



## 非ホロノミック移動ロボットの制御のための拡張経路生成形レギュレータ

メタデータ	言語: eng 出版者: 公開日: 2015-06-11 キーワード (Ja): キーワード (En): 作成者: 代, 軍 メールアドレス: 所属:
URL	<a href="https://doi.org/10.15118/00005128">https://doi.org/10.15118/00005128</a>

**DOCTORAL DISSERTATION**

**The Extended Path-generating Regulator for Control of  
Nonholonomic Mobile Robots**

**Jun DAI**

Division of Production and Information Systems Engineering  
Muroran Institute of Technology

2015.3

## Abstract

Mobile robots, which have been developed for several decades, have been applied widely in various fields. One promising application is environmental surveys of vast areas, which require precise location and orientation control. In this research, the two-wheeled and four-wheeled mobile robot are objects of study. The path-generating regulator (PGR) was extended to track the arc passage and converge to the origin with steering angle saturation.

The PGR, which is a control method for robots so as to orient its heading toward the tangential direction of one of the curves belonging to the family of path functions, is applied to navigation problem for two-wheeled robot originally. Driving environments for robots are usually roads, streets, passages, and indoor. These tracks can be seen as consist of straight lines and arcs. In the case of small interval, arc can be regarded as straight line approximately, therefore we extended the PGR to drive the robot move along arc passage based on the theory that PGR along straight passage. In addition, the adjustable look-ahead method is proposed to improve the robot trajectory convergence property to the target arc. The effectiveness is proved through MATLAB simulations on both the comparisons with the PGR and the improved PGR with adjustable look-ahead method. The results of numerical simulations show that the adjustable look-ahead method has better convergence property and stronger capacity of resisting disturbance.

Another focus of this work analyzes the influence of steering angle saturation to the convergent property in the PGR under the feedback gain switching strategy for car-like robots. The PGR has been extended to car-like robots. Moreover, its convergent region has been expanded by the feedback gain switching strategy. However, under this strategy, when the robot restarts after the feedback gain switches, the command of the steering angle tends to be close to  $\pm\pi/2$  rad, which might exceed the maximum steering angle. This phenomenon causes steering angle saturation. The robot then drives along the minimum turning circle. In this paper, the convergent property of the robot under steering angle saturation is investigated. Results show that the convergent property is related strongly to the number of singular points, which depends on the center location of the minimum turning circle. The convergent properties at different locations are clarified through region division. An extended feedback gain switching strategy method is proposed to change the convergent property in the specific region. Based on simulation and experiment results, we summarize the convergent property related to the region and verify the proposed method.

# 論文要旨

車両型移動ロボットはこの数十年間研究開発が進み、様々な分野に応用されている。その有望な応用の一つに、広大なエリアの環境調査がある。環境調査の用途では正確な位置座標および方位角の制御が必要であるが、一般の車両は非ホロノミック拘束を持つためそのような制御が難しいという側面がある。本研究は二輪車両型と四輪車両型の移動ロボットを対象とする。論文の前段では、経路生成レギュレータ(以下 PGR と呼ぶ) を拡張し、二輪車輪型の移動ロボットを円/弧通路に追従させて走行させる手法について述べる。論文の後段では、四輪車両用 PGR を操舵角飽和がある場合にも適用できるように、原点への収束特性が分析することで得た新たな知見について論じる。

PGR は、運動制御の目的が達成できる経路関数群をあらかじめ用意しておき、移動ロボットはその経路関数一つに原点へ収束させるという手法である。PGR はその後、直線経路への追従問題に拡張された。ロボットの走行環境は道路、廊下および屋内であるが、その走行経路は多くの直線と円弧で構成される。そこで、PGR を円弧経路への追従問題に拡張することで、その適用範囲を広げることができる。ロボットの走行時、時間間隔がとても短い場合には、円弧を直線経路と見なすことができる。したがって、円弧上にその瞬間の局所座標を設定し、直線経路に沿う PGR 制御を逐次適用することで、円弧経路追従ができると考えられる。これを二輪車両型ロボットに適用しシミュレーションにより有効性を確認した。さらに、目標円弧への経路追従性を改善するために、可調整先読みという手法を提案した。改善した PGR の特性を MATLAB でシミュレーションした結果、可調整先読みを行うことで、目標円弧追従特性が向上し、外乱に対しても強い特性を持っていることを示した。

次に、四輪車両型 PGR の操舵角飽和问题について、原点への収束特性について解析した。PGR は前輪操舵後輪駆動の四輪車両型移動ロボット用に拡張され、さらに、制御則に内在する特異点問題を、フィードバックゲイン切替によって回避する方法が提案されている。しかし、この方法の下では、フィードバックゲインを切り替えて、ロボットが走行を再開する際に、操舵角の指令値は $\pm \pi/2[\text{rad}]$ になる。この時、指令値がロボットの最大操舵角を超過するため、操舵角が飽和し、ロボットは最小旋回円に沿って走行することになる。この論文では、操舵角飽和状態のロボットの収束特性を考察した。その結果、最小旋回円上の特異点の数と収束特性に強く関係があることを示す、特異点の数は最小の旋回円の中心位置に依存する。異なる位置での収束特性は領域を分けることによって明確にされる。最後に、特定領域の収束特性を改善するために、新たなフィードバックゲイン切替方法を提案した。さらにシミュレーションと実験結果に基づいて、収束特性の改善を確認した。

# Contents

Chapter 1 Introduction .....	1
1.1 Research Background.....	2
1.2 Nonholonomic constraints mobile robots .....	3
1.3 Previous Research .....	5
1.3.1 Previous Research on Nonholonomic Mobile Robots Control .....	5
1.3.2 Previous Research on Arc Tracking.....	6
1.3.3 Previous Research on the Influence of Steering Angle Saturation to Robot Control .....	7
1.4 Organization of Paper .....	7
Chapter 2 The Original PGR Control of Two-wheeled Mobile Robot .....	9
2.1 The PGR Converge to the Origin (PGR-O) for Two-wheeled Robots.....	10
2.1.1 Mathematic Model .....	10
2.1.2 Simulations.....	12
2.1.3 Experiments.....	12
2.2 The PGR along Straight Passage (PGR-S) for Two-wheeled Robots .....	14
2.2.1 Path Functions for Straight Passage .....	14
2.2.2 Steering Angle $u_2$ .....	15
2.2.3 Translational Velocity $u_1$ .....	15
Chapter 3 The Extended PGR to Track the Arc Passage (PGR-A) for Two-wheeled Robots .....	16
3.1 The extended PGR to Track Arc Passage (PGR-A).....	17
3.2 The Improved PGR-A with the Adjustable Look-ahead Method.....	19
3.3. Numerical Simulations .....	21
3.3.1 The Influence of Coefficient Constant $\lambda$ .....	21
3.3.2 The Influence of Time Intervals $d_t$ .....	25
3.3.3 The Influence with Disturbance Imposed on the Inputs of $u_1$ and $u_2$ .....	30
3.3.4. The PGR-A and Improved PGR-A with Adjustable Look-ahead Method to Track the S-shaped Passage.....	33
3.3.5 Discussion on Numerical Simulation Results .....	35

3.4 Conclusion.....	36
Chapter 4 The Extended PGR Converge to the Origin (PGR-O) for Car-like Robots.....	37
4.1 Car-like Robots State Equations.....	39
4.2 Steering Angle and Velocity Control .....	39
4.3 Path Function Group .....	40
Chapter 5 Singular Points and Feedback Gain Switching Strategy.....	43
5.1 Singular Points.....	44
5.2 Feedback Gain Switching Strategy.....	46
Chapter 6 The Influence of Convergent Property on PGR-O with Steering Angle Saturation .....	48
6.1 Running Area Division based on the Convergence.....	49
6.1.1 Minimum Turning Circle under Steering Angle Saturation.....	49
6.1.2 Robot Restart after Feedback Gain Switching Strategy .....	49
6.1.3 Region Division based on the Center Location of Minimum Turning Circle .....	51
6.2 Equipment.....	52
6.3 Simulation and Experiment Method .....	53
6.3.1 The Theory of Extended Kalman Filter SLAM.....	53
6.3.2 The Control of Car-like Robot with PGR-O and EKF-SLAM .....	58
6.4 Simulation and Experiment Conditions.....	58
6.5 Simulation Results.....	59
6.6 Experiment Results .....	61
6.7 Discussion.....	64
Chapter 7 Conclusion .....	67
Acknowledgments .....	69
References.....	71

## List of Figures

Fig. 1. 1: The environment survey robot work in landfill .....	2
Fig. 1. 2: A landfill.....	3
Fig. 1. 3: The environment survey robot .....	3
Fig. 2. 1: Two-wheeled robot mathematic model .....	10
Fig. 2. 2: Simulation results of two wheeled mobile robot .....	12
Fig. 2. 3: Two wheeled mobile Robot .....	13
Fig. 2. 4: Result of the experiment for two wheeled mobile robot.....	13
Fig. 2. 5: The coordinate system and the path functions for straight passage. The robot drives toward the positive direction of the $x$ -axis. The horizontal lines $y = W1$ and $y = -W2$ represent the boundary of passage, such as walls. ....	14
Fig. 3. 1: The path of target arc .....	17
Fig. 3. 2: Global and local coordinate systems .....	17
Fig. 3. 3: Flaw chart under PGR-A along arc passage .....	18
Fig. 3. 4: Global and local coordinate systems for the improved PGR-A with the adjustable look-ahead method .....	19
Fig. 3. 5: Flow chart under the improved PGR-A with the adjustable look-ahead method along arc passage .....	20
Fig. 3. 6: Trajectories in the $x$ - $y$ plain along the counterclockwise direction and time response of control commands $u1$ , $u2$ by the method in Section 3.1 at different values of $\lambda$ , from (a) to (c).....	22
Fig. 3. 7: Trajectories in the $x$ - $y$ plain along the counterclockwise direction and time response of control commands $u1$ , $u2$ by the PGR-A with the adjustable look-ahead method at different values of $\lambda$ , from (a) to (c).....	23
Fig. 3. 8: Trajectories in the $x$ - $y$ plain along the clockwise direction and time response of control commands $u1$ , $u2$ by the method in Section 3.1 with clockwise at different values of $\lambda$ , from (a) to (c).....	24
Fig. 3. 9: Trajectories in the $x$ - $y$ plain along the clockwise direction and time response of control commands $u1$ , $u2$ by the PGR-A with the adjustable look-ahead method at different values of $\lambda$ , from (a) to (c).....	25
Fig. 3. 10: Trajectories in the $x$ - $y$ plain along the counterclockwise direction and time response of control commands $u1$ , $u2$ by the method in Section 3.1 at different time interval values	

of $dt$ , from (a) to (c).....	26
Fig. 3. 11: Trajectories in the $x$ - $y$ plain along the counterclockwise direction and time response of control commands $u_1$ , $u_2$ by improved PGR-A with the adjustable look-ahead method at different time interval values of $dt$ , from (a) to (c).....	27
Fig. 3. 12: Trajectories in the $x$ - $y$ plain along the clockwise direction and time response of control commands $u_1$ , $u_2$ by the method in Section 3.1 at different time interval values of $dt$ , from (a) to (c).....	28
Fig. 3. 13: Trajectories in the $x$ - $y$ plain along the clockwise direction and time response of control commands $u_1$ , $u_2$ by improved PGR-A with the adjustable look-ahead method at different time interval values of $dt$ , from (a) to (c).....	29
Fig. 3. 14: Trajectories in the $x$ - $y$ plain along the counterclockwise direction and time response of control commands of PGR-A proposed in Section 3.1 at different disturbance values of $e_1$ , $e_2$ , from (a) to (b).....	30
Fig. 3. 15: Trajectories in the $x$ - $y$ plain along the counterclockwise direction and time responses of control commands by improved PGR-A with the adjustable look-ahead method for robot at different disturbance values of $e_1$ and $e_2$ , from (a) to (b).....	31
Fig. 3. 16: Trajectories in the $x$ - $y$ plain along the clockwise direction and time response of control commands of PGR-A proposed in Section 3.1 at different disturbance values of $e_1$ , $e_2$ , from (a) to (b).....	32
Fig. 3. 17: Trajectories in the $x$ - $y$ plain along the clockwise direction and time responses of control commands by improved PGR-A with the adjustable look-ahead method for robot at different disturbance values of $e_1$ and $e_2$ , from (a) to (b).....	33
Fig. 3. 18: The S-shaped model .....	34
Fig. 3. 19: Trajectories in the $x$ - $y$ plain and time responses of control commands by the PGR-A in (a) and the improved PGR-A with the adjustable look-ahead method in (b).....	35
Fig. 4. 1: The two-wheel model is equivalent to the four-wheel car-like robot .....	39
Fig. 4. 2: Target pose angle $\theta_r$ at lattice points when $n=2$ .....	40
Fig. 5. 1: Singular surfaces when $\lambda_1 = \lambda_2$ .....	44
Fig. 5. 2: Initial poses which do not pass singular points .....	44
Fig. 5. 3: The trajectories of initial poses in the outside of $D_1$ .....	45
Fig. 5. 4: The trajectories of initial poses in the outside of $D_2$ .....	45
Fig. 5. 5: Robot restarts after $\lambda_2$ is changed equivalent to the car-like robot to $\lambda_2'$ .....	46
Fig. 5. 6: The trajectories was adopted the feedback gain switching strategy when the initial poses outside of $D_1$ .....	47

