaktır. Öznitelik eşleştirmesi bulunan bütün öznitelikler yerine ilk taramada referans olarak rastgele seçilen iki öznitelik üzerinden gerçekleştirilmiştir. Bu adım aşamalı olarak gerçekleştirilerek öncelikle birince ve ikinci tarama arasında br eşleştirme yappılmakta sonrasında ise ikinci ve üçüncü tarama arasında eşleştirme yapılmaktadır. Eşleştirme işlemine başlamak için öncelikle ilk taramadan rastgele iki öznitelik seçilmektedir. Uzaklık ve açıya bağlı olarak uygulanan ‘Distance&Angle’ yöntemi ile rastgele seçilen bu iki özniteliğe ikinci taramada karşılık gelen öznitelikler bulunmaktadır. Aynı işlemler ikinci ve üçüncü taramada da tekrarlanarak ikinci tarama için rastgele 2 öznitelik seçilmiş ve bu özniteliklerin üçüncü taramadaki karşılıkları uzaklık ve açı hesaplamalarına göre bulunmuştur. Öznitelik eşleştirme işlemlerinin sonucunda ardışıl iki çevre taraması için birbirlerine karşılık gelen ikişer tane öznitelik elde edilmiştir.

Lokalizasyon için son adım olan üçüncü adımda bir önceki adımda eşleştirilmiş olan özniteliklerin birbirlerine göre değişimi ilişkilendirilerek pozisyonun başlangıç anından itibaren ne kadar değiştiği hesaplanmaktadır. Bu adımda ilk taramanın yapıldığı nokta referans nokta olarak kabul edilmiş ve koordinat noktaları (0,0) olarak alınmıştır. Böylece ardışıl yapılan ilk iki tarama için ilk taramanın koordinat bilgileri bilinirken ikinci taramada bu koordinat değerlerinin ne kadar değiştiği bulunmuştur. Değişimin ilişkilendirilmesinde MATLAB ‘Estimate Geometric Transform’ fonksiyonu kullanılmıştır. Bu fonksiyon ile iki farklı tarama haritasında eşleşen öznitelikler kullanılarak koordinat ekseni üzerindeki öteleme ve dönme miktarları elde edilmektedir. İlk iki tarama için elde edilen öteleme ve dönme miktarları referans değerler üzerine eklenerek yeni koordinat noktalarına ulaşılmıştır. İkinci ve üçüncü tarama içinde aynı ilişkilendirme yapılarak elde edilen öteleme ve dönme değerleri bir önceki ilişkilendirmede elde edilen koordinat değerleri üzerine eklenerek son pozisyon noktası bulunmuştur.

Sonuç olarak, lazer sensörden elde edilen nokta kümelerinden 3 adımda gerçekleştirilen lokalizasyon işlemleri sonucunda başlangıç noktası (0,0) olan pozisyonunun ne kadar ötelendiği ve döndürüldüğü bilgisi elde edilmiştir.

1. INTRODUCTION

Localization is a method to determine the position of something. In mobile robotics, localization means specifying the relative position and orientation of robots at the any environment. In order to reliable navigation, a mobile robot has to know its own position. So, localization is an important issue for dependable navigation.

In general, mobile robots are assigned as following a path, searching something or avoiding an obstacle. The mobile robot needs to know its current location in order to carry out these tasks. Due to necessity of this information, localization is an important issue for a mobile robot.

The aim of this thesis is to fulfill indoor environment localization with mobile robot navigation. It is assumed that mobile robot's position at time is 0 is known and while the mobile robot is moving, current position information is updated according to reference point positon.

There is one more than technique for localization. One of it is odometry. As in mentioned in **[1]**, odometry determines position with rotation information which is taken from wheels. This information is integrated and calculated position. Due to integration base, odometer includes cumulative errors. While errors are getting increase with every scan, position differences between actual and calculated is bigger in time which is named as drift condition. Because of the cumulative error disadvantage of the odometry method, the necessity of the new techniques without integration method occurs.

Scanning environment with a laser sensor is a method to obtain position information without error accumulation. Laser sensor scans environment consecutively and creates point clusters. Features extraction method is applied to clusters and obtained keystone points for localization information. Features can be chosen as point, line, plane,etc.

In this study, Laser Measurement System 200 (LMS 200) is used which scans environment in 2D. The sensor data is transferred to the MATLAB for processing. Due to 2D scanning, features are extracted from length of walls of the environment. For instance, the length of the walls in a corridor is a line for a laser sensor. Because of the 2D scan results the sensor scans the wall as one plane.

When laser sensor scans the line there is many points for one line. Instead of keeping all points, the points are processed and transferred to a line equation. In order to definite a line; kinematic state vector values are needed. After these values are calculated and saved and robot is moved, if there is any position differences an iteration algorithm is run in order to calculate actual robot position.

Kinematic state vectors include 3 values which are x, y coordinates and heading angle (Q). Eachsen neyden bahsedyodun update for x, y and Q values give current localization information. After the analyzing the robot's position, the following step is to control the obtained robot position is actually on its desired position. If it is no, a control method is needed to remove differences between desired and current position. Therefore, robot position stays in desired position with changing wheel's velocity according to information coming from the control feedback. The studies in this thesis are based on the paper which is referenced as [1].

According to literature surveys, in [2], two algorithms which are based on matching tangent lines defined on two scans and minimizing a distance function and iteratively establishes correspondences between points in the two scans and then solves the point-to-point least-squares problem to compute the relative pose are developed in order to indicate a range scan to a previous scan so as to compute relative robot positions an an unknown environment. In [3], two-dimensional laser range scans are used to estimate motions without a kinematic model of the robot and, moreover, without using odometry. Features extracted from each scan, a data association method used to simultaneous observation of features from two consecutive scans to determine correct features associations. Two different strategies are tested to estimate relative robot position using common features of two consecutive scans. In [4], a Split-and-Merge segmentation algorithm for line extraction in 2-D range I mages is proposed. A prototype-based fuzzy clustering algorithm in a split-and-merge framework is used with this study. In [5], a graph theoretic algorithm is mentioned for data association which is the process of relating features observed in the environment to features viewed previously or to features in a map.

In chapter 2, laser measurement sensor's model is described. Some technical description of the laser sensor and measurement principles are mentioned.

In chapter 3, laser sensor measurement data analysis is indicated. Communication structure between laser sensor and MATLAB is described.

In chapter 4, mobile robot properties is described with its mechanical and mathematical structure.

In chapter 5, geometrical fundemantals which are based on polar and cartesian representations are told and also euclidean distance measurement is mentioned.

In chapter 6, localization methods which are used in this study is described. Feature extarction, data asscociation and data relation steps are described.

In chapter 7, experimental studies with regard to localization methods are shown.

In chapter 8, thesis studies are concluded.

2. MODEL OF LASER RANGE MEASUREMENT SENSOR

The Laser Measurement System 200 (LMS 200), is based on a time-of-flight measurement principle [6]. The means of this type of measurement is that the laser sensor sends a laser pulse and the object reflects this pulse within the range of the sensor as pictured in Figure 2.1. The distance between object and sensor is calculated with the time that passing during sending and receiving pulse.

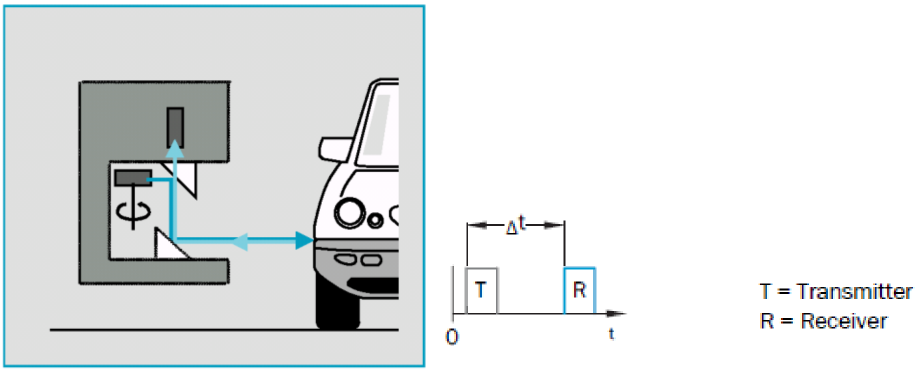


Figure 2.1 Measurement principle LMS

Objects are detected with regardless of their color and surfaces structure with time-of-flight measurement principle. Therefore, this technique serves as a more reliable object determination.

The LMS 200 is used in a wide area as in mentioned **[6]:**

- Determining the volumes of objects
- Determining the position of objects
- Controlling docking processes (positioning)
- Process automation
- Classification of objects (vehicle detection, camera trigger)
- Monitoring open sources for building security
- Determining the volumes or contours of bulk materials
- Collision prevention for vehicles or cranes
- Checking overhang/area monitoring in automated multi-storey car parks

In order to set the connection between LMS 200 and PC, the following components which are shown in Table 2.1 are required:

Table 2.1 Requirements of LMS 200 connection.

Item/Specification	**Description**
1 x LMS unit with cable connectors	LMS200-30106
1 x power supply and output signals	Power supply DC 24V/2.5A
1 x cable for power supply and output signals	Cable set: 1:5 meter length
1 x cable for data interface	Cable set 2:10 meter length

The laser sensor LMS 200 is shown in Figure 2.2.

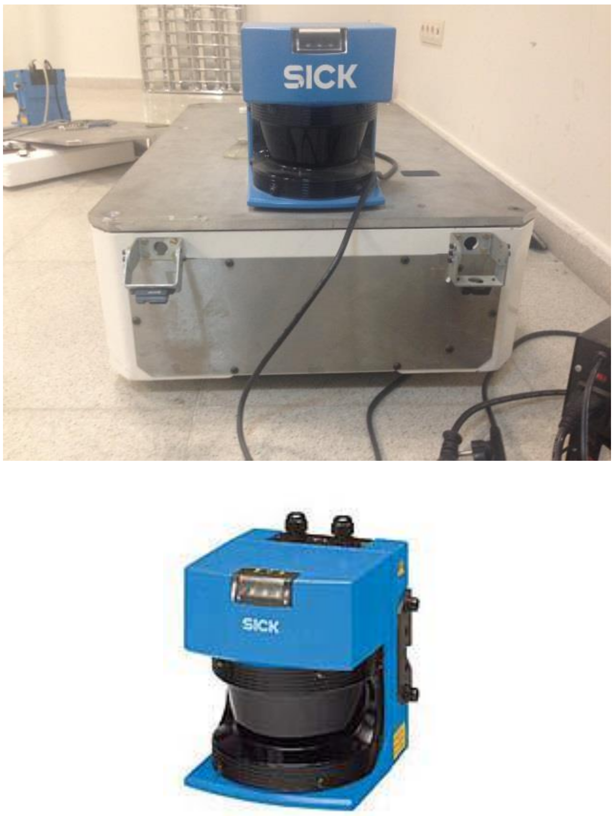
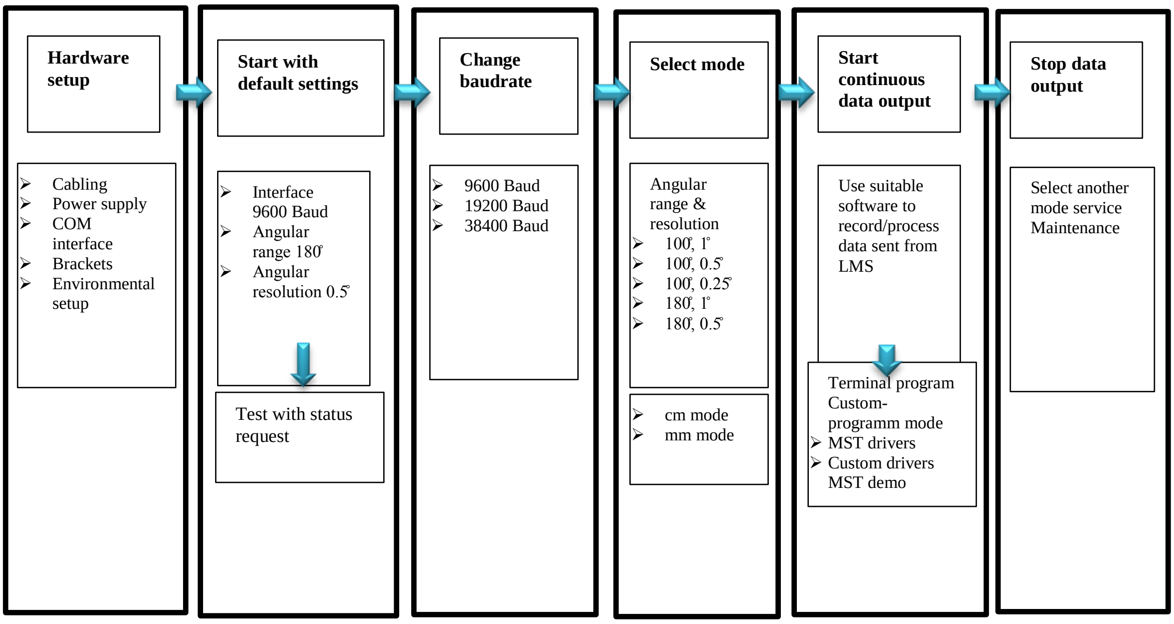


Figure 2.2SICK LMS 200

The computer and sensor is connected via serial interface. Schematic demonstration for LMS communication is as in Figure 2.3LMS communication scheme.Figure 2.3, as mentioned in **[6]**:

Figure 2.3LMS communication scheme.