Fig. 5. 7: The trajectories was adopted the feedback gain switching strategy when the initial poses outside of $D_2$ .....	47
Fig. 6. 1: Minimum turning circles in $x$ - $y$ plane .....	49
Fig. 6. 2: Trajectory which the robot drives along the minimum turning circles .....	50
Fig. 6. 3: Outline of $F_1$ and $F_2$ Fig. 6. 4: $F(\theta)$ ( $x_r < -R, y_r < -R$ ) .....	52
Fig. 6. 5: $F(\theta)$ ( $-R < x_r < 0, -R < y_r < 0$ )                      Fig. 6. 6: $F(\theta)$ ( $x_r < -R, -R < y_r < 0$ ).....	52
Fig. 6. 7: $F(\theta)$ ( $-R < x_r < 0, y_r < -R$ )                      Fig. 6. 8: Region division based on the center ...	52
Fig. 6. 9: Robot car .....	53
Fig. 6. 10: Experiment environment.....	53
Fig. 6. 11: Flow chart of control system applied for robot car .....	59
Fig. 6. 12: Simulation results for condition 1 ( $\lambda 2' = 0.1162$ ) .....	60
Fig. 6. 13: Simulation results for condition 2 ( $\lambda 2' = 0.2416$ ) .....	60
Fig. 6. 14: Simulation results for condition 3 ( $\lambda 2' = 0.5754$ ) .....	61
Fig. 6. 15: Simulation results for condition 4 ( $\lambda 2' = 0.0334$ ) .....	61
Fig. 6. 16: Experiment results for condition 1 .....	62
Fig. 6. 17: Experiment results for condition 2 .....	62
Fig. 6. 18: Experiment results for condition 3 .....	63
Fig. 6. 19: Experiment results for condition 4 .....	63
Fig. 6. 20: Running area can not converge to origin in region II .....	64
Fig. 6. 21: $F(\theta)$ when $\lambda 1 = 0.1$ , $\lambda 2' = 0.5346$ Fig. 6. 22: $F(\theta)$ when $\lambda 1' = 0.2$ , $\lambda 2' = 0.5346$ .....	64
Fig. 6. 23: Experiment results for condition 3 with the extended feedback gain switching strategy .....	65

## List of Tables

Table 3. 1: Default parameter values for all the numerical simulations. ....	21
Table 3. 2: Parameter values for numerical simulations with different coefficient constant. ....	22
Table 3. 3: Parameter values for numerical simulations in different time intervals. ....	26
Table 3. 4: Parameter values for numerical simulations imposed disturbance. ....	30
Table 3. 5: Parameter values for numerical simulations for S-shaped passage .....	34
Table 6. 1: Default parameter values for numerical simulation and experiment.....	59
Table 6. 2: The robot initial poses in numerical simulation and experiment .....	59

# **Chapter 1**

## **Introduction**

## 1.1 Research Background

Nonholonomic mobile robots have constraints imposed on the motion that are not integrable, i.e., the constraints cannot be written as time derivatives of some function of the generalized coordinates. The position control of nonholonomic mobile robots has been an important class of control problems.

In recently years, a variety of places are making environment surveys. The surveyed areas are generally very wide, if surveyed by manpower, it will waste a lot of human capital and time. Moreover, some areas have the gas that maybe presents a hazard to investigator because the gas can explode or burn. For solved these problems, the robot was considered to using for environment survey. The purpose of this research is for the environment survey in landfill with an autonomous car-like robot.

Shown in Fig. 1.1, Bacteria digest these organic wastes and produce methane gas and carbon dioxide as natural byproducts which were named LFG. This presents a hazard because the methane can explode or burn. So, the landfill gas must be removed. To do this, a series of pipes are embedded within the landfill to collect the gas. In some landfills, this gas is vented or burned.

Figure 1.2 is a landfill. Many points are LFG monitoring objects at the surrounding the landfill. If we are motioning these points by manpower, it will be spent a lot of human capital, time and LFG may be harmful. As the result, an on-site LFG monitoring system was proposed based on survey robot. Shown in Fig. 1.3, a monitoring pipe stands in the landfill, the robot equipped with a LFG monitoring sensor in the arm. When the survey robot closes to the object pipe, it outspreads the arm, puts the LFG monitoring sensor into the pipes, to collect the gas information. After the survey robot was finished in an object, go to the next object, until finished motioning the all objects. In order to verify the feasibility of solving the problem in PGR algorithm, the simulation and experiment was executed in robot car.

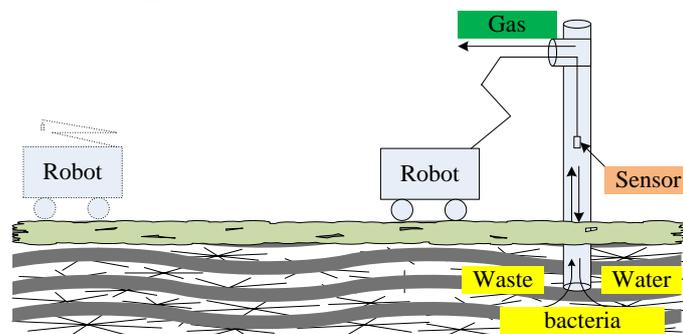


Fig. 1. 1: The environment survey robot work in landfill



Fig. 1. 2: A landfill



Fig. 1. 3: The environment survey robot

## 1.2 Nonholonomic constraints mobile robots

Nonholonomic systems are characterized by constraint equations involving the time derivatives of the system configuration variables. These equations are non integrable. They typically arise when the system has less controls than configuration variables. For instance a car-like robot has two controls (linear and angular velocities) while it moves in a 3-dimensional configuration space. As a consequence, any path in the configuration space does not necessarily correspond to a feasible path for the system. This is basically why the purely geometric techniques developed in motion planning for holonomic systems do not apply directly to nonholonomic ones.

We denote by  $A$  the robot and  $W$  its workspace. A configuration of  $A$  is specification of the position of every point in  $A$  with respect to a Cartesian frame embedded in  $W$ . The configuration space of  $A$  is the space, denoted by  $C$ , of all the possible configurations of  $A$ . the configuration space of a mechanical system made of rigid bodies is a smooth manifold. For instance, the configuration space of a two-dimensional rigid body translating and rotating in  $W = R^2$  is  $C = R^2 * S^1$ , where  $S^1$  denotes the unit circle. In virtually any practical situation, the range of positions reachable by the robot's bodies can be bounded, make  $C$  into a compact manifold.

In the following, we will represent a configuration  $q$  of  $A$  by a list of  $n$  parameters,  $(q_1, q_2, \dots, q_n)$ , where  $n$  is the dimension of  $C$ . This representation corresponds to defining an atlas of  $C$ . Each configuration  $q$  belongs to at least one neighborhood covered by a chart of the atlas. The parameters  $q_1, q_2, \dots, q_n$  are the coordinates of  $q$ . These parameters are also called generalized coordinates of  $A$ . For instance, we will represent the configuration of a car-like robot by  $q = (X_r, Y_r, \theta)$ , where  $X_r$  and  $Y_r$  are the coordinates of the midpoint between the two rear wheels of the car in Cartesian frame embedded in  $W$  and  $\theta$  is the orientation of the main axis of the robot relatively to the  $x$  axis of this Cartesian frame. Obviously, there is not a unique set of generalized coordinates for a given robot. By definition, the various charts put on a smooth manifold are C-related, which allows to extend differential properties established in a chart.

Now suppose that a scalar constraint of the form:

$$F(q, t) = 0 \tag{1-1}$$

With  $q \in C$  and  $t$  denoting time, applies to the motion of  $A$ . Let us further assume that  $F$  is smooth with non-zero derivative. Then, in theory one could use the equation to solve for one of generalized coordinates in terms of the other coordinates and time. Thus, equation (1-1) defines a  $(n-1)$ -dimensional submanifold of  $C$ . this submanifold in fact the actual configuration space of  $A$  and then  $n-1$  remaining coordinates its actual generalized coordinates. Constraint (1-1) is called a holonomic equality constraint. More generally, there may be  $k$  constraints of the form (1-1). If they are independent, i.e., their Jacobian matrix has full rank, they determine a  $(n - k)$  - dimensional submanifold of  $C$ , which is the actual configuration space of  $A$ .

A constant of form:

$$F(q, t) < 0 \text{ or } F(q, t) \leq 0$$

Where  $F$  is smooth with non-zero derivative, is a holonomic inequality constraint. It typically acts as a mechanical stop or an obstacle. It simply determines a submanifold of  $C$  having the same dimension as  $C$ .

Constraint (1-1) is only a kinematic constraint of some sort. Now, suppose that a scalar constraint of the form:

$$G(q, \dot{q}, t) = 0 \tag{1-2}$$

applies to the motion of  $A$ , with  $\dot{q} \in T_q(C)$ , the tangent space of  $C$  at  $q$ . the pair  $(q, \dot{q})$  belongs to  $TB(C)$ , the tangent bundle associated with the manifold  $C$ . the tangent space represents the space of the velocities of  $A$ . the tangent bundle is also called the phase space in physics and the state space in control theory. The tangent space of a smooth manifold is a vector space of the same dimension as the manifold. Hence,  $T_q(C)$  has dimension  $n$  for every  $q \in C$ . The tangent bundle  $TB(C)$  is a smooth manifold of dimension  $2n$ .

A kinematic constraint of the form (1-2) is holonomic if it is integrable, i.e.  $\dot{q}$  can be eliminated and equation (1-2) rewritten in the form (1-1). Otherwise, the constraint is called a nonholonomic equality constraint. As we will see below, a nonholonomic equality constraint restricts the space of velocities achievable by  $A$  at any configuration  $q$  to a  $(n-1)$ -dimensional linear subspace of  $T_q(C)$ , without affecting the dimension of the configuration space. If there are  $k$  independent nonholonomic equality constraints of the form (1-2), the space of achievable velocities is a subspace of  $T_q(C)$  of

dimension  $n-k$ .

A constraint of the form:

$$G(q, \dot{q}, t) < 0 \text{ or } G(q, \dot{q}, t) \leq 0$$

where  $G$  is not integrable, is a nonholonomic inequality constraint. It restricts the set of achievable velocities at any configuration  $q$  to a subset of  $T_q(C)$  having the same dimension as  $T_q(C)$ . A constraint bounding the steering angle of a car-like robot is a typical nonholonomic inequality constraint.

A nonholonomic constraint is generally caused by a rolling contact between two rigid bodies. It expresses that the relative velocity of the two points of contact is zero. When the motion in contact combines rolling and sliding, the expression, which depends on the friction coefficient of the two bodies, is nonlinear. When there is no sliding, the nonholonomic constraint is linear in  $\dot{q}$ . The second case, although less general than the first, is much simpler and quite widespread in practice.

In this research, the two kinds of nonholonomic mobile robots as the research projects, which are two-wheeled mobile robots and four-wheeled car-like robots.

### 1.3 Previous Research

#### 1.3.1 Previous Research on Nonholonomic Mobile Robots Control

Nonholonomic systems cannot be applied to methods of linear control theory, and they are not transformable into linear control problems. Due to both their richness and hardness, such nonlinear control problems have motivated a large number of researches involving various techniques of automatic control. Another difficulty in controlling nonholonomic mobile robots is that in the real world there are uncertainties in their modeling. Taking into account intrinsic characteristics of mobile robots such as actual vehicle dynamics, inertia and power limits of actuators and localization errors, their dynamic equations could not be described as a simplified mathematical model. A survey of recent developments in control of nonholonomic systems is described in [2]. To the authors' knowledge, the problem of dealing with model uncertainties is one of research problems for nonholonomic systems that require much attention but have yet to be extensively studied. Among previous researches, Jiang and Pomet [3, 4] applied back stepping technique to the adaptive control of nonholonomic systems with unknown parameters. A controller robust against localization errors of nonholonomic mobile robots was proposed by Hamel et al. [5], which considered the parking problem of mobile robots. In [6], a robust path-following controller for mobile robots was proposed guaranteeing exponential stability.

Nonholonomic mobile robots, which have been developed for several decades, have been applied widely in various fields. It is known that stabilization of nonholonomic wheeled mobile robots with restricted mobility to an equilibrium state is in general quite difficult. A well-known work of Brockett identifies nonholonomic systems as a class of systems that cannot be stabilized via smooth state feedback. It implies that problems of controlling

One promising application is environmental surveys of vast areas [7], which require precise location and orientation control. Car-like robots are nonholonomic systems. They are characterized by

constraint equations involving the time derivatives of the system configuration variables. It is difficult for the robot to converge to the target state by deriving a control law. Some earlier reports [24, 29, 31, 33] have proposed approaches to control the nonholonomic systems with closed loop control. Most such approaches design the feedback control system by converting variables such as input conversion and coordinate transformation of the mathematical model to a format called Chained form. It is noteworthy that the conversion variables can not be defined globally in such approaches. As a result, the feedback control law can not be defined globally either.

As described in this paper, the Path-generating Regulator (PGR) algorithm is applied to robot control [15]. Unlike other methods, the PGR is neither the coordinate transformation nor the input transformation. It is a control method that carries out asymptotic convergence of nonholonomic mobile robots to a given path function group. Its member functions pass through the origin. The gradient at the origin is equal to zero. This method has been extended to car-like robots. The convergent regions have been discussed [16]. The convergent regions are surrounded by singular points. If the robot starts from outside of the convergent regions, then it reaches the singular points on the way to the origin. At that instant, the speed command of the PGR becomes zero. To solve the limitation on singular points, the feedback gain switching strategy has been proposed [34]. Under this strategy, after the robot stops at the singular point, a new feedback gain is applied to shift the singular surfaces from the original position. However, when the robot restarts after the feedback gain switches, the command of the steering angle tends to be close to  $\pm\pi/2$  rad, which might exceed the maximum steering angle. This phenomenon causes steering angle saturation. In some cases, it might happen that the robot intersects with the new singular surface.

### **1.3.2 Previous Research on Arc Tracking**

Over the last few years, the development on robots has been paid close attention. There are some research projects such as cleaner robot of iRobot (IRBT) [8], office robot of Double Robotics [9], remote-presence robot of Mobile Access Consultation Services [10], and so on. These two-wheeled robots have common characteristic that move along the given route. So path tracking serves as an essential task for such autonomous robots.

For arc tracking problem, many approaches have been proposed, a dual estimation algorithm estimated the robot's position and wheel slips based on the Kalman filtering [11], but it is necessary to have previous knowledge about the system and measuring devices. A block iterative method known as Four Point-Explicit Group via Nine-Point Laplacian (4EG9L) was used for solving robot path planning problem [12]. Most such approaches design the feedback control system by converting variables such as input conversion of the mathematical model to a format called Chained form. It is noteworthy that the conversion variables can not be defined globally in such approaches.

Compared with the other papers, the originality of this paper is proposed the PGR and improved PGR with adjustable look-ahead method to track the arc for two-wheeled robot. It is a control method that carries out asymptotic convergence of nonholonomic mobile robots to a given path function group.

Two-wheeled robots belong to nonholonomic constraints system [13], which make it difficult for robots to converge to the target state by deriving a control law [14]. In one of our previous works, we

proposed the path-generating regulator (PGR) method, which controls the robot to move forward the tangential direction of the curve which passes through the robot current position among the family of path functions [15-17]. This method allows us to make the robot stop at the origin of the rectangular coordinate system. Simultaneously, the global asymptotic stability of PGR has been proven. Because driving environments for robots are usually roads, streets, passages, and ridges. These tracks can be seen as the consist of straight lines and arcs. Recently, the PGR has been extended to path tracking problem along straight passage for two-wheeled robots [18] and the validity has been verified by simulations and experiments. Therefore, we further investigate the PGR and the improved PGR with adjustable look-ahead algorithm to track the arc passage in this paper.

### **1.3.3 Previous Research on the Influence of Steering Angle Saturation to Robot Control**

For the influence of steering angle saturation to robot control, some approaches have been proposed. For instance, on linear sliding mode control for an unmanned agricultural tractor has been analyzed in the presence of sliding and control saturation [23]. Semiglobal stabilization for nonholonomic mobile robots has been discussed based on dynamic feedback with inputs saturation [27]. The influence to the multi-axles driving for wheeled mobile robots with geometry and kinematical constraint control has been addressed [29]. However, these approaches are not incorporated directly into the PGR.

## **1.4 Organization of Paper**

This study investigates the extended PGR to track arc passage and the improved PGR with the adjustable look-ahead method first, then we investigates the influence of steering angle saturation to the convergent property in the PGR under the feedback gain switching strategy for car-like robots, and clarifies the convergent properties at different locations through region division. Moreover, we propose a method of extension to feedback gain switching to change the convergent property in the specific region.

The remainder of this paper is organized as follows. Chapter 2 reviews the PGR converge to the origin and along straight passage for two-wheeled robots. In Chapter 3, we propose Extended PGR to track arc passage and the improved PGR with the adjustable look-ahead method. This is the first main issue of this paper. As explained in Chapter 4, the PGR is extended for car-like robot converge to origin. Chapter 5 introduce the singular points in the extend PGR and proposed the solving method of feedback gain switching strategy. The second main issue of this paper is Chapter 6 that analysis the influence of steering angle saturation based on the minimum turning circle. The simulation and experiment are conducted and discussed based on the analysis. Concluding remarks are presented in Chapter 7.



## **Chapter 2**

### **The Original PGR Control of Two-wheeled Mobile Robot**

## 2.1 The PGR Converge to the Origin (PGR-O) for Two-wheeled Robots

### 2.1.1 Mathematic Model

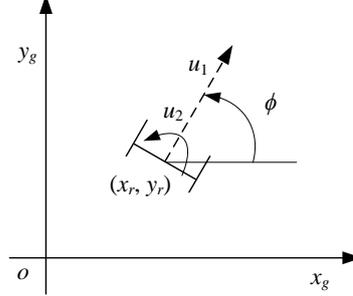


Fig. 2. 1: Two-wheeled robot mathematic model

In this research, the mathematic model shown in Fig. 2.1. The kinematic variables of the two-wheeled robot are as follows:  $x_r$  and  $y_r$  are the coordinates of the robot in the world reference frame. The translational velocity of the center of the robot  $u_1$ , is related to the velocity in the  $x$  and  $y$  directions,  $\dot{x}_r$  and  $\dot{y}_r$ , through (2-1) and (2-2), where  $\theta$  is the orientation angle of the robot with respect to the reference frame. The steering angle of the robot  $u_2$ , is the change rate  $\dot{\theta}$  of the orientation angle, through (2-3).

$$\dot{x}_r = u_1 \cos \phi \quad (2-1)$$

$$\dot{y}_r = u_1 \sin \phi \quad (2-2)$$

$$\dot{\theta} = u_2 \quad (2-3)$$

Path tracking serves as an essential task for autonomous mobile robots. To solve the nonholonomic problem of mobile robot, A control method called Path-generating Regulator or PGR in short has been proposed which designs a nonlinear regulator carrying out asymptotic convergence to a given trajectory family. It is a method aimed at controlling mobile robot to move in the tangential direction of the path which passes through the current position of the robot among the path group [16]. The purpose is to make the robot stop at the origin of the rectangular coordinate system fixed to the ground. And the global asymptotic stability of this method for two-wheeled mobile robots has been proved [17].

The control target is to make the robot start at random initial state  $(x, y, \theta)$  and converge at the origin  $(0, 0, 0)$ .

Here we set the target angle as  $\theta_r$  and we will have the following function.

$$\{\theta_r = \theta_r(x, y) : \theta_r(0,0) = 0 \quad (2-4)$$

We use the  $\theta_r$  to control the angle of robot when it moves to the origin. The deviation between target angle  $\theta_r$  and the actual angle  $\theta$  is set as  $e$ .

$$e = \theta - \theta_r \quad (2-5)$$

We use  $e$  to set up the following first order system in which  $\lambda > 0$ .

$$\dot{e} = -\lambda e \quad (2-6)$$

The following expression can be obtained based on (2-3) and (2-6).

$$\dot{e} = \dot{\theta} - \dot{\theta}_r = u_2 - \frac{\partial \theta_r}{\partial x} \dot{x} - \frac{\partial \theta_r}{\partial y} \dot{y} \quad (2-7)$$

Therefore, we set  $u_2$  as follows to satisfy the expression (2-6).

$$u_2 = -\lambda e + \frac{\partial \theta_r}{\partial x} \dot{x} + \frac{\partial \theta_r}{\partial y} \dot{y} \quad (2-8)$$

According to (2-1), (2-2) and (2-8), we will get the following expression of  $u_2$ .

$$u_2 = -\lambda e + \left\{ \frac{\partial \theta_r}{\partial x} \cos(e + \theta_r) + \frac{\partial \theta_r}{\partial y} \sin(e + \theta_r) \right\} u_1 \quad (2-9)$$

We set up the error system by rewriting the mathematical model of two-wheeled mobile robot with deviation.

$$\begin{aligned} \dot{x} &= u_1 \cos(e + \theta_r) \\ \dot{y} &= u_1 \sin(e + \theta_r) \\ \dot{e} &= -\lambda e \end{aligned} \quad (2-10)$$

To study the stability of this control method, a candidate of Lyapunov function can be set as follows.

$$V = \frac{1}{2}(e^2 + \lambda_1 x^2 + \lambda_2 y^2) \quad (2-11)$$

Here,  $\lambda_1, \lambda_2 > 0$ . The time derivative can be obtained as follows.

$$\frac{d}{dt} V = e\dot{e} + \lambda_1 x\dot{x} + \lambda_2 y\dot{y} = -\lambda e^2 + \{ \lambda_1 x \cos(e + \theta_r) + \lambda_2 y \sin(e + \theta_r) \} u_1 \quad (2-12)$$

Here, if we set  $u_1$  as follows,

$$u_1 = -\{ \lambda_1 x \cos(e + \theta_r) + \lambda_2 y \sin(e + \theta_r) \} \quad (2-13)$$

We will get the following expression.

$$\frac{d}{dt} V = -\lambda e^2 + \{ \lambda_1 x \cos(e + \theta_r) + \lambda_2 y \sin(e + \theta_r) \}^2 \quad (2-14)$$

From the expression (2-14), we can know that it will not be positive definitely beside origin. Thus, (2-11) can be used as Lyapunov function. We choose the following function as the path function.

$$y = ax^n \quad (2-15)$$

Here,  $n > 1$ ,  $a$  is random constant. Then we can get its partial differential.

$$\frac{dy}{dx} = anx^{n-1} \quad (2-16)$$

After some calculations, we can have the target angle  $\theta_r$ .

$$\tan \theta_r = \frac{ny}{x} \quad (2-17)$$

Finally, the expressions of control law can be listed as follows.

$$u_1 = -(\lambda_1 x \cos \theta + \lambda_2 y \sin \theta) \quad (2-18)$$

$$u_2 = -\lambda \left\{ \theta - \tan^{-1} \left( \frac{ny}{x} \right) \right\} + \frac{n(x \sin \theta - y \cos \theta)}{x^2 + n^2 y^2} u_1 \quad (2-19)$$

### 2.1.2 Simulations

The parameters of control law are set to  $n=2$ ,  $\lambda=4$ ,  $\lambda_1=2$ ,  $\lambda_2=1$ . We will change the initial states of the robot from 1 to 5 and in state 6 we will only change the value of  $n$ .