The LMS200 scans counter clockwise direction every 0.25°, 0.5° or 1° in 100° or 180° measurement range as dependence of set values which are shown in Table 2.3.

Table 2.3 Angular resolution, scanning angle and number of measured values.

Angular resolution	0.25°	0.5°	1°
Maximum scanning angle*)	100°	180°	180°
Maximum number of measured values	401	361	181
*) symmetrical, from the middle			

In this study, LMS 200 scans counter clockwise direction every 1° in 180° measurement range. The picture about this measurement range and mesasurement scheme are shown in Figure 2.4 and Figure 2.5:

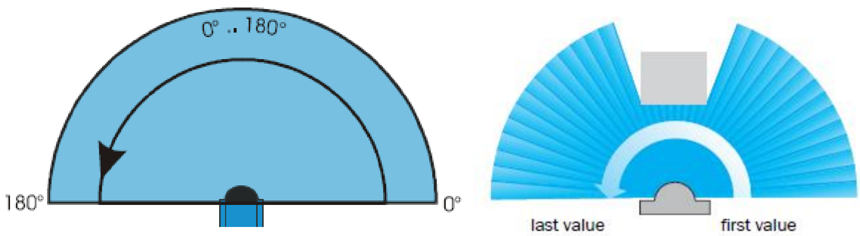


Figure 2.4 Measurement range 0°…180°

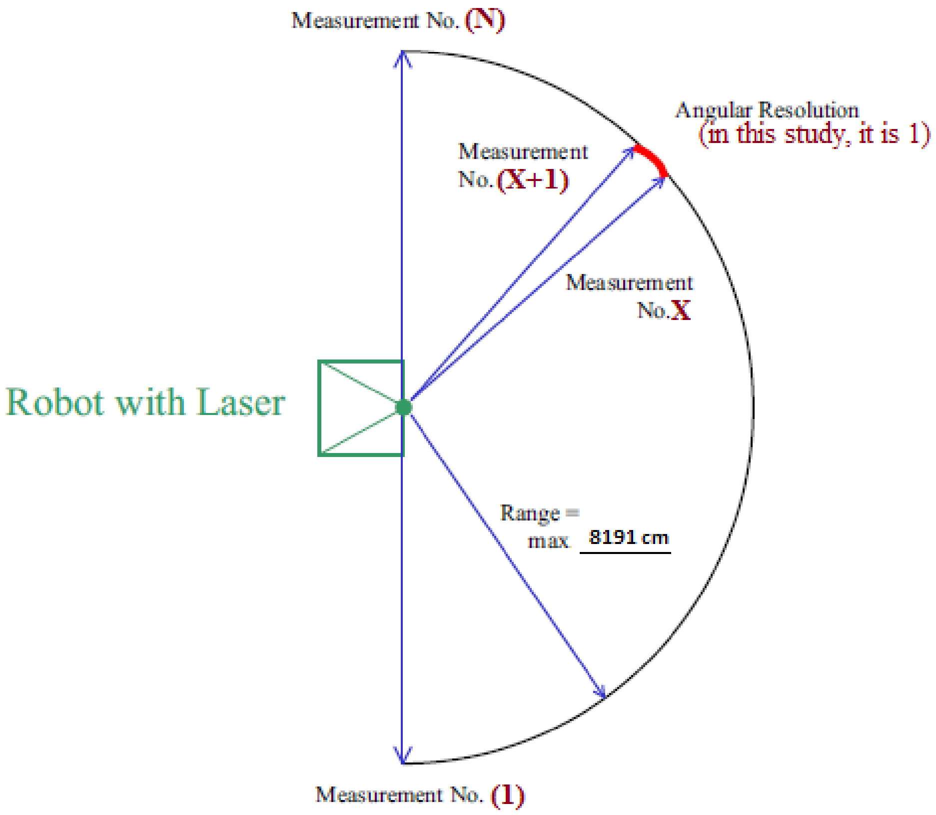


Figure 2.5 Laser sensor measurement scheme

In summary for this chapter is to analyze laser sensor with its physical and l models. Localization procedure which is told in next chapters are based on these laser sensor's abilities.

3. LOCALIZATION WITH LASER RANGE MEASUREMENT SENSOR

The scanned data with LMS200 is analyzed with MATLAB. Communication between MATLAB and laser sensor is provided by serial interface RS422.

After power connection of the LMS, the yellow and red indicators are active until the start-up procedure has been completed. When only either the green or the red led indicator is active, the unit is ready for communication [6].

The first step of using LMS 200 is to set scan procedure such as resolution, range, etc. The default values of LMS 200 are as in Table 3.1:

Table 3.1LMS 200 used values.

Parameter	**Value**
	LMS 200
Serial interface	9600 Baud 8 data bits No parity 1 stop bit No flow control
Angular range	0°…180°
Angular resolution	1°
Distance measurement mode	cm

To send to command LMS 200, the telegram structure is shown in below Table 3.2:

Table 3.2LMS 200 telegram structure.

	Frame				**Commands and data**		**Frame**	
Description	STX	Address	Length		Command/Response	Data	Checksum	
Byte position	1	2	3	4	5	6 to n	n+1	n+2

Some details about the format of the data stream are shown in below Table 3.3 and Table3.4:

Table 3.3 Output data string in byte unit.

STX	ADR	LenL Low byte	LenH High byte	CMD	Data LenL	Data LenH	Data 0° Low byte	Data 0° High byte	Data 1° Low byte	Data 1° High byte	… (more data)	Status	CRC Low byte	CRC High byte

Table 3.4 Description of output data string.

Designation	**Data size** (number of bits)	**Remarks**
STX	8	Start byte (STX=02)
ADR	8	Address of subscriber (in this case the PC) addressed. Typically, the value is 81
Len	16	Length of the total LMS output data string. Number of following output data string bytes excluding checksum (CRC=2 bytes)
CMD	8	Command byte, in this case (B0 hex), which is the command for continuous data output.
DataLen	16	Number of measurement data bytes (depending on measurement mode)
Data…	nx16	Measurement data values (2 bytes each) according to the measurement mode settings.
Status	8	Status byte Indication of system error, pollution etc. refer to telegram listing for exact information.
CRC	16	CRC Checksum

The LMS 200 is configured in the following order İn Table 3.5:

Table 3.5 Configuration process.

Step	Activity	Note
1	Switch on the LMS200 (power on)	-
2	The LMS200 sends a "power on" strings	During switch-on only
3	The command "switch to installation mode" is sent to the LMS200	Command 20h
4	The LMS200 responds with "Acknowledge".	06h
5	The LMS200 responds to the command.	Response A0h
6	The command to set the parameters is send to the LMS200.	Normally command 77h
7	The LMS200 responds with "Acknowledge".	06h
8	The LMS200 responds with "Parameters successfully changed".	Response F7h
9	The command "Switch to monitoring mode" is sent to the LMS200.	Command 20h
10	The LMS200 responds with "Acknowledge".	06h
11	The LMS200 responds with "Mode successfully changed".	Response A0h
12	Wait for the next request or start data transmission	Start the next action

In summary for this chapter is to give detailed information about communication between laser sensor and MATLAB. Via this given communication structure, laser sensor scan data is transferred to MATLAB.

4. DIFFERENTIAL DRIVE OF MOBILE ROBOT SYSTEM

4.1 Mathematical Representation in 2-Dimensional Space

The laser sensor is located on the mobile robot. In order to prevent hazardous effects on robot which occur with harmful surface obstacles that cannot be sensed by laser, four sharp infrared (IR) distance sensor are also located on the robot. In addition to these sensors, the mobile robot has two Maxon EC45 250W brushless DC motors with Maxon EPOS 70/10 drivers and two encoders (encoders directly communicate with the DSP via CAN bus), two batteries and Texas Instruments C2000 Series DSP.

The mobile robot has differential drive system. Thus, 2 independent controlled servo motor and their drivers are on the mobile robot. DC motors are preferred as servo motor because of DC motor's high torque production ability relative to their volumes. In addition to 2 wheels which provide differential drive, the mobile robot has 2 extra wheels in order to provide vehicle's balance and prevent tilting.

LMS 200 can scan only a certain level plane that changes according to mobile robots' structure due to 2D scan ability. Thus, any object that is under this level cannot be sensed by LMS 200.

Energy requirement of the mobile robot is provided by 2 batteries which have appropriate DC voltage. In addition to providing robot motion, DC-DC converter is used in order to charge DSP systems and other sensors and prevent asymmetric discharging between two batteries.

The mobile robot which structured as mentioned above is shown in Figure 4.1

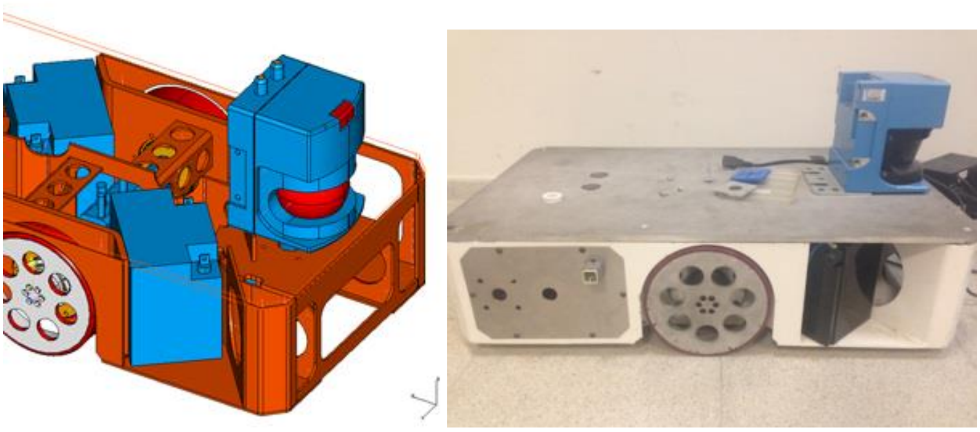


Figure 4.1CAD Drawing and photo of LMS 200 with a mobile robot

4.2 Mathematical Model of Mobile Robot

The assumptions that are used to obtain mathematical model of the mobile robot are as below:

1. Kinematic model based assumption
2. Dynamic model based assumption

In the kinematic model based assumption method, dynamic properties of the system behavior are ignored and it is assumed that robot velocity and servo motor values have minimum error range.

In the dynamic model based assumption method, whole system model can be obtained. Due to this fact that dynamic model based assumption has more advantages. In contrast, kinematic model study is easier than dynamic model studies.

4.2.1 Kinematic model

System's kinematic model is based on the following assumptions.

- Mobile robot moves on a planar line.
- Mobile robot's wheels are always perpendicular to ground plane.
- Mobile robot has no flexible parts.
- Mobile robot moves around center of rotation transiently.

- Mobile robot is assumed as a solid body. There is no moving part out of wheels.

Configuration equation of the mobile robot and its configuration space equation are assumed as below in equation (4.1):

$$q = \begin{bmatrix} q1 \\ q2 \\ . \\ . \\ qn \end{bmatrix} \epsilon Q \tag{4.1}$$

Mobile robots are characterized by its velocity constraints which cannot be integrated. In general, these constraints arise from their movement assumptions which are rolling without sliding and driving without skidding. The constraints can be defined as follows in equation (4.2):

$$a_j = (q)\dot{q} = 0 \tag{4.2}$$

As it is seen in the constraint equation (4.2), it includes velocity in addition to position. These types of constraints are named as nonholonomic constraints. Mechanical systems with wheels can be named as nonholonomic system because of having nonholonomic constraints.

It is assumed that mobile robot system has r number of constraints. When these constraints are written as matrix form, it is named as constraint matrix and it is shown in equation (4.3) and equation (4.4).

$$A(q) = \begin{bmatrix} \propto_1 (q) \\ \propto_2 (q) \\ . \\ . \\ \propto_r (q) \end{bmatrix} \tag{4.3}$$

$$A(q)\dot{q} = 0 \tag{4.4}$$

Kinematic model block diagram of the mobile robot is shown in the Figure 4.2:

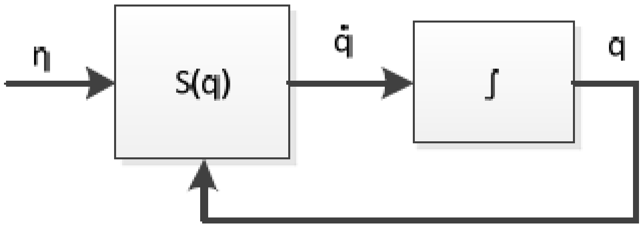


Figure 4.2 Kinematic model block diagram of the mobile robot

Mathematical equation of this block diagram is as follows in equation (4.5):

$$\dot{q} = S(q)\eta \qquad \textbf{(4.5)}$$

$\eta \epsilon R^m$: *Input vector which comprises of mobile robot's velocity that is defined in the local plane*

$S(q) \epsilon R^{nxm}$: *Transformation matrix which is between mobile robot's velocity vector and defined in general plane velocity vector.*

As it is seen in the equation 4.5, robot's local plane velocity, general plane velocity or mobile robot's configuration vector can be chosen as an output. According to control method convenience, velocity vector which is defined in the robot's local plane is chosen as control input vector. Thus, kinematic model of the mobile robot can be shown as in the equation (4.6).

$$\dot{q} = \sum_{i-1}^{m} u_i s_i(q) \qquad \textbf{(4.6)}$$

u_i: *Control inputs*

s_i: *Vector plane that set the direction of the control inputs*

When they are written as the matrix form, equation (4.7) and equation (4.8) are obtained:

$$S(q) = [s_1(q)s_2(q)\ldots\ldots s_m(q)]nxm_2 \quad \textbf{(4.7)}$$

$$\eta = \begin{bmatrix} u_1 \\ u_2 \\ \cdot \\ \cdot \\ u_m \end{bmatrix} \quad \textbf{(4.8)}$$

s_i is orthogonal to the $\propto_i$, due to this relation the following equations (4.9) and (4.10) can be obtained:

$$s_i^T \propto_i^T = 0 \quad \textbf{(4.9)}$$

$$S^T A^T = 0 \quad \textbf{(4.10)}$$

4.2.1.1 Kinematic model of differential drive mobile robot

The kinematic model owns to mobile robot which is used in this study is given in this section. Relative to this kinematic model graph is shown in Figure 4.3:

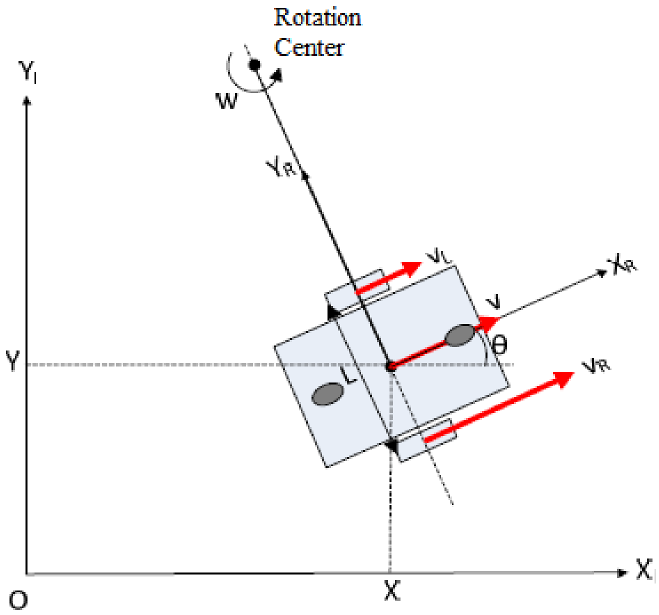


Figure 4.3 Kinematic model graph of the mobile robot

Configuration vector of this mobile robot is as in the equation (4.11):

$$q = [X \quad Y \quad Q]T \qquad \textbf{(4.11)}$$

X and Y: *Mobile robot position of the general plane*

Q: *Mobile robot orientation of the general plane*

When kinematic model equations which are shown previous chapter are updated according to differential drive mobile robot, the following equations (4.12), (4.13), (4.14), and (4.15) are obtained:

$$\dot{q} = S(q)\eta \qquad \textbf{(4.12)}$$

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Q} \end{bmatrix} = \begin{bmatrix} \cos Q & 0 \\ \sin Q & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \qquad \textbf{(4.13)}$$

$$\eta = \begin{bmatrix} v \\ w \end{bmatrix} \qquad \textbf{(4.14)}$$

$$S(q) = \begin{bmatrix} \cos Q & 0 \\ \sin Q & 0 \\ 0 & 1 \end{bmatrix} \qquad \textbf{(4.15)}$$

4.2.2 Dynamic model

System’s dynamic model is based on the following assumptions.

- Mobile robot moves on 2D plane
- Center of mobile robot is on the axis that is passing through center of rotation and distance from the center of rotation is d.
- Mobile robot has identical wheels and their radiuses’ are same.
- The wheels always touch the ground and their touch point is only one.

- The wheels are always perpendicular to ground plane.
- For each wheel, the forces that are perpendicular to the ground are constant.
- The wheels' energies are neglected.

Configuration equation of the mobile robot and its configuration space equation are assumed as in equation (4.16):

$$q = \begin{bmatrix} q1 \\ q2 \\ . \\ . \\ qn \end{bmatrix} \epsilon Q\text{: configuration vector (joint variable vector)} \qquad \textbf{(4.16)}$$

Matrix for of this dynamic model is shown as the following equation (4.17):

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + R(\dot{q}) + \tau_d = B(q)\tau - A^T(q)\lambda \qquad \textbf{(4.17)}$$

$M(q)\epsilon R^{nxn}$: *Symmetric and positive mass matrix*

$C(q,\dot{q})\epsilon R^{nxn}$: *Centripedal matrix*

$R(\dot{q})\epsilon R^{n}$: *Friction force matrix*

$\tau_d \epsilon R^{n}$: *Distortion matrix which includes modeled system dynamics*

$B(q)\epsilon R^{nxp}$: *System torque transition matrix*

$\tau \epsilon R^{n}$: *System torque vector*

$A(q)\epsilon R^{Ixn}$: *System boundary matrix*

$\lambda \epsilon R^{n}$: *Boundary force vector*

In this study, in order to obtain dynamic model of mobile robot Euler-Lagrange method is used.

Euler-Lagrange method is based on system's kinetic and potential energy. Kinetic energy and potential energy equations are shown in equation (4.18) and (4.19), respectively.

$$L = KE - PE \quad \textbf{(4.18)}$$

$$\frac{d}{dx}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q_i + \varphi_i \quad \textbf{(4.19)}$$

q_i : $ith\ element\ of\ configuration\ vector$

Q_i: $ith\ element\ of force$

v_i, w_i: $linear\ and\ angular\ velocvity$

F: $force\ on\ the\ center\ mass$

M: $moment\ on\ the\ center\ mass$

φ_i: $ith\ force\ of\ constraint$

A: $constraint\ matrix$

λ: $lagrange\ constant$

ith element of force and ith force of constraint are shown in equation (4.20) and equation (4.21), respectively.

$$Q_i = F\frac{\partial v_i}{\partial \dot{q}_i} + M\frac{\partial w_i}{\partial \dot{q}_i} \quad \textbf{(4.20)}$$

$$\varphi = A^T \lambda \quad \textbf{(4.21)}$$

4.2.2.1 Dynamic model of differential drive mobile robot

The dynamic model owns to mobile robot which is used in this study. Relative to this kinematic model graph is shown in Figure 4.4:

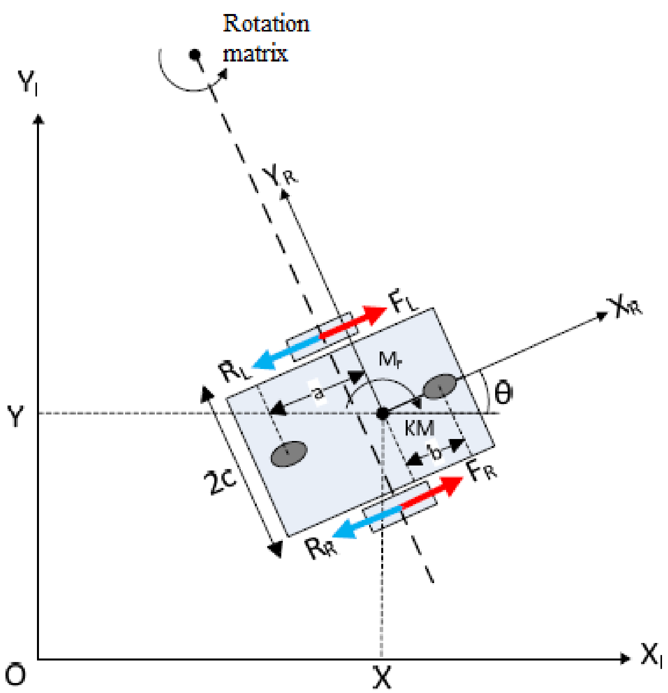


Figure 4.4 Dynamic model graph of the mobile robot

$(O_1X_1Y_1)$: *General coordinate system*

$(O_RX_RY_R)$: *Local coordinate system which is defined on mobile robot*

R_L: *Friction force between left wheel and ground*

R_R: *Friction force between left wheel and ground*

F_L: *Drive force to the left wheel*

F_R: *Drive force to the right wheel*

$2c$: *Perpendicular distance between left and right wheel (it is assumed that center of mass has equal distance to each wheel)*

d: *Perpendicular distance between center of rotation and center of mass*

M: *moment*

M_r: *resistance moment*

Energy equation is shown in equation (4.22):

$$KE = \frac{1}{2} mV^T V + \frac{1}{2} I w^2 \quad \textbf{(4.22)}$$

$PE = constant$

$V: linear\ veloctiy\ vector$

$w: angular\ velocity\ vector$

$m: total\ mass\ of\ the\ mobile\ robot$

$I: inertia\ tensor$

It is assumed that the mobile robot moves on a smooth plane. So, potential energy of the mobile robot is constant during movement. Due to this, potential energy has no effect on the dynamic model. The new equations without potential energy effect is formed as in equations (4.23), (4.24), and (4.25).

$$V = \begin{bmatrix} \dot{V} \\ \dot{Y} \end{bmatrix} \quad \textbf{(4.23)}$$

$$V^T V = \dot{X}^2 + \dot{Y}^2 \quad \textbf{(4.24)}$$

$$w = \dot{Q} \quad \textbf{(4.25)}$$

Lagrange equation is as equation in (4.26)

$$L = \frac{1}{2} m(\dot{X}^2 + \dot{Y}^2) + \frac{1}{2} I \dot{Q}^2 - PE \quad \textbf{(4.26)}$$

Euler Lagrange movement equations are shown in equations (4.27), (4.28), (4.29), (4.30), (4.31), (4.32), (4.33), and (4.34).