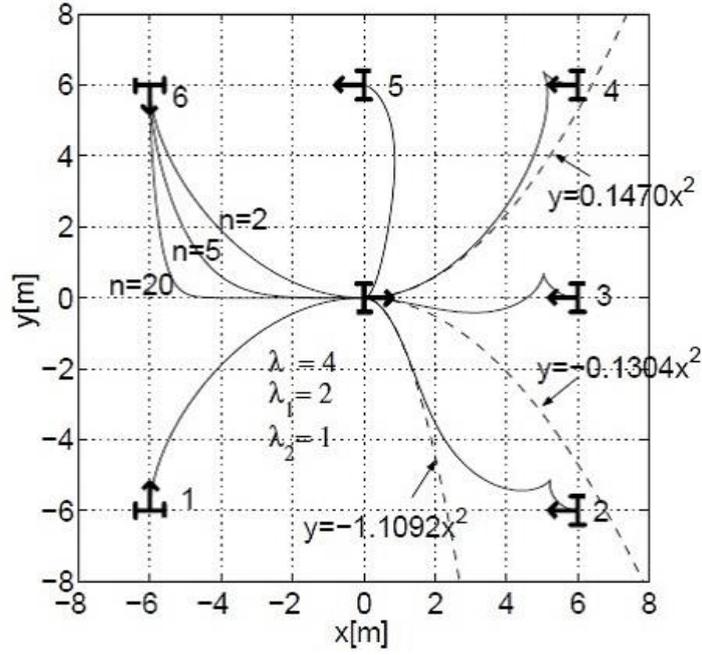


Fig. 2. 2: Simulation results of two wheeled mobile robot

We can see from the results of simulations in Fig. 2.2 that in all the simulations the robot will finally converge at the origin. When the values of  $n$  change, the slope of trajectories will also change. What's more, when all the parameters are set to the same, the generated trajectories can be different when the initial states are different. For example, as explained before, we use  $y = ax^2$  as path function. In state 2,  $a$  is set to be -1.1092. In state 3, it is set to be -0.1304 and 0.1470 in case 4. In these examples, the robot will move forward first then move backward to the origin.

### 2.1.3 Experiments

The proposed conventional path-generating regulator is used to carry on experiment on the two wheeled mobile robot in the Fig. 2.3. The distance between two wheels is 71 mm and the radius of the wheel is 27 mm. An infrared LED is equipped on the top of the robot and a PSD is used to get the absolute position of the robot.

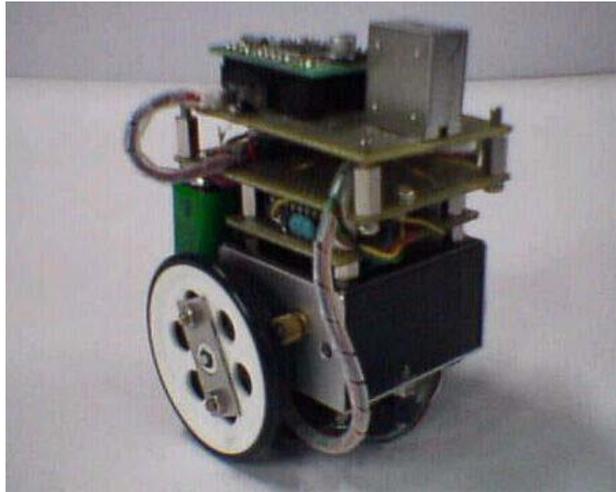


Fig. 2. 3: Two wheeled mobile Robot

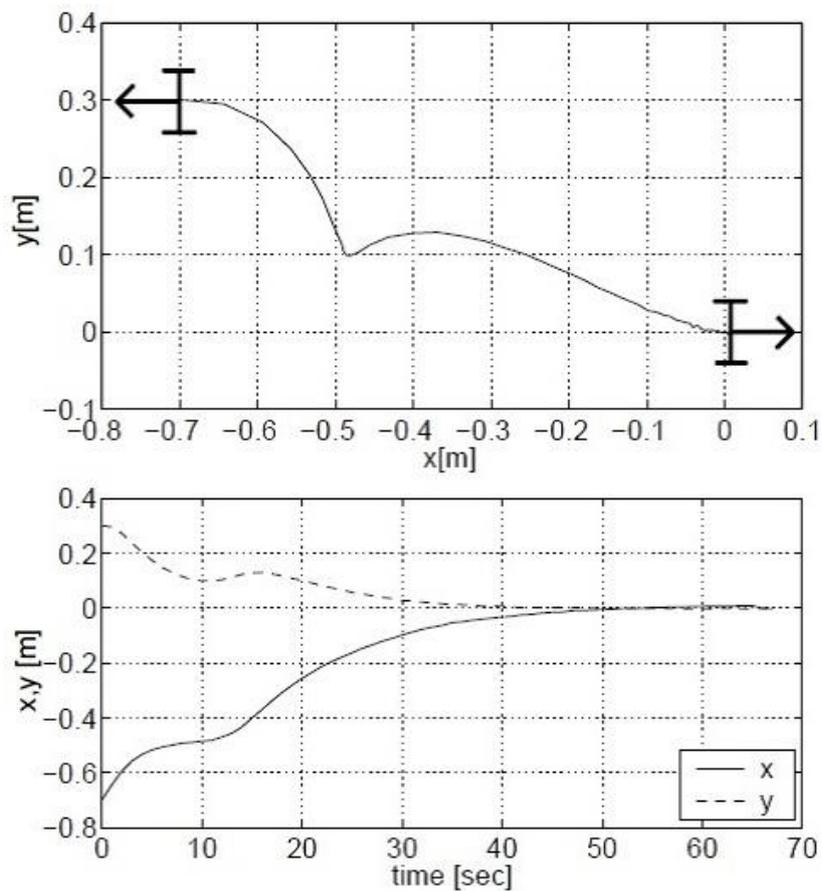


Fig. 2. 4: Result of the experiment for two wheeled mobile robot

One example of experiments is shown in Fig. 2.4. The initial state is set to be  $(x_0, y_0, \theta_0) = (-0.7[\text{m}], 0.3[\text{m}], \pi[\text{rad}])$ . And the parameters are set to be  $n=2, \lambda=0.1, \lambda_1=0.1, \lambda_2=0.1$ . We can see from the figure that the robot moved backward first until  $x$  was around  $-0.5$  m, then it turned to move forward and converged at the origin.

## 2.2 The PGR along Straight Passage (PGR-S) for Two-wheeled Robots

The PGR was used to solve the path tracking problem along straight passage for two-wheeled robots. To illustrate this method clearly, we establish the robot's mathematic model, definite the path functions and carry out necessary mathematical calculation, and then deduce the steering angle and the moving speed formulas.

### 2.2.1 Path Functions for Straight Passage

The family of path functions for a half-side of straight passage is defined as the following formula from reference [11].

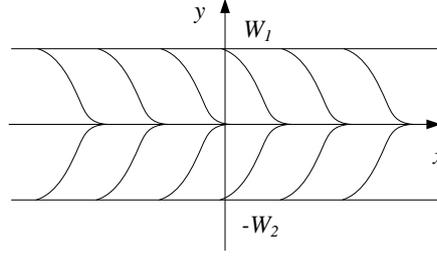


Fig. 2. 5: The coordinate system and the path functions for straight passage. The robot drives toward the positive direction of the  $x$ -axis. The horizontal lines  $y = W_1$  and  $y = -W_2$  represent the boundary of passage, such as walls.

$$y = \begin{cases} W & x < \frac{-\pi+b}{a} \\ \frac{W}{2}(1 - \cos(ax - b)) & \frac{\pi+b}{a} < x < \frac{b}{a} \\ 0 & x > \frac{b}{a} \end{cases} \quad (2-20)$$

where  $a$  is a positive constant which adjusts the slope of the curve,  $b$  is a translation value of path functions family, and  $W$  is the half-width of the passage. Instead of  $W$ , we take  $W_1$  as the left half width, i.e., in the domain of  $y > 0$ ,  $W_2$  as the right half width, i.e., in the domain of  $y < 0$ . When  $y$  is negative, the sign of the right side of (2-20) needs to be changed. The graph of the functions is shown in Fig. 2.5. The robot drives through the passage toward the positive direction of the  $x$ -axis.  $\phi_r$  can be expressed as (5).

$$\phi_r = \begin{cases} \tan^{-1}(-a\sqrt{(W_1 - y)y}) & y \geq 0 \\ \tan^{-1}(a\sqrt{-(W_2 + y)y}) & y < 0 \end{cases} \quad (2-21)$$

Note that  $\phi_r$  is calculated only by the  $y$  coordinate in the region  $-W_2 \leq y \leq W_1$ . The partial derivative of  $\phi_r$  with respect to  $y$  is calculated as follows.

$$\frac{\partial \phi_r}{\partial y} = \begin{cases} -\frac{a(W_1 - 2y)\sqrt{(W_1 - y)y}}{2y(1 + a^2(W_1 - y)y)(W_1 - y)} & y \geq 0 \\ \frac{a(W_2 + 2y)\sqrt{-(W_2 + y)y}}{2y(1 - a^2(W_2 + y)y)(W_2 + y)} & y < 0 \end{cases} \quad (2-22)$$

### 2.2.2 Steering Angle $u_2$

The deviation between the target angle  $\phi_r$  and the actual angle  $\phi$  is set as  $\delta$ .

$$\delta = \phi - \phi_r \quad (2-23)$$

Under the control of  $u_2$ ,  $\delta$  obeys the following derivative equation of the first order delay system.

$$\dot{\delta} = -\lambda\delta \quad (2-24)$$

where  $\lambda$  is a coefficient constant, when  $\delta$  converges to 0,  $\phi$  approaches  $\phi_r$  simultaneously. According to (3), (6) and (8), we obtain  $u_2$  for straight passage as follows.

$$u_2 = \begin{cases} g_1(y, \phi) & 0 < y < W_1 - \varepsilon \\ g_2(y, \phi) & -W_2 + \varepsilon < y < 0 \end{cases} \quad (2-25)$$

$$g_1(y, \phi) = -\lambda \left( \phi + \tan^{-1}(a\sqrt{(W_1 - y)y}) \right) - \frac{a(W_1 - 2y)\sqrt{(W_1 - y)y}}{2y(1 + a^2(W_1 - y)y)(W_1 - y)} u_1 \sin\phi \quad (2-26)$$

$$g_2(y, \phi) = -\lambda \left( \phi - \tan^{-1}(a\sqrt{-(W_2 + y)y}) \right) + \frac{a(W_2 + 2y)\sqrt{-(W_2 + y)y}}{2y(1 - a^2(W_2 + y)y)(W_2 + y)} u_1 \sin\phi \quad (2-27)$$

where  $\varepsilon$  is a small positive constant. To avoid division by zero in computer calculation of (10) and (11), when the value of  $y$  is around 0,  $W_1$  or  $W_2$  and the speed command  $u_1$  need to be nonzero value.

### 2.2.3 Translational Velocity $u_1$

The  $u_1$  is derived by Lyapunov's stability method. A hybrid continuous control algorithm in (12), of which the second part puts more emphasis on advance close to the  $x$ -axis and the first part guarantees stability in other location, is applied. The control algorithm can be expressed as follows.

$$u_1 = -\left(1 - K_m e^{-c_m y^2}\right) \frac{1 - e^{-c_m y \sin\phi}}{1 + e^{-c_m y \sin\phi}} V_m + K_m e^{-c_m y^2} V_m \quad (2-28)$$

where,  $K_m e^{-c_m y^2}$  is the modification coefficient used to adjust the emphasis between two parts.  $K_m$  is within the limit of  $0 \leq K_m \leq 1$  and  $e^{-c_m y^2}$  will be equal to 1 on the  $x$ -axis and close to 0 away from the  $x$ -axis.  $c_m$  is an adjustable parameter and  $c_m > 0$ .