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{X}}\right) - \frac{\partial L}{\partial X} = F_X - R_X - \varphi_X \tag{4.27}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{Y}}\right) - \frac{\partial L}{\partial Y} = F_Y - R_Y - \varphi_Y \tag{4.28}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{Q}}\right) - \frac{\partial L}{\partial Q} = M - M_r - \varphi_Q \tag{4.29}$$

$$F_X = (F_L + F_R)(cosQ) \tag{4.30}$$

$$F_Y = (F_L + F_R)(sinQ) \tag{4.31}$$

$$M = (-F_L + F_R)\ c \tag{4.32}$$

$$\tau_L = \frac{F_L}{r} \tag{4.33}$$

$$\tau_R = \frac{F_R}{r} \tag{4.34}$$

r: *Radius of wheel (each wheel of mobile robot is identical and has same radius)*

Correspond to r parameter, new equations (4.35), (4.36), (4.37), (4.38), and (4.39) are obtained.

$$R_X = (R_L + R_R)(cosQ) \tag{4.35}$$

$$R_Y = (R_L + R_R)(sinQ) \tag{4.36}$$

$$M = (-R_L + R_R)\, c \tag{4.37}$$

$$R_L = \mu_L N_L sgn(V_L) \tag{4.38}$$

$$R_R = \mu_R N_R sgn(V_R) \tag{4.39}$$

N_L: $perpendicular\ left\ force\ to\ the\ ground$

N_R : $perpendicular\ right\ force\ to\ the\ ground$

g: $gravity$

V_L: $linear\ velocity\ of\ left\ wheel$

V_R: $linear\ velocity\ of\ right\ wheel$

$$N_L = N_R = \frac{mg}{2}\frac{b}{(b+d)} \tag{4.40}$$

$$V_L = V_X - c\dot{Q} \tag{4.41}$$

$$V_R = V_X + c\dot{Q} \tag{4.42}$$

V_X, velocity

$$V_X = \dot{X}\, cosQ + \dot{Y}\, sinQ \tag{4.43}$$

When equations (4.27), (4.28), (4.29) and equations (4.40), (4.41), (4.42) and (4.43) is used and written as matrix form, the dynamic model of the differential drive mobile robot is shown in equation (4.44).

$$M(q)\ddot{q} + C + R(\dot{q}) + \tau_d = B(q)\tau - A^T\lambda \tag{4.44}$$

Open form of the equation (4.44) is shown in equation (4.45) and (4.46).

$$M(q) = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix}, R(\dot{q}) = \begin{bmatrix} R_X \\ R_Y \\ M_r \end{bmatrix}, B(q) = \frac{1}{2}\begin{bmatrix} cos\ Q & \cos Q \\ \sin Q & \sin Q \\ -c & c \end{bmatrix}, \quad \tau = \begin{bmatrix} \tau_L \\ \tau_R \end{bmatrix} \tag{4.45}$$

$$A(q) = [-sinQ\ cosQ - d] \tag{4.46}$$

λ: *Constraint force*

τ_d*: Distortion vector*

Derive equation (4.10) to time, equation (4.47) is obtained.

$$\ddot{q} = S(q)\dot{\eta} + \dot{S}(q)\eta \tag{4.47}$$

The parameters in equation (4.47) is shown in equation (4.48) with details.

$$q = \begin{bmatrix} X \\ Y \\ Q \end{bmatrix}, S(q) = \begin{bmatrix} cos\ Q & -\mathrm{d}\cos Q \\ \sin Q & \mathrm{d}\sin Q \\ 0 & 1 \end{bmatrix}, \eta = \begin{bmatrix} V \\ w \end{bmatrix} \tag{4.48}$$

When equation (4.47) is substituted into equation (4.44) and multiplied by S^T for two side, equation (4.49) is obtained.

$$S^T MS\dot{\eta} + S^T M\dot{S}\eta + S^T R + S^T \tau_d = S^T B\, \tau - S^T A^T\, \lambda \tag{4.49}$$

When equation (4.49) is written in simpler form equation (4.50) is obtained.

$$\overline{M}\dot{\eta} + \overline{C}\eta + \overline{R} + \overline{\tau_d} = \overline{B}\, \tau \tag{4.50}$$

The open forms of matrices and vectors in equation (4.50) are given in equations (4.51), (4.52), (4.53), (4.54), and (4.55).

$$\overline{M} = S^T M S \tag{4.51}$$

$$\overline{C} = S^T M \dot{S} \tag{4.52}$$

$$\overline{R} = S^T R \tag{4.53}$$

$$\overline{\tau_d} = S^T \tau_d \tag{4.54}$$

$$\overline{B} = S^T B \tag{4.55}$$

In summary for this chapter is to analyze dynamic and kinematic models of mobil robot. The aim of this chapter is to show the dynamic and kinematic models of the mobile robot.

5. GEOMETRICAL FUNDAMENTALS

In this study, laser scans the environment and obtains 181 point because of the 0° to 180° scan range with 1° resolution. These points which is summarized in Figure 2.4 and Figure 2.5 are plotted on MATLAB with distance and angle parameters, firstly. This plot is the form of polar representation. In order to extract features easily, the plot converts to cartesian representation and obtained x, y coordinate values.

5.1 Mathematical Representation in 2-Dimensional Space

In order to define geometrical objects two methods are applied:

1. Cartesian Representation
2. Polar Representation

By the help of the transformations, geometrical objects can be transformed one space to another space.

5.1.1 Cartesian representation

There are two parameters to define the coordinates in cartesian representation. These two parameters are the length of X- axis and Y-axis which are represented as (x, y), respectively. The two parameters are represented as in equation (5.1)

$$P_{cartesian} = \begin{pmatrix} x \\ y \end{pmatrix} \quad \textbf{(5.1)}$$

$P_{cartesian}: cartesian\ coordinate\ points$
$x : length\ of\ X - axis$
$y : length\ of\ Y - axis$

The reference point on the cartesian space is represent as origin as in equation (5.2).

$$O = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \textbf{(5.2)}$$

5.1.2 Polar representation

There are two parameters to define the coordinates in polar representation. These two parameters are distance parameter and angle parameter which are respresented as in equation (5.3).

$$P_{polar} = \begin{pmatrix} \rho \\ \theta \end{pmatrix} \quad \textbf{(5.3)}$$

P_{polar}: *polar coordinate points*

ρ: *distance parameter which is distance between origin and coordinate*

θ: *angle parameter which is angle from x axis counter clockwise to the coordinate*

The representations of polar and cartesian forms are shown in Figure 5.1.

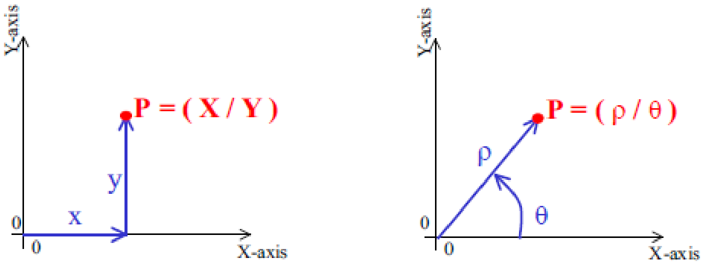


Figure 5.1 Cartesian and Polar Representation

The coordinate transformation from the polar representation to the cartesian representation is as in equation (5.4):

$F: R^{polar} \rightarrow R^{cartesian}$

$$C_{polar} = \begin{pmatrix} \rho \\ \theta \end{pmatrix} \equiv C_{cartesian} = \begin{pmatrix} \rho \cos Q \\ \rho \sin Q \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} \quad \textbf{(5.4)}$$

The coordinate transformation from the cartesian representation to the polar representation is as in equation (5.5):

$F: R^{cartesian} \rightarrow R^{polar}$

$$C_{cartesian} = \begin{pmatrix} x \\ y \end{pmatrix} \equiv C_{polar} = \begin{pmatrix} \rho \\ \theta \end{pmatrix} = \begin{pmatrix} \sqrt{x^2 + y^2} \\ \sin^{-1}\begin{pmatrix} y \\ \rho \end{pmatrix} \end{pmatrix} = \begin{pmatrix} \sqrt{x^2 + y^2} \\ \cos^{-1}\begin{pmatrix} x \\ \rho \end{pmatrix} \end{pmatrix} \quad \textbf{(5.5)}$$

According to above information, the point representation of a geometrical object will be as in equation (5.6) and (5.7):

$$P_{cartesian} = \binom{x}{y}\text{ , point representation in cartesian coordinate} \quad \textbf{(5.6)}$$

$$P_{polar} = \binom{\rho}{\theta}\text{, point representation in polar coordinate} \quad \textbf{(5.7)}$$

According to above information, the straight line representation of a geometrical object will be as follow:

Cartesian representation is shown in equation (5.8):

$$y = mx + c \quad \textbf{(5.8)}$$

$m: line\ slope$

$c: y\ axis\ offset$

Line parameters of two line points are as in equation (5.11) which is obtained by equations (5.9) and equation (5.10):

$$P_n = \binom{x_n}{y_n} \quad \textbf{(5.9)}$$

$$P_m = \binom{x_m}{y_m} \quad \textbf{(5.10)}$$

with $\Delta x = x_n - x_m$

$\Delta y = y_n - y_m$

$$L = \binom{m}{c} = \begin{pmatrix} \frac{\Delta y}{\Delta x} \\ y_n - \left(x_n \frac{\Delta y}{\Delta x}\right) \end{pmatrix} = \begin{pmatrix} \frac{\Delta y}{\Delta x} \\ y_m - \left(x_m \frac{\Delta y}{\Delta x}\right) \end{pmatrix}\ with\ \Delta x \neq 0 \quad \textbf{(5.11)}$$

Polar representation is shown in equation (5.12) :

$$\rho = x\cos(\theta) + y\sin(\theta) \quad \textbf{(5.12)}$$

ρ : *orthogonal distance to line from origin*

θ: *counter clockwise angle between* ρ *and x axis*

Line parameters of two line points are as in equation (5.15) which is obtained by equations (5.13) and (5.14):

$$P_n = \binom{x_n}{y_n} \quad \textbf{(5.13)}$$

$$P_m = \binom{x_m}{y_m} \quad \textbf{(5.14)}$$

with $\Delta x = x_n - x_m$

$\Delta y = y_n - y_m$

$$L = \binom{\rho}{\theta} = \begin{pmatrix} x_n \cos(\tan^{-1}(\frac{\Delta y}{\Delta x})) + y_n \sin(\tan^{-1}(\frac{\Delta y}{\Delta x})) \\ \tan^{-1}(\frac{\Delta y}{\Delta x}) \end{pmatrix} = \begin{pmatrix} x_m \cos(\tan^{-1}(\frac{\Delta y}{\Delta x})) + y_m \sin(\tan^{-1}(\frac{\Delta y}{\Delta x})) \\ \tan^{-1}(\frac{\Delta y}{\Delta x}) \end{pmatrix} \text{ with } \theta \in]-\frac{\pi}{2}; +\frac{\pi}{2}] \text{ and } \rho \in \quad \textbf{(5.15)}$$

Figure 5.2 shows the differences between carteasian and polar line representation.

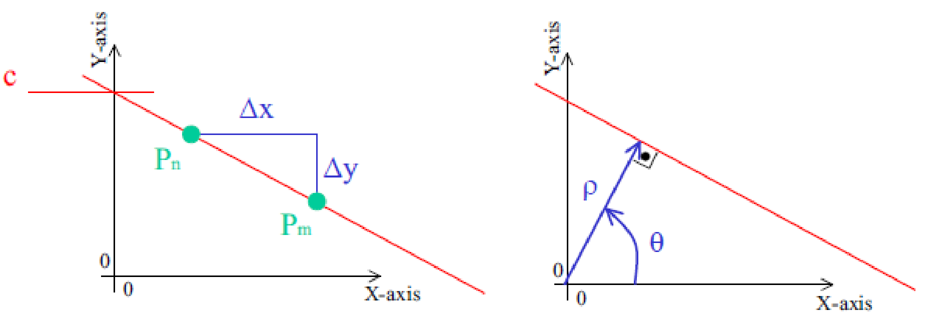


Figure 5.2 Cartesian and Polar Line Representation

5.2 Distance Measurement

Distance between a point and a line is an important issue to decide is the point is inside of the data point group or not. In this study, distance measurement is performed by Euclidean distance formula which is based on perpendicular distance between two cartesian points.

5.2.1Euclidean distance

The Euclidean distance formulas between point to point and point to line are given below:

5.2.1.1 Point to point distance

The distance between two points which are P_m and P_n is calculated as in the equation (5.16):

$$d = \sqrt{(x_n - x_m)^2 + (y_n - y_m)^2} \tag{5.16}$$

with

$P_n = \binom{x_n}{y_n}$ and $P_m = \binom{x_m}{y_m}$

5.2.1.2 Point to line distance

The line is between P_k and P_l and the distance from P_0 to this line is calculated as in the equation (5.17):

$$d = \frac{|(x_l - x_k)(y_k - y_0) - (y_l - y_k)(x_k - x_0)|}{\sqrt{(x_l - x_k)^2 + (y_l - y_k)^2}} \tag{5.17}$$

with

$P_k = \binom{x_k}{y_k}$, $P_l = \binom{x_l}{y_l}$ and $P_0 = \binom{x_0}{y_0}$

In summary for this chapter is to analyze polar and cartesian representation. Laser sensor gives data as polar form and adhering to above polar and carteasian representation information, data form transformations are carried out . Also, to compare and analyze data points, distance measurement is used with euclidean distance formulas.

6. LOCALIZATION METHOD FOR INDOOR ENVIRONMENT

In this study, localization method is based on three steps. First of all, the features which are obtained from each scan are extracted. Then, consecutive two scans' extracted features are associated in order to observe change. After all, roto-translation is found in order to obtain new location. As mentioned in, [7] and [8] studies, this method steps are as following [1]:

1. Extract features within each scan obtaining the actual features map.
2. Data Association, in other words, search for pair wise corresponding features from both features maps.
3. Calculate the roto-translation between two consecutive scans.
4. Update the robot pose.
5. Save the current feature map for the following iteration.

6.1 Feature Extraction

Feature extraction concerns reducing the points required to describe a large set of data. There are different methods for feature extraction. These methods are described in [9], in detail. The feature extraction methods are named as split and merge method, line tracking method, RANSAC method, Relative angles method, etc. In this reference [9], average number of features found and executions times of these methods are compared.

In this study, Split and Merge method is used for extraction. Split and merge algorithm is most commonly used algorithm for laser scanning. As all other feature extraction methods; Split and Merge Method is also cannot show a good performance with a complex environment.

6.1.1 Split and merge method

Split and Merge method consists of two parts as it name refers. The first part is split which aims to specify all edges and corners and the second part is merge which aims to eliminate false corners that can come from split part. The Split and Merge is a recursive algorithm and it operates as follows:

Split:

1. The first and last point of all points are initialized
2. Fit a line between these two extreme points
3. Fit another line between this line and the farthest point from this line.
4. If the last line distance is greater than threshold, the line between extreme points are split that means divide into two segments and repeat the above steps for each right and left segments until there will be no new segment.

Merge:

1. For each edge that is found in split part, fit the line between two extreme points.
2. Fit another line between this line and the farthest point from this line.
3. If the last line distance is less than or equal to threshold, the line between extreme points are merge that means eliminate the two edges and join the two extreme points.

The Split and Merge algorithm example is shown in Figure 6.1and Figure 6.2:

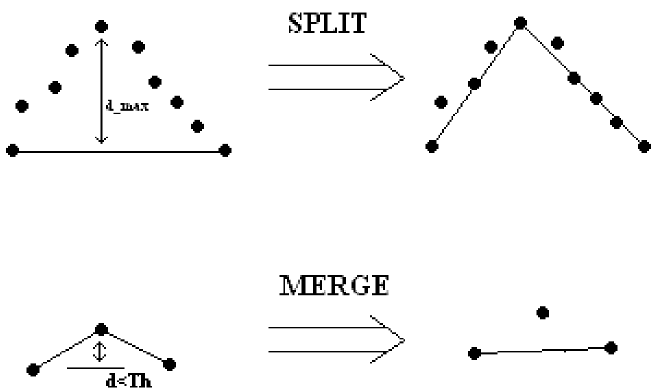


Figure 6.1 Split and Merge Algorithm example

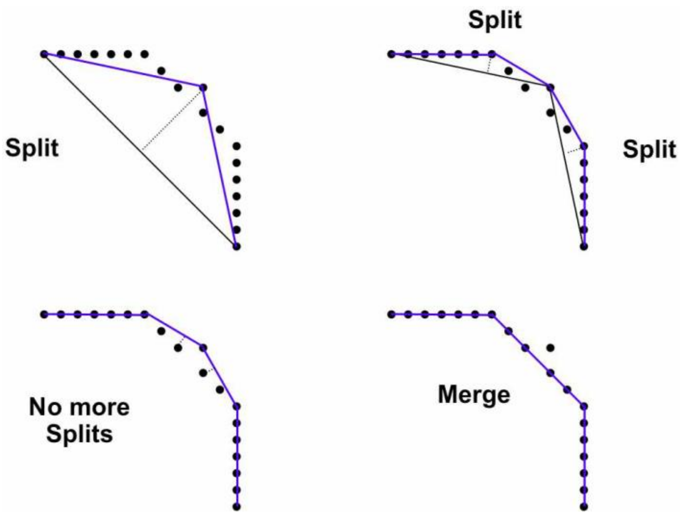


Figure 6.2 Split and Merge Algorithm example

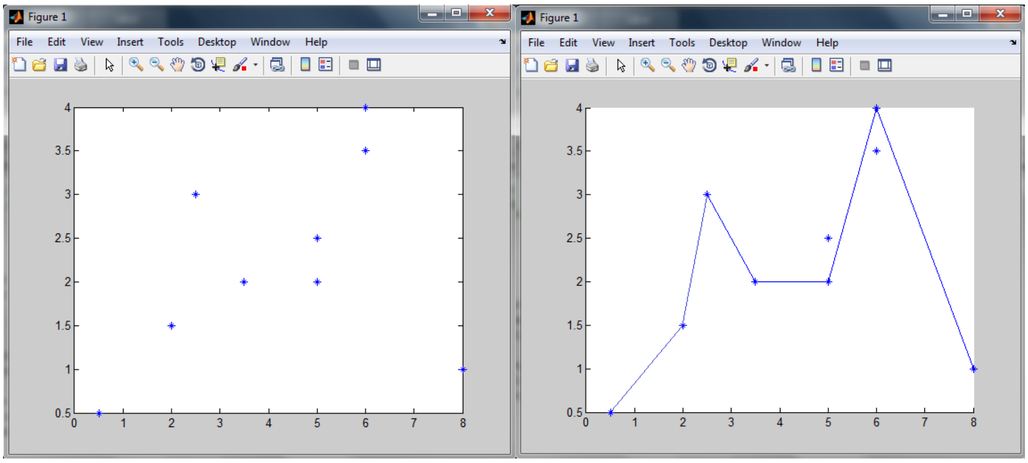


Figure 6.3 Split and Merge Algorithm Example with MATLAB

6.2 Data Association

The set of features extracted by each scan will be considered as a local map, so that it is a representation of the environment [3]. The features in the local map are associated with the previous scan feature map in order to perform data association algorithm.

6.2.1 Distance and angle (D&A) method

D&A algorithm is based on the following assumptions [1]:

1. Each inter-feature relationship implicitly defines a segment d connecting two point-type features; the relationship is defined as the length of the segment itself.

2. The inter-feature relationship defined by d must be invariant between two feature maps if features are both seen in the two maps.
3. The angle between two segments d must be, as for inter-feature relationship, invariant between two maps if they are visible in the two maps.

The aim of the D&A method is to find similarity between current and previous scans' feature maps.

To avoid wrong association, the D&A algorithm works as follow [1]:

1. At first two d_R points are chosen from the first map randomly as reference points. (purple segments in Figure 6.4)
2. Corresponding to reference points two associated points are searched from the second map. (green segments in Figure 6.4).
3. The distance between each feature and its previous end next feature for its map should be "close" to the one in the other map to let association to be considered as valid.
4. The angle between each segment d and the reference $d_R$for its map should be "close" to the one in the other map to let association to be considered as valid.

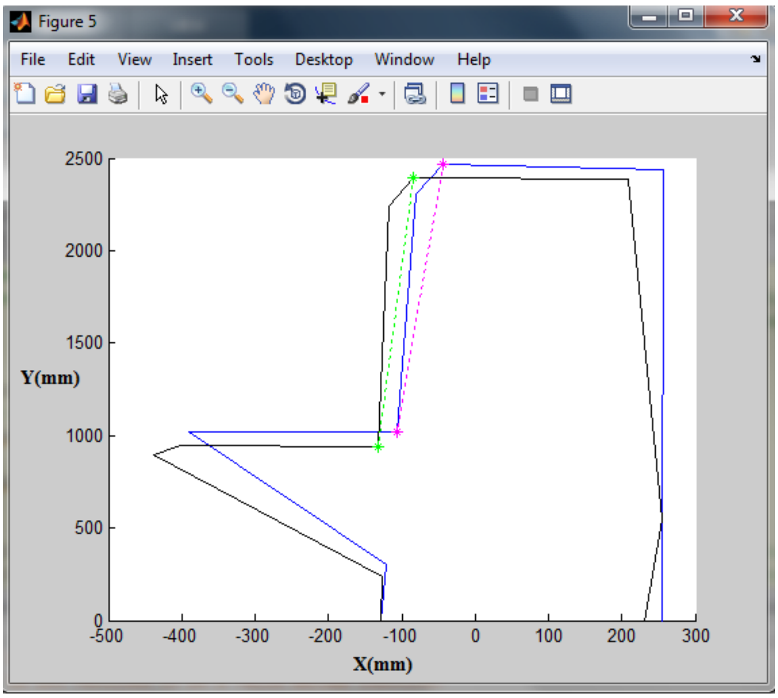


Figure 6.4 Example of data association between the first maps (blue) and the second one (black).

6.3 Data Relation

The concluding step is to obtain new coordinate points and rotation angle of mobile robot according to previous reference points. Based on the preceding steps, the new local map and previous one are matched and the associative features are extracted. In this part, there are two sets of points which are represented in two different cartesian reference frames. The two scans, as it is shown in Figure 6.5, differs only by a rotation θ and a translation T [3].

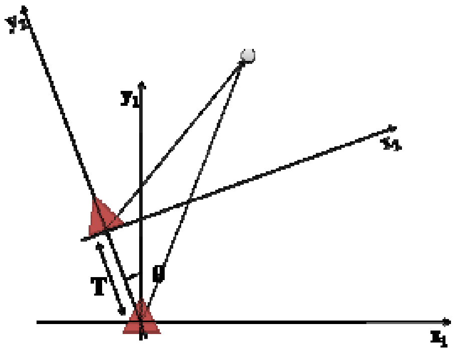


Figure 6.5 The frame of two consecutive scans

6.3.1 Estimate geometric transformation

MATLAB offers a function that calculates matched point pairs. After the data association step, four point sets is obtained. There are two point sets for the first map and also two point sets which correspond to first map's point sets in the second map.

Due to this MATLAB function, transform and rotation parameters are obtained according to associated data points. Estimate Geometric Transformation results is checked via roto-transform matrix.