## **Chapter 3**

# **The Extended PGR to Track the Arc Passage (PGR-A) for Two-wheeled Robots**

### 3.1 The extended PGR to Track Arc Passage (PGR-A)

In a small interval, arc can be seen as line approximately. The PGR along arc passage based on the theory of straight passage is proposed in this Section.

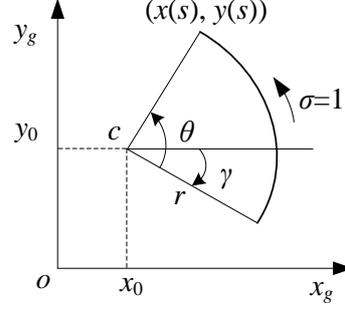


Fig. 3. 1: The path of target arc

As shown in Fig. 3.1, we set the target arc path functions as (3-1) and (3-2).

$$x(s) = x_0 + r \cos\left(\sigma \frac{s}{r} + \gamma\right) \quad (3-1)$$

$$y(s) = y_0 + r \sin\left(\sigma \frac{s}{r} + \gamma\right) \quad (3-2)$$

where  $s$  is the length of arc,  $r$  is the radius of arc,  $\gamma$  is the inclination angle of arc.  $c(x_0, y_0)$  is the center of arc.  $\sigma$  determines the move direction of arc,  $\sigma = 1$  means counterclockwise direction,  $\sigma = -1$  means clockwise direction.  $\theta$  is central angle,  $0 < \theta \leq 2\pi$  and  $\theta = s/r$ .

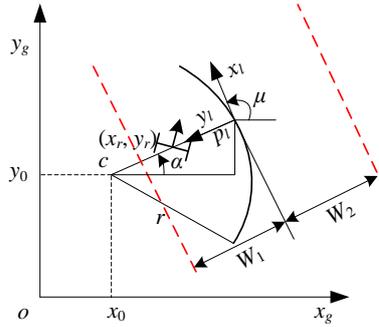


Fig. 3. 2: Global and local coordinate systems

The global and local coordinate systems are shown in Fig. 3.2.  $o-x_g y_g$  is the global coordinate system,  $p_l-x_l y_l$  is the local coordinate system.  $(x_r, y_r, \phi)$  is the pose of robot in the global coordinate system.  $c(x_0, y_0)$  is the central point of arc. The angle between the line which passes through  $(x_r, y_r)$  and  $c(x_0, y_0)$  and  $x$ -axis is name as  $\alpha$  that expressed as (3-3). The width of passage inside of the arc is  $W_1$  and  $W_1 < r$ , the opposite side is  $W_2$ .

$$\alpha = \tan^{-1}\left(\frac{y_r - y_0}{x_r - x_0}\right) \quad (3-3)$$

The theory of PGR-S along straight passage can be used for arc passage in a small interval  $d_t$  in the local coordinate system  $p_l-x_ly_l$ . In order to simplify computation, we select the intersection point  $p_l(x_l, y_l)$ , expressed as (3-4), (3-5), the line through  $c(x_0, y_0)$  and  $(x_r, y_r)$  with the arc as the origin of the local coordinate system.  $x_l$ -axis is the tangential direction of the arc,  $y_l$ -axis is the normal direction.  $\mu$  means the rotation angle that the local coordinate system relatives to global coordinate system from (3-6).

$$x_l = x_0 + r \cos \alpha \quad (3-4)$$

$$y_l = y_0 + r \sin \alpha \quad (3-5)$$

$$\mu = \frac{\pi}{2} + \alpha \quad (3-6)$$

Therefore, we can obtain the pose of robot  $(x'_r, y'_r, \phi')$  in the local coordinate system.

$$x'_r = x_r \cos \mu + y_r \sin \mu - x_l \cos \mu - y_l \sin \mu \quad (3-7)$$

$$y'_r = -x_r \sin \mu + y_r \cos \mu + x_l \sin \mu - y_l \cos \mu \quad (3-8)$$

$$\phi' = \phi - \mu \quad (3-9)$$

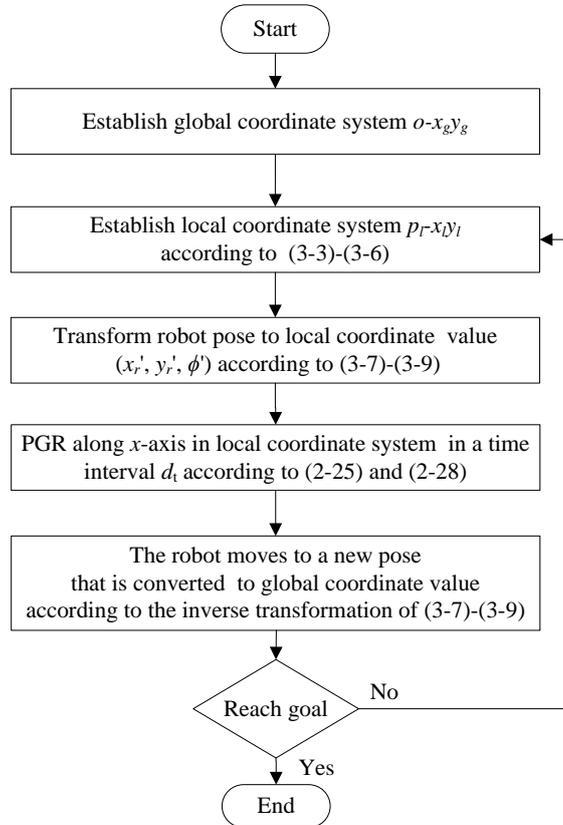


Fig. 3. 3: Flow chart under PGR-A along arc passage

The flow chart is shown in Fig. 3.3. We establish global coordinate system  $o-x_g y_g$  and local coordinate system  $p_l-x_l y_l$  according to (3-3) - (3-6) firstly, Then the pose of robot  $(x_r, y_r, \phi)$  is converted to the local coordinate value  $(x'_r, y'_r, \phi')$  according to (3-7) - (3-9). Within the time interval  $d_t$ , the PGR-S along  $x$ -axis of local coordinate system is to control robot and a new pose is obtained according to (2-25) and (2-28). Then the new pose of robot is converted to global coordinate value according the inverse transformation of (3-7) - (3-9). Finally, the robot judges whether the new location is the goal or not. If the new location is not the goal, the new local coordinate system is needed to establish based on the new robot pose and the center of arc. Program is running along this cycle process until the robot reaches the goal.

### 3.2 The Improved PGR-A with the Adjustable Look-ahead Method

The PGR-S is extended to drive the robot move along arc passage based on the straight passage theory in Section 3.1. The improved PGR-A with the adjustable look-ahead method is proposed to make the robot's trajectory converge to the target arc.

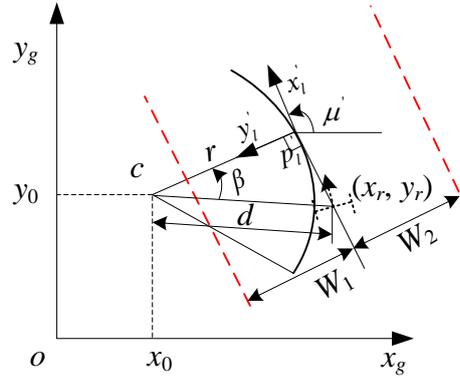


Fig. 3. 4: Global and local coordinate systems for the improved PGR-A with the adjustable look-ahead method

Global and local coordinate systems with the adjustable look-ahead method are shown in Fig. 3.4. The distance between the robot location  $(x_r, y_r)$  and the center of the arc  $c(x_0, y_0)$  is assumed as  $d$  in (3-10). The central angle between robot location and local coordinate origin is assumed as  $\beta$  in (3-11), which is named as adjustable look-ahead central angle, because  $\beta$  will change if  $d$  changes.

$$d = \sqrt{(x_r - x_0)^2 + (y_r - y_0)^2} \quad (3-10)$$

$$\beta = \cos^{-1}\left(\frac{r}{d}\right) \quad (3-11)$$

The origin  $p'_l(x'_l, y'_l)$  which is described in (3-12), (3-13) in the new local coordinate system is the tangent point of the arc at the front of the robot location, whose tangent line passes through the  $(x_r, y_r)$ .  $x'_l$ -axis is the tangential direction of the arc,  $y'_l$ -axis is the normal direction.  $\mu'$  means the rotation angle that the new local coordinate system relates to the global coordinate system.

$$x'_l = x_r + d \sin \beta \cos\left(\frac{3\pi}{2} - \beta - \alpha\right) \quad (3-12)$$

$$y'_l = y_r + d \sin \beta \sin \left( \frac{3\pi}{2} - \beta - \alpha \right) \quad (3-13)$$

$$\mu' = \frac{\pi}{2} + \alpha + \beta \quad (3-14)$$

The local coordinate system is established at the front of robot location and is adjusted with the change of  $\beta$ , so this method is named as adjustable look-ahead method.

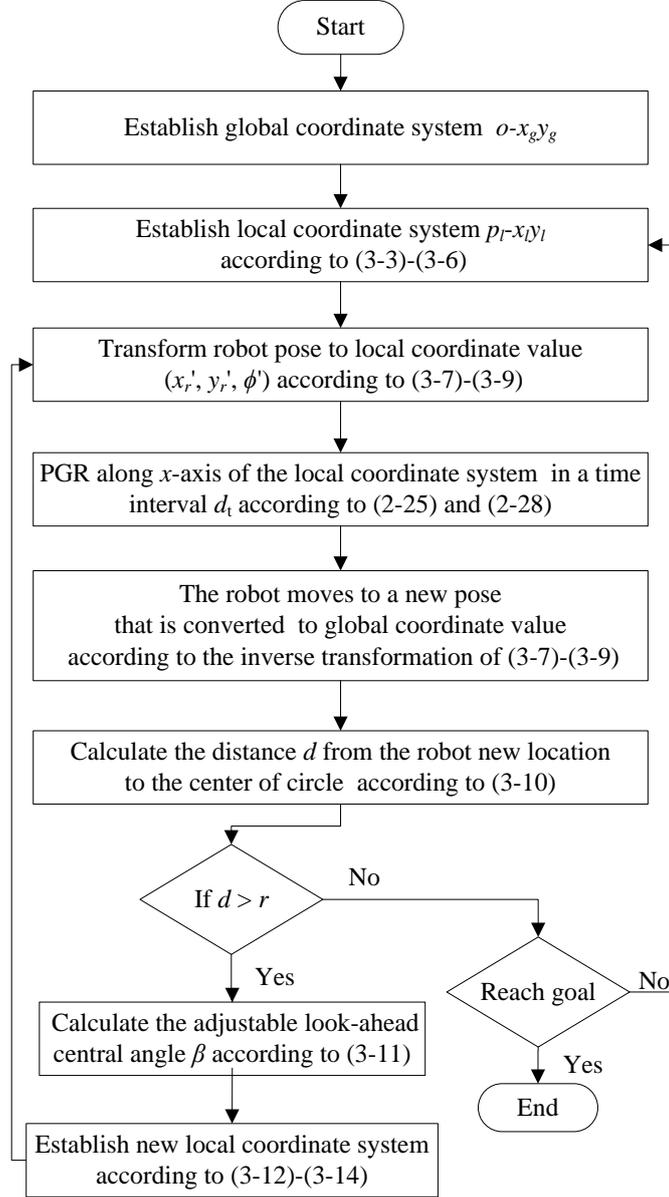


Fig. 3. 5: Flow chart under the improved PGR-A with the adjustable look-ahead method along arc passage

The flow chart under the improved PGR-A with the adjustable look-ahead method along arc passage is shown in Fig. 3.5. The difference with the method in Section 3.1 is the introduction of  $\beta$ . When the robot's new location is outside of target arc, i.e.,  $d > r$ , the program calculates  $\beta$ , then the new local coordinate system is established based on (3-12) - (3-14). Otherwise, when the robot's new

location is inside of target arc, i.e.,  $d < r$ , the process is same with the flow chart in Fig. 3.3.

### 3.3. Numerical Simulations

To confirm the efficiency of the proposed PGR-A method in Section 3.1 and Section 3.2 for the robot's path tracking along arc passage, the numerical simulations are performed with counterclockwise and clockwise in several aspects respectively.

We investigate the influence of coefficient constant  $\lambda$  and the time interval  $d_t$  to the robot's trajectory. In addition, limited random disturbances are imposed to the inputs of velocity  $u_1$  and steering angle  $u_2$  to investigate the robustness of the two methods. The overall simulation results are discussed finally. In the simulation, the simulation time  $t$  is set separately as 40 seconds in arc passage and 50 seconds in the S-shaped passage, that is to say, when the simulation time  $t$  is equal to 40 seconds or 50 seconds, the robot receives a command of velocity  $u_1=0$  and steering angle  $u_2=0$  and stop.

Table 3. 1: Default parameter values for all the numerical simulations.

Parameters	Values	Unit
$r$	4	m
$W_1, W_2$	2.5	m
$a$	1	-
$\varepsilon$	0.001	-
$V_m$	1	m/s
$c$	1	-
$c_m$	1	-
$k_m$	1	-
$\gamma$	$\pi/4$	rad
$\theta$	$2\pi$	rad
Initial condition	$(0, -1, -\pi/4)$	-

The numerical simulations are performed along the counterclockwise and clockwise direction with the software MATLAB 7.11.0 (The MathWorks, Inc.). The default parameter values for the simulations are listed in Table 3. 1. In the simulation results, the red dashed line of (c) or (d) represents the target arc. The red cross represents the center of arc. The robot's trajectories are shown in the  $x$ - $y$  plain. Initial condition is designated by an icon with an arrow. The thick arrow represents the direction of the robot's motion.

#### 3.3.1 The Influence of Coefficient Constant $\lambda$

In the numerical simulation process, we only change the value of coefficient constant  $\lambda$ , and keep the other parameters constant. The parameters are listed in Table 3. 2.

The robot's trajectories along the counterclockwise direction and control commands of PGR-A proposed in Section 3.1 at different  $\lambda$  values are depicted in Fig. 3.6. The value of  $\lambda$  is 0.2 for (a), 0.5 for (b), and 0.9 for (c). As described in (8),  $\lambda$  is a coefficient constant for adjusting the response speed of  $u_2$ , therefore, the trajectory tends to converge the target circle if  $\lambda$  becomes large. It is found

Table 3. 2: Parameter values for numerical simulations with different coefficient constant.

No.	$\lambda$	$d_t$	$t$	Color
(a)	0.2	0.1s	40s	green
(b)	0.5	0.1s	40s	magenta
(c)	0.9	0.1s	40s	blue

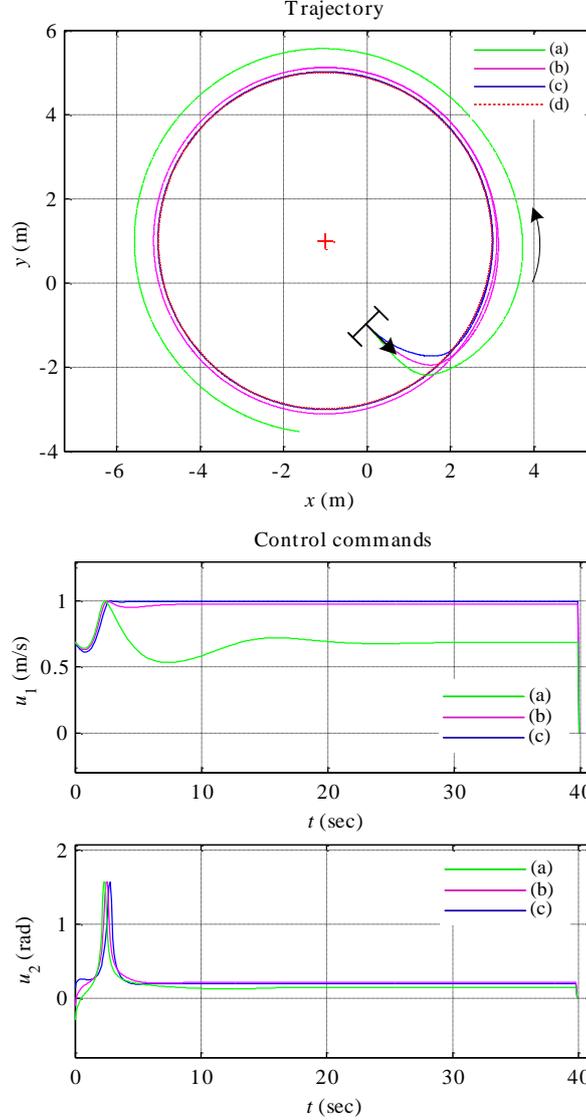


Fig. 3. 6: Trajectories in the  $x$ - $y$  plain along the counterclockwise direction and time response of control commands  $u_1$ ,  $u_2$  by the method in Section 3.1 at different values of  $\lambda$ , from (a) to (c).

from the top of Fig. 3.6 that the convergence property is the best when  $\lambda$  is 0.9, but the trajectory doesn't yet converge to the target arc.

Figure 3.7 depicts the robot's trajectories along the counterclockwise direction and control commands of the improved PGR-A with the adjustable look-ahead method at different  $\lambda$  values. The value of  $\lambda$  is 0.2 for (a), 0.5 for (b), and 0.9 for (c). It is observed from the top of Fig. 3.7 that even if  $\lambda$  is large or small, the trajectories can converge to the target arc well, which indicates that the value of  $\lambda$  has no influence to the convergence property. The improved PGR-A with the adjustable look-

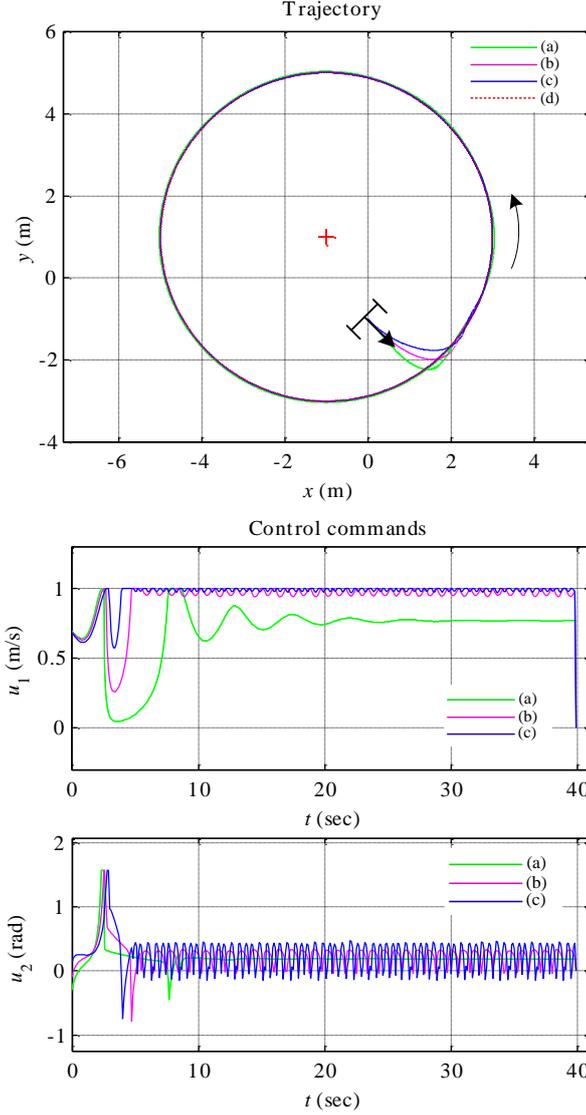


Fig. 3. 7: Trajectories in the  $x$ - $y$  plain along the counterclockwise direction and time response of control commands  $u_1$ ,  $u_2$  by the PGR-A with the adjustable look- ahead method at different values of  $\lambda$ , from (a) to (c).

ahead method has improved the performance of path tracking along arc passage for robot. In the bottom two graphs, the values of  $u_1$  and  $u_2$  have small fluctuation after 5 seconds when  $\lambda$  is 0.5 and 0.9, respectively. The fluctuation range of  $u_1$  is 0.04 m/s when  $\lambda$  is 0.5, and 0.09 m/s when  $\lambda$  is 0.9. Because the fluctuation range is very little, it can be ignored. The fluctuation range of  $u_2$  is 0.3 rad when  $\lambda$  is 0.5, and 0.5 rad when  $\lambda$  is 0.9. It is considered that the fluctuation is caused by the local coordinate system changing constantly, and fluctuation range of  $u_2$  has proportional relation with  $\lambda$ .

The robot's trajectories along the clockwise direction and control commands of PGR-A proposed in Section 3.1 at different  $\lambda$  values are depicted in Fig. 3.8. The value of  $\lambda$  is 0.2 for (a), 0.5 for (b), and 0.9 for (c). The results have the same property with the trajectory along the counterclockwise direction in Fig. 3.6.

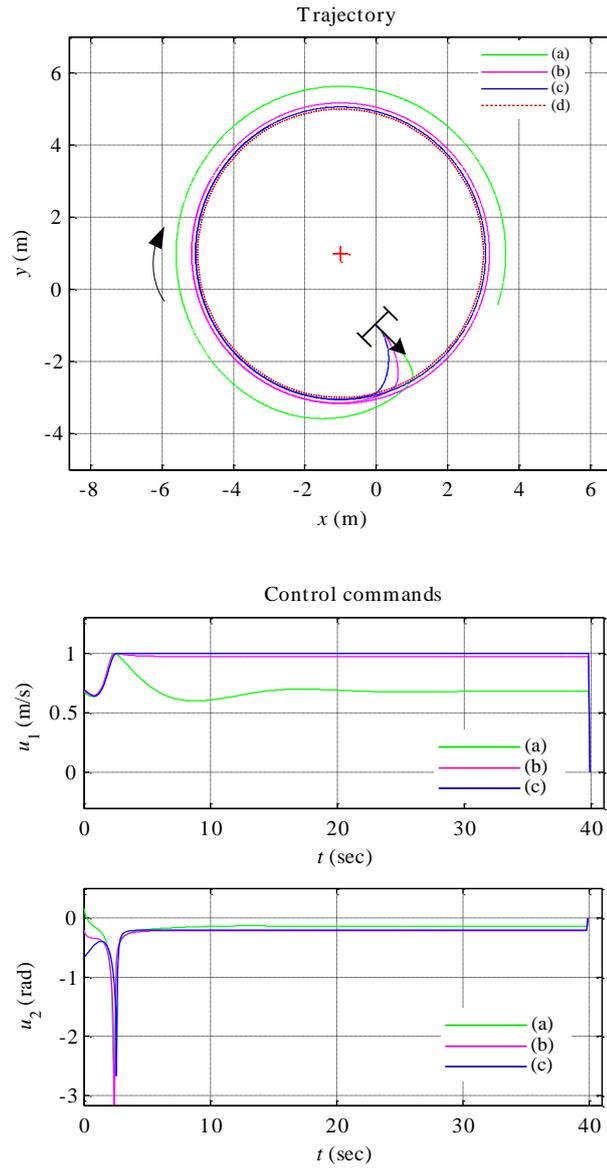


Fig. 3. 8: Trajectories in the  $x$ - $y$  plain along the clockwise direction and time response of control commands  $u_1$ ,  $u_2$  by the method in Section 3.1 with clockwise at different values of  $\lambda$ , from (a) to (c).