6.3.2 Roto-transform matrix

Data from the current local map can be expressed in the frame of the previous one by using of a simple roto-translational matrix as expressed in below equation (6.1) [3].

$$P_1 = R(Q,T).P_2 \quad \textbf{(6.1)}$$

$P_1: Point\ 1\ according\ toframe\ 1$

$P_2: Point\ 1\ according\ toframe\ 2$

$R: Transformation\ matrix$

The open forms of each parameter in the equation (6.1) are shown in equation (6.2)

$$P_1 = \begin{bmatrix} x_1 & x_2 \\ y_1 & y_2 \end{bmatrix} \; R(Q,T) = \begin{bmatrix} cosQ & sinQ & T_x \\ sinQ & cosQ & T_y \\ 0 & 0 & 1 \end{bmatrix} \; P_2 = \begin{bmatrix} x'_1 & x'_2 \\ y'_1 & y'_2 \end{bmatrix} \qquad \textbf{(6.2)}$$

P1 and P2 matrices are augmented because of the matrix dimension for multiplication. After augmentation, equation (6.3) is obtained.

$$\begin{bmatrix} x_1 & x_2 \\ y_1 & y_2 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} cosQ & sinQ & T_x \\ sinQ & cosQ & T_y \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x'_1 & x'_2 \\ y'_1 & y'_2 \\ 1 & 1 \end{bmatrix} \qquad \textbf{(6.3)}$$

Transformation matrix elements' signs changes according to rotation direction. If the rotation occurs clockwise direction the transformation matrix is as in the equaqtion (6.4):

$$R(Q,T) = \begin{bmatrix} cosQ & sinQ & T_x \\ -sinQ & cosQ & T_y \\ 0 & 0 & 1 \end{bmatrix} \qquad \textbf{(6.4)}$$

When the equation (6.3) is solved with transformation matrix (6.4), equation s (6.5), (6.6), (6.7), and (6.8) are obtained.

$$x_1 = x'_1 * cosQ + y'_1 * sinQ + T_x \qquad \textbf{(6.5)}$$

$$y_1 = -x'_1 * sinQ + y'_1 * cosQ + T_y \qquad \textbf{(6.6)}$$

$$x_2 = x'_2 * cosQ + y'_2 * sinQ + T_x \qquad \textbf{(6.7)}$$

$$y_2 = -x'_2 * sinQ + y'_2 * cosQ + T_y \qquad \textbf{(6.8)}$$

If the rotation occurs counter clockwise direction, the transformation matrix is as in the equation (6.9):

$$R(Q,T) = \begin{bmatrix} cosQ & -sinQ & T_x \\ sinQ & cosQ & T_y \\ 0 & 0 & 1 \end{bmatrix} \quad \textbf{(6.9)}$$

When the equation (6.3) is solved with transformation matrix (6.4), equation s (6.5), (6.6), (6.7), and (6.8) are obtained.

$$x_1 = x_1' * cosQ - y_1' * sinQ + T_x \quad \textbf{(6.10)}$$

$$y_1 = x_1' * sinQ + y_1' * cosQ + T_y \quad \textbf{(6.11)}$$

$$x_2 = x_2' * cosQ - y_2' * sinQ + T_x \quad \textbf{(6.12)}$$

$$y_2 = x_2' * sinQ + y_2' * cosQ + T_y \quad \textbf{(6.13)}$$

For the above equations, x and y coordinate values are known from the data associtation step and MATLAB Estimate Geometric Transformation gives T_x ,T_y, and Q values. These T_x ,T_y, and Q values are substituted into the above equations with known x' and y' values and calculated x and y values. Then, checked calculated x and y values and cureent x and y values which are obtained from data association step. The values is matched. So, it means that the translational and rotation values are corrected.

In summary for this chapter is to describe localization steps which is follwed in this study. The steps are feature extraction, data association and date relation, respectively. Detailed information and examples about this steps is given in this chapter.

7. EXPERIMENTAL STUDIES

The LMS 200 laser sensor is located in ITU Control and Automation Laboratory corridor and scans the environment consecutively three times. Then, the scan data are saved and the procedure which is mentioned in previous chapters is performed.

7.1 Scan Data

The consecutive environment scans are shown in Figure 7.1, Figure 7.2, and Figure 7.3, respectively.

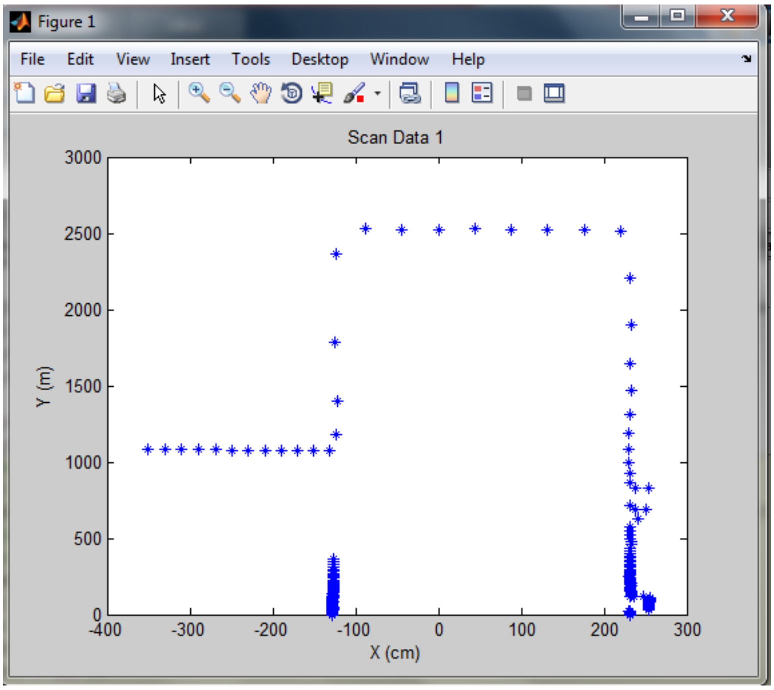


Figure 7.1 First Scan Data

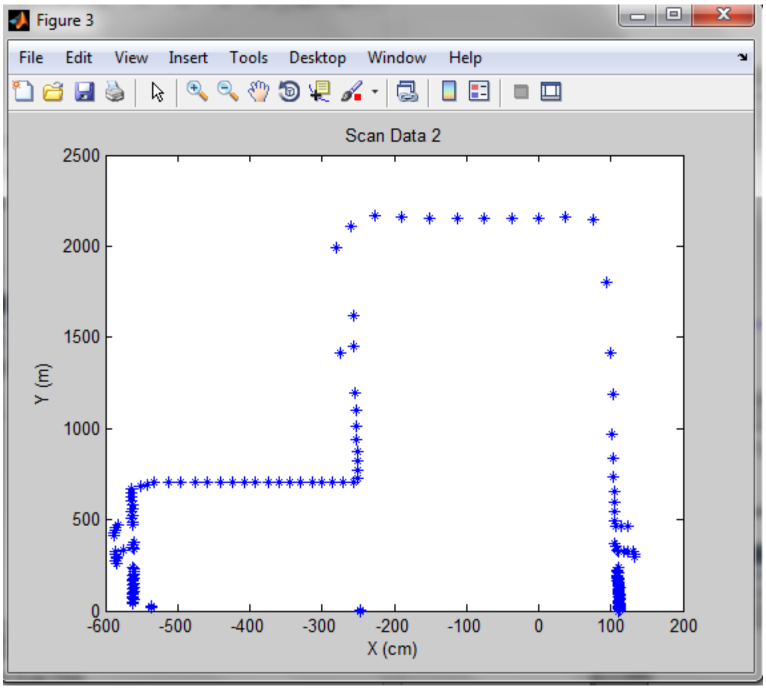


Figure 7.2 Second Scan Data

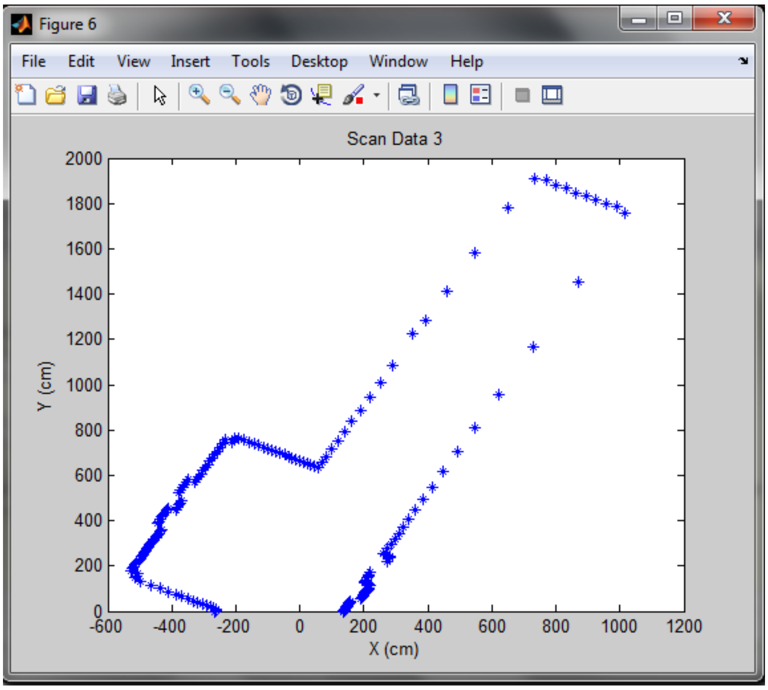


Figure 7.3 Third Scan Data

7.2 Split and Merge Method

After the environment scan, the results of feature extraction which are obtained by split and merge method for each scan data are shown in Figure 7.4, Figure 7.5, and Figure 7.6, respectively.