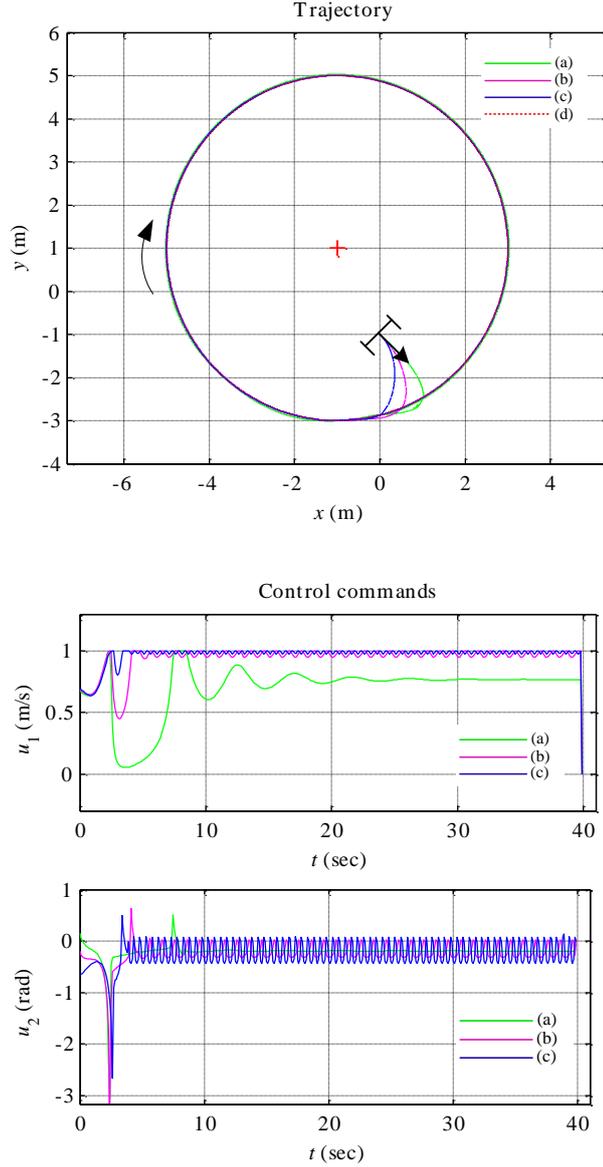


Fig. 3. 9: Trajectories in the  $x$ - $y$  plain along the clockwise direction and time response of control commands  $u_1$ ,  $u_2$  by the PGR-A with the adjustable look- ahead method at different values of  $\lambda$ , from (a) to (c).

Figure 3.9 depicts the robot's trajectories along the clockwise direction and control commands of the improved PGR-A with the adjustable look-ahead method at different  $\lambda$  values. The value of  $\lambda$  is 0.2 for (a), 0.5 for (b), and 0.9 for (c). It is the same with the results along the counterclockwise in Fig. 3.7. It is observed from the top of Fig. 3.9 that even if  $\lambda$  is large or small, the trajectories can converge to the target arc well, which indicates that the value of  $\lambda$  has no influence to the convergence property in the improved PGR-A with the adjustable look-ahead method.

### 3.3.2 The Influence of Time Intervals $d_t$

In this numerical simulation, three values of time interval  $d_t$  are given. The parameters are listed

in Table 3. 3.

The robot's trajectories along the counterclockwise direction and control commands of PGR-A proposed in Section 3.1 at different time intervals  $d_t$  are depicted in Fig. 3.10. The value of  $d_t$  is 0.1s for (a), 0.5s for (b), and 1s for (c). When the value of time interval  $d_t$  becomes short, the move distance of the robot becomes short and the orientation is unchange. The results point out that the

Table 3. 3: Parameter values for numerical simulations in different time intervals.

No.	$d_t$	$\lambda$	$t$	Color
(a)	0.1s	0.5	40s	green
(b)	0.5s	0.5	40s	magenta
(c)	1s	0.5	40s	blue

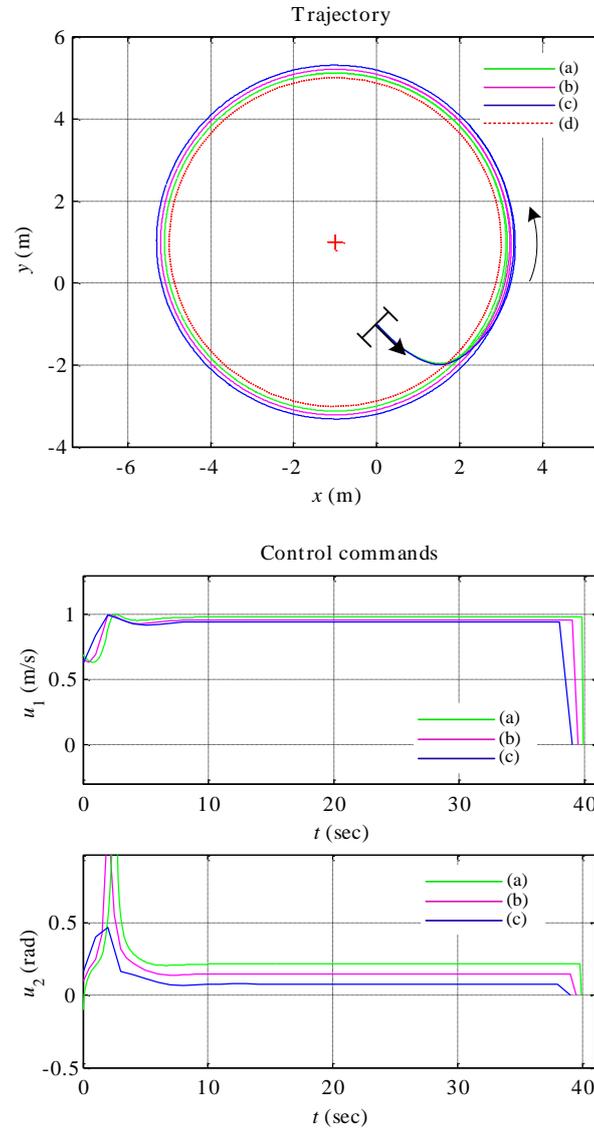


Fig. 3. 10: Trajectories in the  $x$ - $y$  plain along the counterclockwise direction and time response of control commands  $u_1$ ,  $u_2$  by the method in Section 3.1 at different time interval values of  $d_t$ , from (a) to (c).

convergence property tends to be better when the time interval  $d_t$  becomes shorter, but the trajectory isn't able to converge to the target arc. The velocity  $u_1$  tends to be stable after 8 seconds. The steering angle  $u_2$  tends to be stable after 6 seconds.

Figure 3.11 depicts the robot's trajectories along the counterclockwise direction and control commands by improved PGR-A with the adjustable look-ahead method at different  $d_t$  values. The value of  $d_t$  is 0.1s for (a), 0.5s for (b), and 1s for (c). From the top of figure, we discover when the value of  $d_t$  is 1s, the trajectory in the interval of  $d_t$  tends to become straight line, the trajectory fluctuates around the target arc, which indicates that the convergence property becomes poor when  $d_t$  becomes long. If we chose the proper  $d_t$  value, the robot trajectory is able to converge to the target arc perfectly. The bottom two graphs show that the fluctuation range of velocity

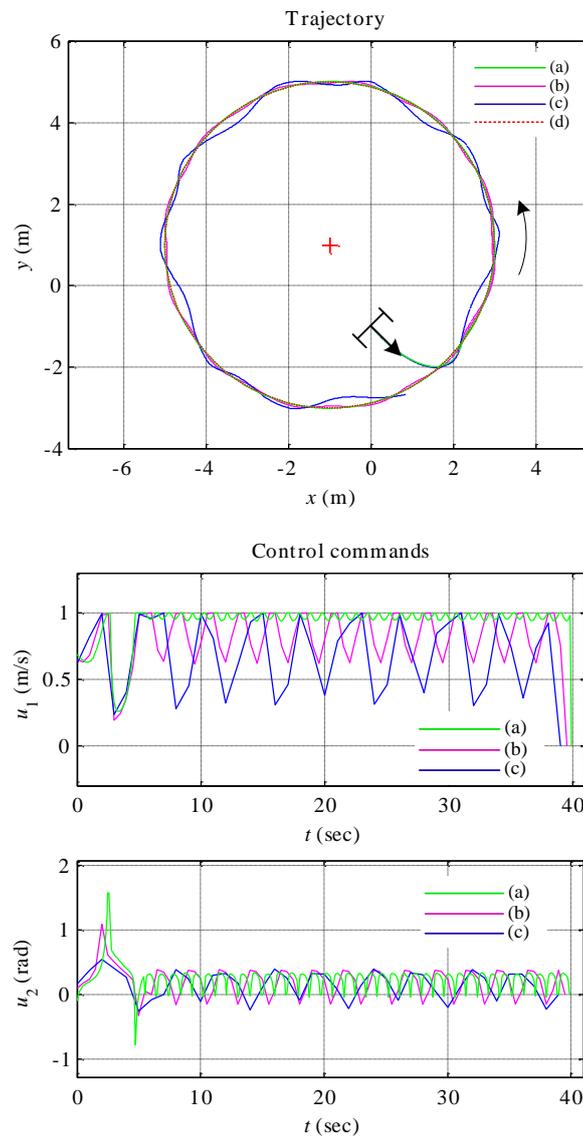


Fig. 3. 11: Trajectories in the  $x$ - $y$  plain along the counterclockwise direction and time response of control commands  $u_1$ ,  $u_2$  by improved PGR-A with the adjustable look-ahead method at different time interval values of  $d_t$ , from (a) to (c).

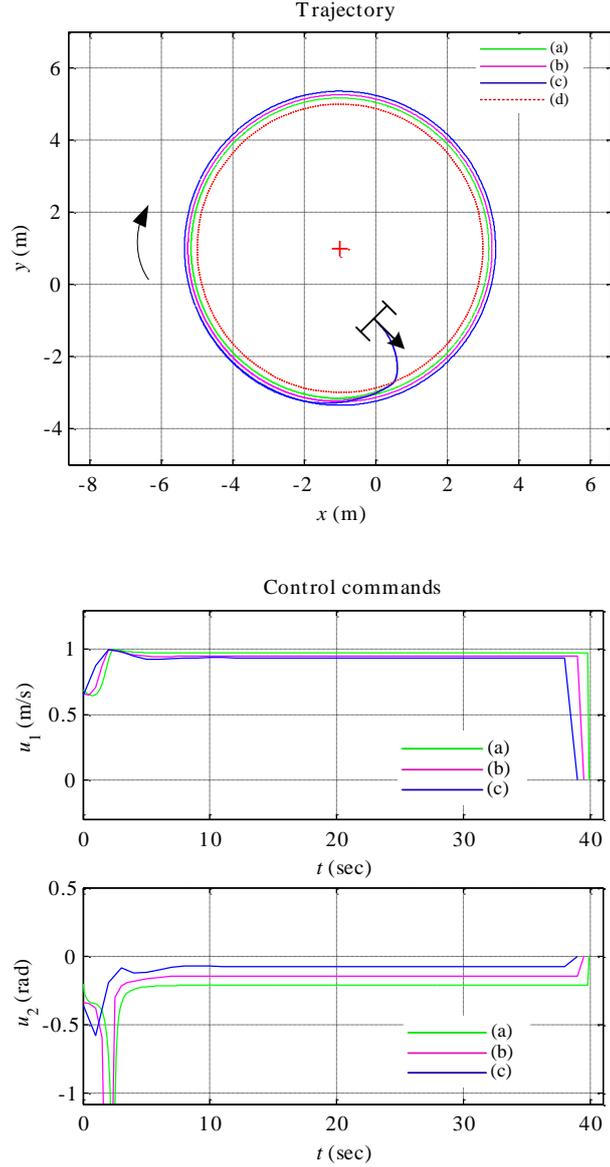


Fig. 3. 12: Trajectories in the  $x$ - $y$  plain along the clockwise direction and time response of control commands  $u_1$ ,  $u_2$  by the method in Section 3.1 at different time interval values of  $d_t$ , from (a) to (c).

$u_1$  tends to be wide when the value of  $d_t$  becomes long. The steering angles  $u_2$  have the same fluctuation range with the different values of  $d_t$ , but the fluctuation period tends to be long when  $d_t$  becomes long.

The robot's trajectories along the clockwise direction and control commands of PGR-A proposed in Section 3.1 at different time intervals  $d_t$  are depicted in Fig. 3.12. The value of  $d_t$  is 0.1s for (a), 0.5s for (b), and 1s for (c). It is similar with the result of Fig. 3.10 along the counterclockwise direction. When the value of time interval  $d_t$  becomes short, the move distance of the robot becomes short and the orientation is unchange. The results point out that the convergence property tends to be better when the time interval  $d_t$  becomes shorter, but the trajectory is not able to converge to the target arc.