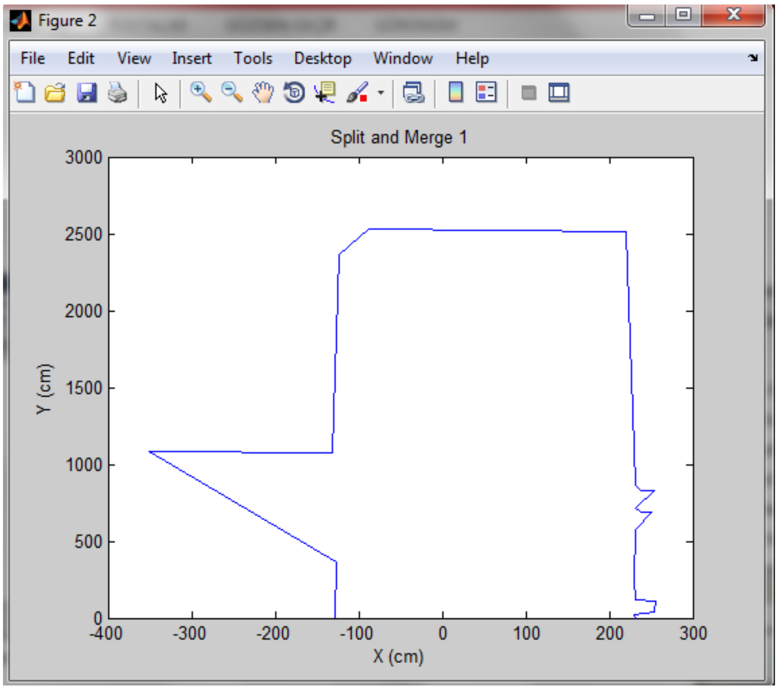


Figure 7.4 Result of Split and Merge Algortihm for First Scan Data

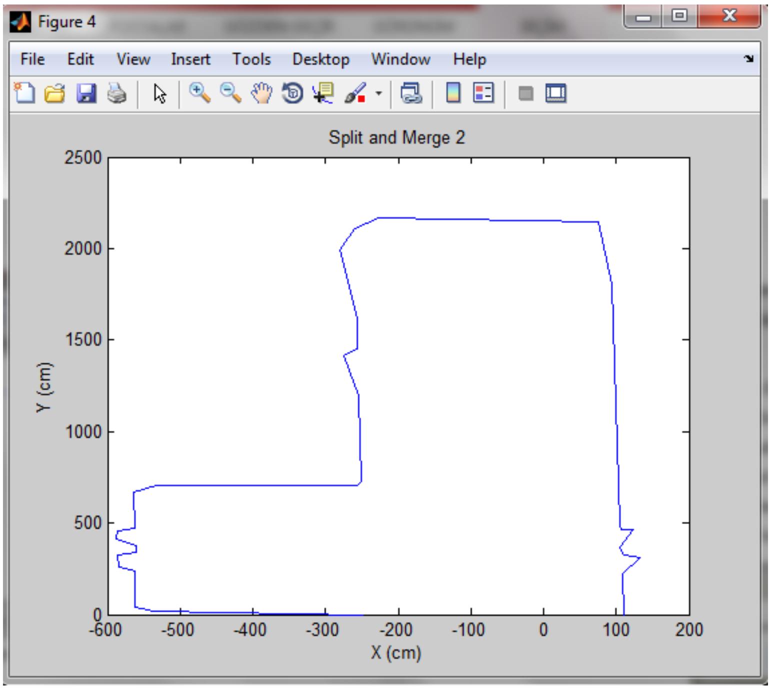


Figure 7.5 Result of Split and Merge Algortihm for Second Scan Data

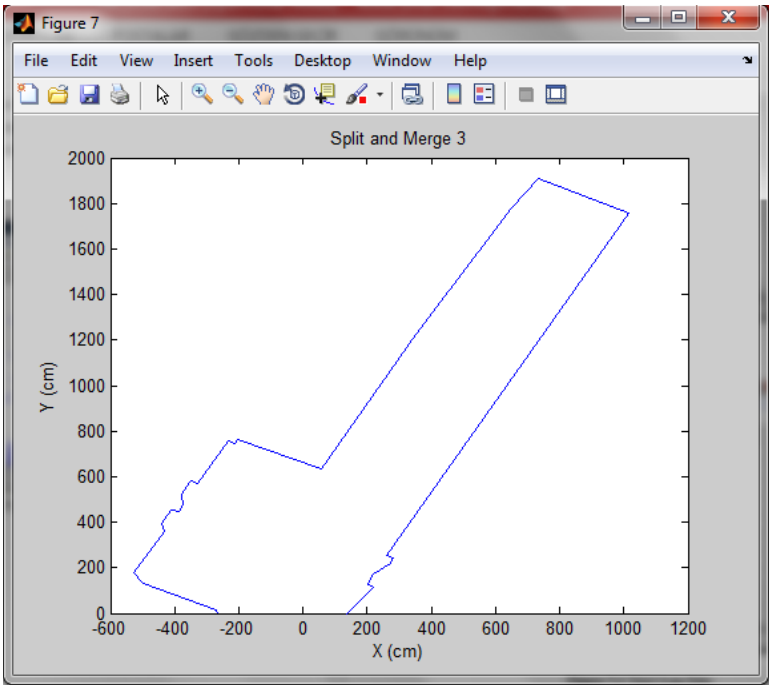


Figure 7.6 Result of Split and Merge Algortihm for Second Scan Data

7.3 D&A Method

As mentioned previous chapters, features between two local maps are associated with D&A method and obtain the features of new map which correspond to the features of previous map.

The initial reference features for the first map are selected randomly by MATLAB. So, this random selection is verified for logic. Thus, a threshold value is specified for distance and angle values. If this random selection is between this threshold values, the random selection is corrected and applied D&A method. If it is not, MATLAB will assign new random reference features until the features will be logical. These conditions are informed via message boxes as shown in Figure 7.7 and Figure 7.8:

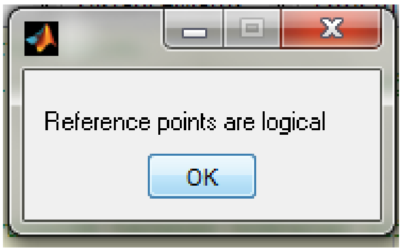


Figure 7.7 Message box about reference feature choice

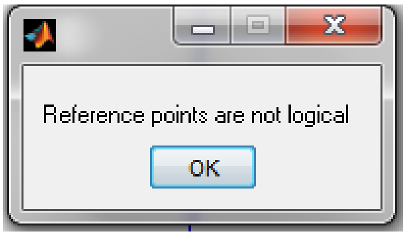


Figure 7.8 Message box about reference feature choice

When appropriate refrence points are chosen in the first scan data, in the second scan data the points which are associated with these reference points are find with D&A method. The associate points between two consecutive scans are shown in Figure 7.9 and Figure 7.10, respectively.

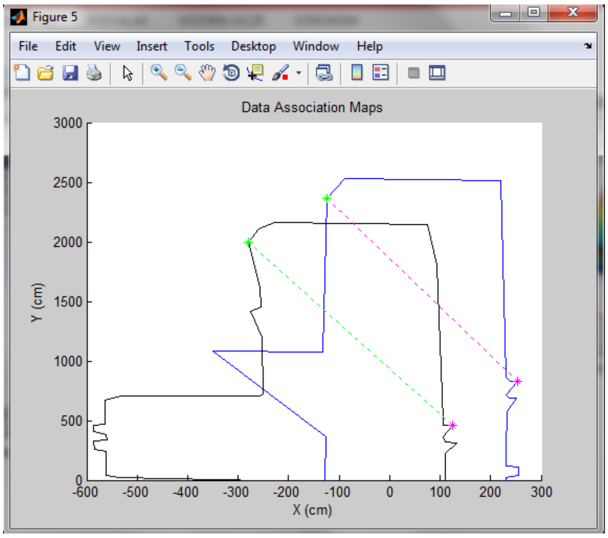


Figure 7.9 Associated features between first and second map

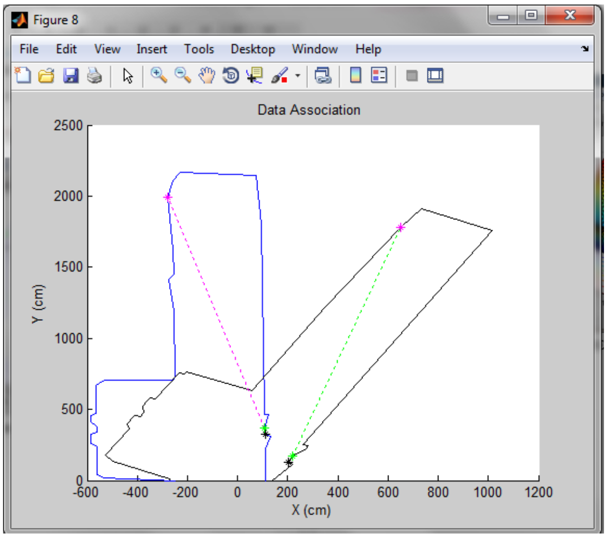


Figure 7.10 Associated features between second and first map

7.4 Data Relation

MATLAB Estimated Geometric Transformation gives the Tx, Ty and Q values according to associated points. Due to this transformation tha laser sensor position changes from the reference points (0,0) to the following positions, respectively.

Results of the Estimated Geometric Transformation is shown as a message box on MATLAB screen as in Figure 7.12 and Figure 7.15:

Estimated Geometric Transformation uses the associted points which are shown in Figure 7.11 and Figure 7.14, respectively.

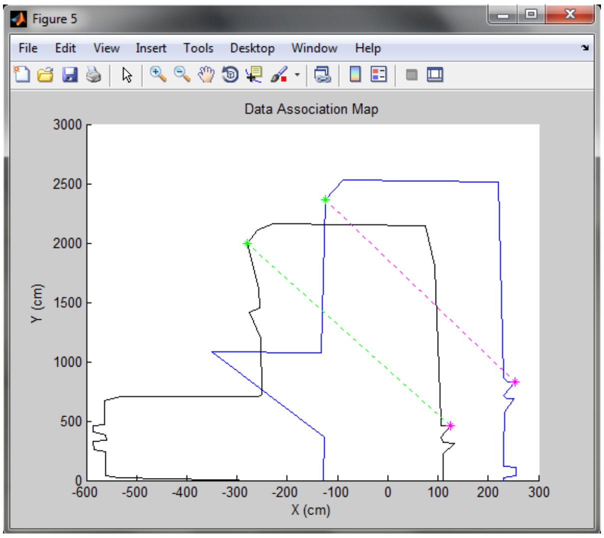


Figure 7.11 Estimated Geometric Transformation point pairs

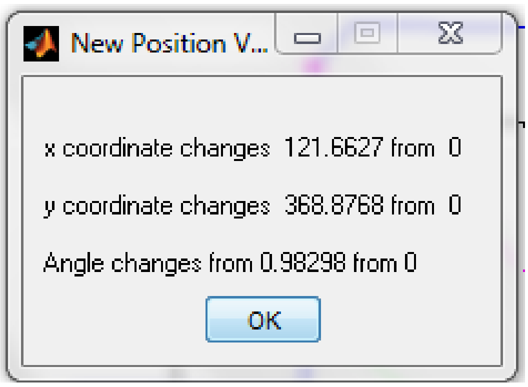


Figure 7.12 Estimated Geometric Transformation results

It is assumed that the changes between [-10 10] is 0 because of the small movement of the sensor due to manual movement. With the help of Estimated Geometric

Transformation results, the new coordinates are obtained and the new values for position are shown in Figure 7.13 and Figure 7.16, respectively.

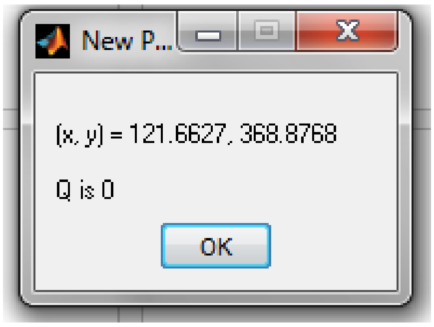


Figure 7.13 New coordinate values

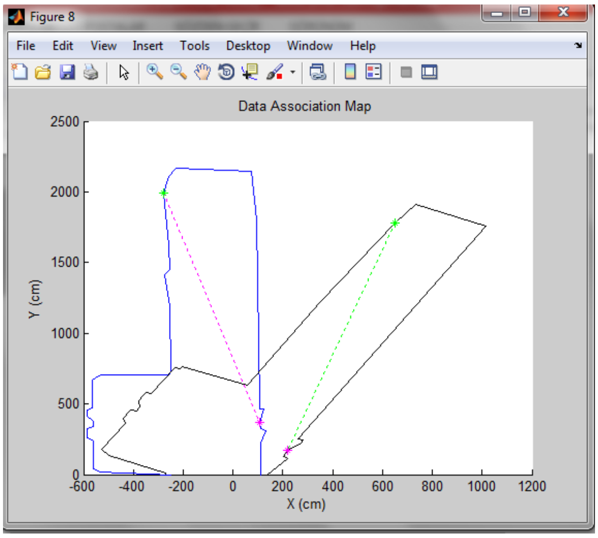


Figure 7.14 Estimated Geometric Transformation point pairs

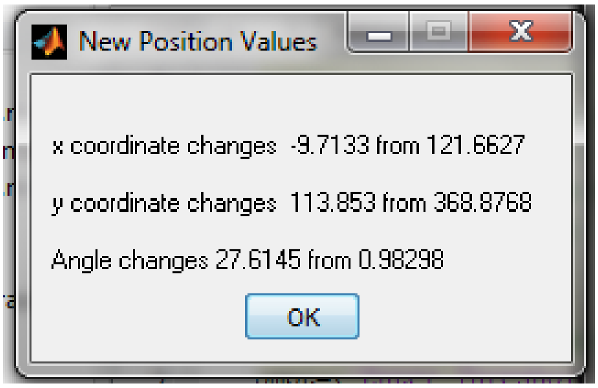


Figure 7.15 Estimated Geometric Transformation Results

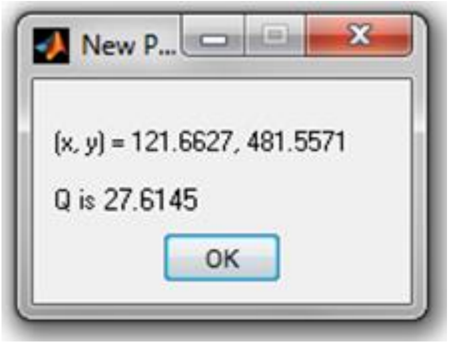


Figure 7.16 New coordinate values

Proof :

$$P_1 \quad = \quad R(Q,T) \quad * \quad P_2 \qquad \textbf{(7.1)}$$

$$\begin{bmatrix} x_1 & x_2 \\ y_1 & y_2 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} cosQ & sinQ & T_x \\ -sinQ & cosQ & T_y \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_1' & x_2' \\ y_1' & y_2' \\ 1 & 1 \end{bmatrix} \qquad \textbf{(7.2)}$$

For the first couple scan data Estimate Geometric Transformation results are as follows:

Tx= 121.6627 cm

Ty = 368.8768 cm

Q=0.9830

Associated points are as in the Figure 7.17:

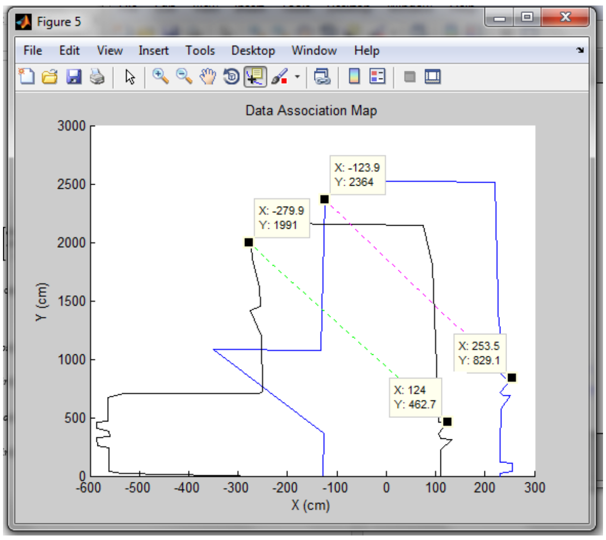


Figure 7.17 Associated points for the first couple scan

As it seen in the Figure 7.17 rotation direction is clockwise direction. So, the roto-transform equation is as follows:

$$R(Q,T) = \begin{bmatrix} cosQ & sinQ & T_x \\ -sinQ & cosQ & T_y \\ 0 & 0 & 1 \end{bmatrix} \quad \textbf{(7.3)}$$

$$\begin{bmatrix} x_1 & x_2 \\ y_1 & y_2 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} cosQ & sinQ & T_x \\ -sinQ & cosQ & T_y \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x'_1 & x'_2 \\ y'_1 & y'_2 \\ 1 & 1 \end{bmatrix} \quad \textbf{(7.4)}$$

$$\begin{bmatrix} -123.9 & 253.5 \\ 2364 & 829.1 \\ 1 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} \cos(0.9830) & \sin(0.9830) & 121.6627 \\ -\sin(0.9830) & \cos(0.9830) & 368.8768 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} -279.9 & 124 \\ 1991 & 462.7 \\ 1 & 1 \end{bmatrix}$$

$x_1 = x'_1 * cosQ + y'_1 * sinQ + T_x$

$x_1 = -279.9 * \cos(0.9830) + 1991 * \sin(0.9830) + 121.6627$

$x_1 = -278.9 + 34.15 + 121.6627$

$x_1 = -123.0873$

$y_1 = -x'_1 * sinQ + y'_1 * cosQ + T_y$

$y_1 = 279.9 * \sin(0.9830) + 1991 * \cos(0.9830) + 368.8768$

$y_1 = 4.802 + 1990.706 + 368.8768$

$y_1 = 2364.38$

$x_2 = x'_2 * cosQ - y'_2 * sinQ + T_x$

$x_2 = 124 * \cos(0.9830) + 462.7 * \sin(0.9830) + 121.6627$

$x_2 = 123.98 + 7.937 + 121.6627$

$x_2 = 253.58$

$y_2 = x'_2 * sinQ + y'_2 * cosQ + T_y$

$y_2 = 124 * \sin(0.9830) + 462.7 * \cos(0.9830) + 368.8768$

$y_2 = 2.127 + 462.63 + 368.8768$

$y_2 = 833.63$

For the second couple scan data Estimate Geometric Transformation results are as follows:

Tx= -6.7279 cm

Ty = 112.6803 cm

Q=28.2722

Associated points are as in the Figure 7.18:

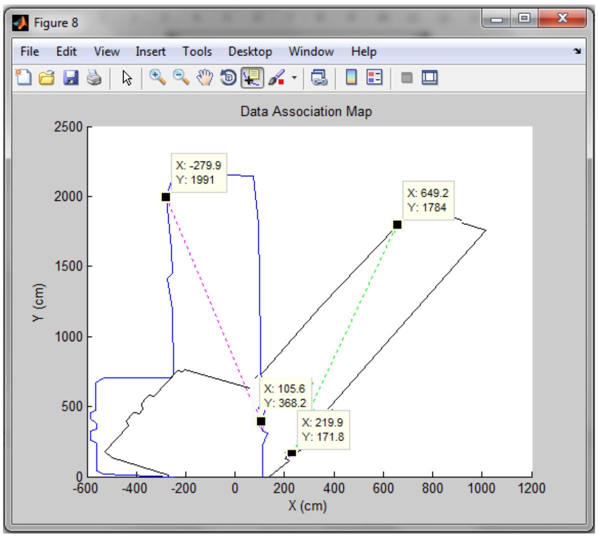


Figure 7.18 Associated points for the first couple scan

As it seen in the Figure 7.18 rotation direction is counter clockwise direction. So, the roto-transform equation is as follows:

$$R(Q,T) = \begin{bmatrix} cosQ & -sinQ & T_x \\ sinQ & cosQ & T_y \\ 0 & 0 & 1 \end{bmatrix} \quad \textbf{(7.5)}$$

$$\begin{bmatrix} x_1 & x_2 \\ y_1 & y_2 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} cosQ & -sinQ & T_x \\ sinQ & cosQ & T_y \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_1' & x_2' \\ y_1' & y_2' \\ 1 & 1 \end{bmatrix} \quad \textbf{(7.6)}$$

$$\begin{bmatrix} -279.9 & 105.6 \\ 1991 & 368.2 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} \cos(28.2722) & \sin(28.2722) & -6.7279 \\ -\sin(28.2722) & \cos(28.2722) & 112.6803 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 649.2 & 219.9 \\ 1784 & 171.8 \\ 1 & 1 \end{bmatrix}$$

$x_1 = x_1' * cosQ - y_1' * sinQ + \ T_x$

$x_1 = 649.2 * \cos(28.2722) - 1784 * \sin(28.2722) - 6.7279$

$x_1 = 571.76 - 845.01 - 6.7279$

$x_1 = \ -279.97$

$y_1 = x_1' * sinQ + y_1' * cosQ + \ T_y$

$y_1 = 649.2 * \sin(28.2722) + 1784 * \cos(28.2722) + \ 112.6803$

$y_1 = 307.5 + 1571.18 + 112.6803$

$y_1 = 1991.36$

$x_2 = \ x_2' * cosQ - y_2' * sinQ + \ T_x$

$x_2 = 219.9 * \cos(28.2722) - 171.8 * \sin(28.2722) - 6.7279$

$x_2 = 193.68 - 81.375 - 6.7279$

$x_2 = 105.57$

$y_2 = x_2' * sinQ + y_2' * cosQ + \ T_y$

$y_2 = 219.9 * \sin(28.2722) + 171.8 * \cos(28.2722) + \ 112.6803$

$y_2 = 104.16 + 151.3 + 112.6803$

$y_2 = 368.14$

In summary for this chapter is to give detailed experimantal studies about localization techniques given in the previous chapter. Scanning environment, feature extraction from scanned data, association of this data and relating between associated data is carried out as numerical.

8. CONCLUSION

In this study, indoor self-localization for 2D environmental is performed. After a gathering scan data, feature extraction method is applied for each scan feature maps with a split and merge algorithm. After obtaining less number of features instead of many points in one scan, the features are compared in two scan results and obtained changes with D&A algorithm. At the end of this procedure, in order to determine the rotation and transformation of the position, estimate geometric transfromation is applied between two consecutive feature maps.

Consequently, when the mobile robot is moved laser sensor scans the new environment and obtains new position successfully with applying scan matching methods.

REFERENCES

[1] **Filippo Bonaccorso, Giovanni Muscato, and Salvatore Baglio.** (2012). Laser Range Data Scan-Matching Algorithm for Mobile Robot Indoor Self-Localization. World Automation Conference, University of Catania - DIEEI

[2] **Feng Lu, and Evangelos E. Milios.** (1994). Robot Pose Estimation in Unknown Environments by Matching 2D. IEEE

[3] **F. Bonaccorso, F. Catania, and G. Muscato.** (2010) Evaluation of algorithms for indoor mobile robot self-localization through laser range finders data. In Proc. 7th IFAC Symposium on Intelligent Autonomous Vehicles.

[4] **G. A. Borges , and M.-J. Aldon**. (2000) A Split-and-Merge Segmentation Algorithm for Line Extraction in 2-D Range Images. IEEE

[5] **T. Bailey, E. M. Nebot, J. K. Rosenblatt and H. F. Durrant-Whyte.** (2000) Data Association for Mobile Robot Navigation: A Graph Theoretic Approach. IEEE

[6] **Technical Description** LMS200/211/221/291 Laser Measurement Systems

[7] **K. Lingemann, H. Surmann, A. Nuchter, and J. Hertzberg.**(2004) Indoor and Outdoor Localization for Fast Mobile Robots, In Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems.

[8] **T. Bailey, and E. Nebot**. (2001) Localisation in large-scale environments. Robotics and Autonomous Systems, vol. 37, pp. 261–281.

[9] **Chris Clark**. (2011) Autonomous Robot Navigation Lecture Notes

APPENDICES

APPENDIX A: Estimate Geometric Transformation

APPENDIX A: Estimate Geometric Transformation

Estimate geometric transform from matching point pairs.

tform=estimateGeometricTransform
(matchedPoints1,matchedPoints2,transformType)

returns a 2-D geometric transform object, tform. The tform object maps the inliers in matchedPoints1 to the inliers in matchedPoints2.

The function excludes outliers using the M-estimator Sample Consensus (MSAC) algorithm. The MSAC algorithm is a variant of the Random Sample Consensus (RANSAC) algorithm. Results may not be identical between runs because of the randomized nature of the MSAC algorithm.

Use the Estimate Geometric Transformation to find the transformation matrix which maps the greatest number of point pairs between two images. A point pair refers to a point in the input image and its related point on the image created using the transformation matrix. You can select to use the RANdom SAmple Consensus (RANSAC) or the Least Median Squares algorithm to exclude outliers and to calculate the transformation matrix. You can also use all input points to calculate the transformation matrix.

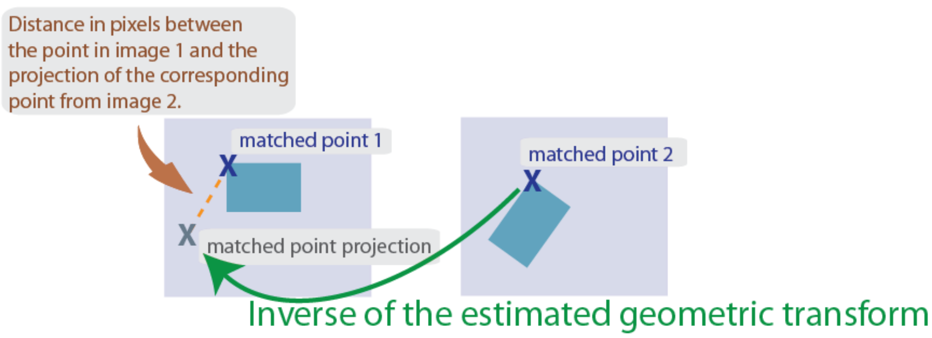


Figure A.1: Estimate geometric transformation

CURRICULUM VITAE

Name Surname: Hatice Erdoğan

Place and Date of Birth: Karabük– **06.07.1989**

E-Mail: haticeerdogn@gmail.com

EDUCATION:

B.Sc.: Yeditepe University – Systems Engineering