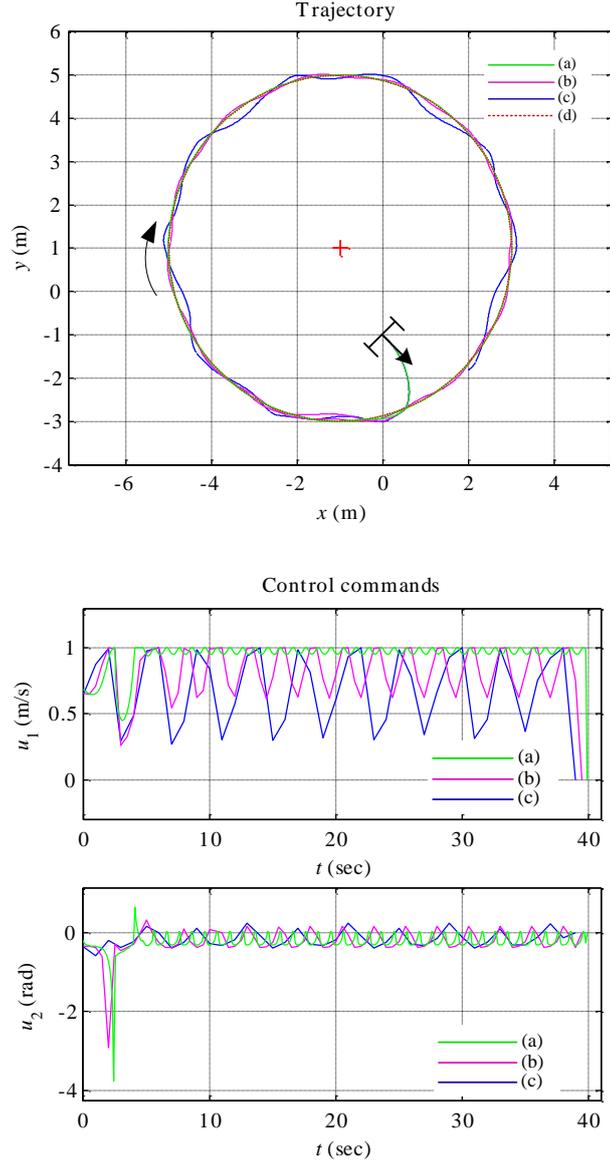


Fig. 3. 13: Trajectories in the  $x$ - $y$  plain along the clockwise direction and time response of control commands  $u_1$ ,  $u_2$  by improved PGR-A with the adjustable look-ahead method at different time interval values of  $d_t$ , from (a) to (c).

Figure 3.13 depicts the robot's trajectories along the clockwise direction and control commands by improved PGR-A with the adjustable look-ahead method at different  $d_t$  values. The value of  $d_t$  is 0.1s for (a), 0.5s for (b), and 1s for (c). The results are similar with the Fig. 3.11. it indicates that the convergence property becomes poor when  $d_t$  becomes long. If we chose the proper  $d_t$  value, the robot trajectory is able to converge to the target arc perfectly. The bottom two graphs show that the fluctuation range of velocity  $u_1$  tends to be wide when the value of  $d_t$  becomes long. The steering angles  $u_2$  have the same fluctuation range with the different values of  $d_t$ , but the fluctuation period tends to be long when  $d_t$  becomes long.

### 3.3.3 The Influence with Disturbance Imposed on the Inputs of $u_1$ and $u_2$

In fact, because of the restrictions of robot itself and external environment, the inputs are often accompanied by some disturbance. For instance, when the robot runs on uneven ground, the steering wheel may receive disturbance force from the ground. To investigate the robustness of the two proposed PGR-A methods, two different sizes limited random disturbance  $e_1$  and  $e_2$  (listed in Table 3. 4) are imposed to the inputs of velocity command  $u_1$  and steering angle command  $u_2$ , respectively.

Table 3. 4: Parameter values for numerical simulations imposed disturbance.

No.	$e_1, e_2$	$\lambda$	$d_t$	$t$	Color
(a)	$0.2 \times \text{rand}(1) - 0.1$	0.5	0.1s	40s	blue
(b)	$0.5 \times \text{rand}(1) - 0.25$	0.5	0.1s	40s	green

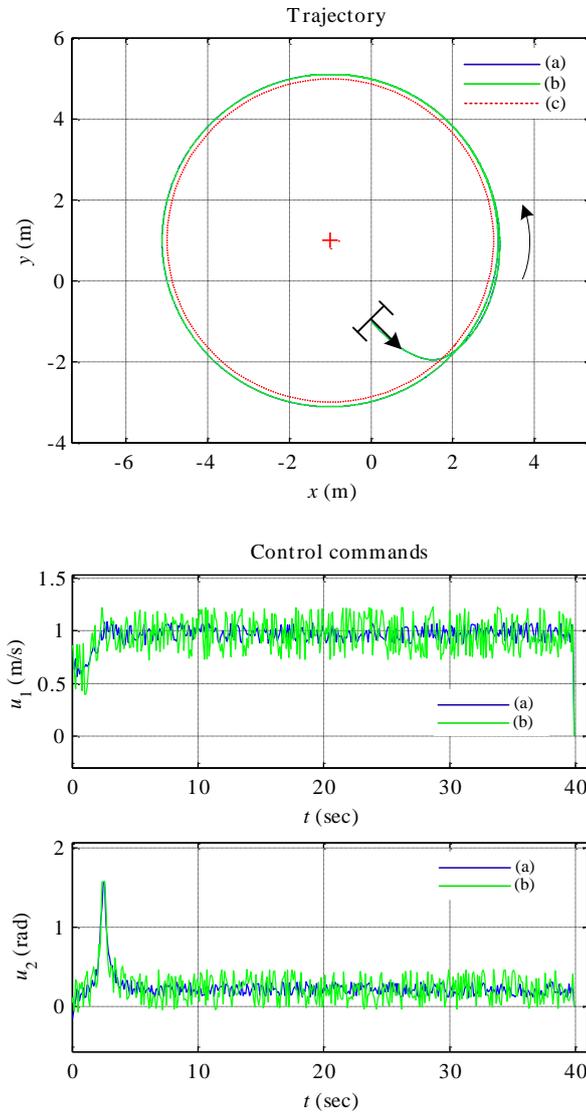


Fig. 3. 14: Trajectories in the  $x$ - $y$  plain along the counterclockwise direction and time response of control commands of PGR-A proposed in Section 3.1 at different disturbance values of  $e_1$ ,  $e_2$ , from (a) to (b).

The robot's trajectories along the counterclockwise direction and control commands by the method proposed in Section 3.1 at different disturbances are depicted in Fig. 3.14. The trajectories are same with the two different disturbances, which indicates that this method has strong capacity of resisting disturbance. The fluctuation range of control commands tends to be wide with the increasement of disturbance.

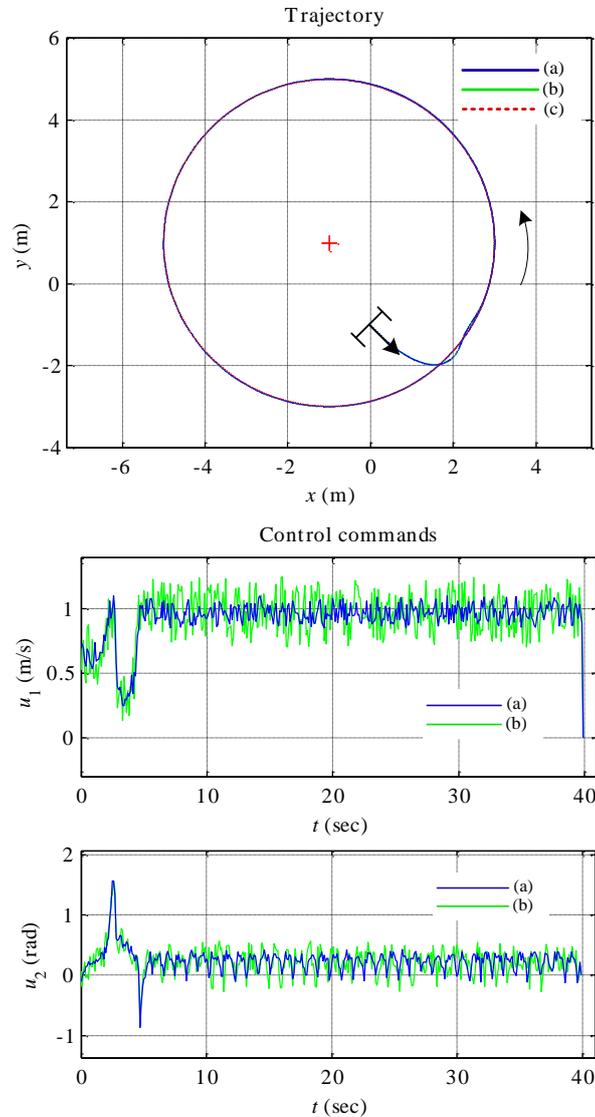


Fig. 3. 15: Trajectories in the  $x$ - $y$  plain along the counterclockwise direction and time responses of control commands by improved PGR-A with the adjustable look-ahead method for robot at different disturbance values of  $e_1$  and  $e_2$ , from (a) to (b).

Similarly, Figure 3.15 depicts the robot's trajectories along the counterclockwise direction and control commands by the improved PGR-A with the adjustable look-ahead method at different disturbances. The results point out that the trajectories converge to the target perfectly with the two different disturbances, which indicates that the improved PGR-A has strong capacity of resisting disturbance as well. The fluctuation range of control commands tends to be wide with the increasement of disturbance.

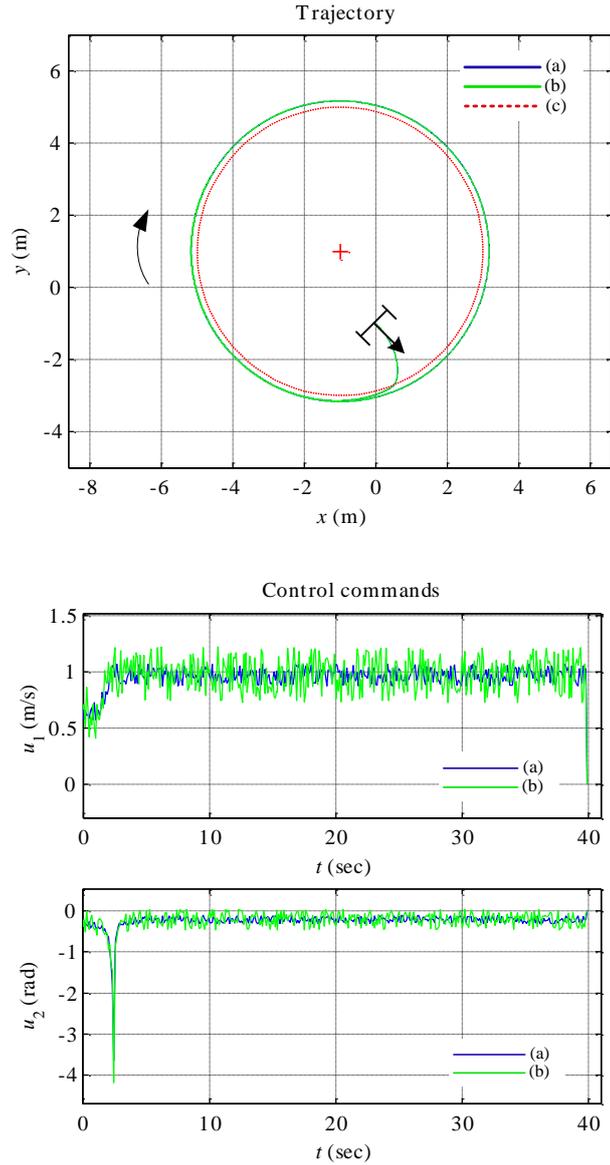


Fig. 3. 16: Trajectories in the  $x$ - $y$  plain along the clockwise direction and time response of control commands of PGR-A proposed in Section 3.1 at different disturbance values of  $e_1$ ,  $e_2$ , from (a) to (b).

The robot's trajectories along the clockwise direction and control commands by the method proposed in Section 3.1 at different disturbances are depicted in Fig. 3.16. the results indicate that this method has strong capacity of resisting disturbance. The fluctuation range of control commands tends to be wide with the increasement of disturbance.

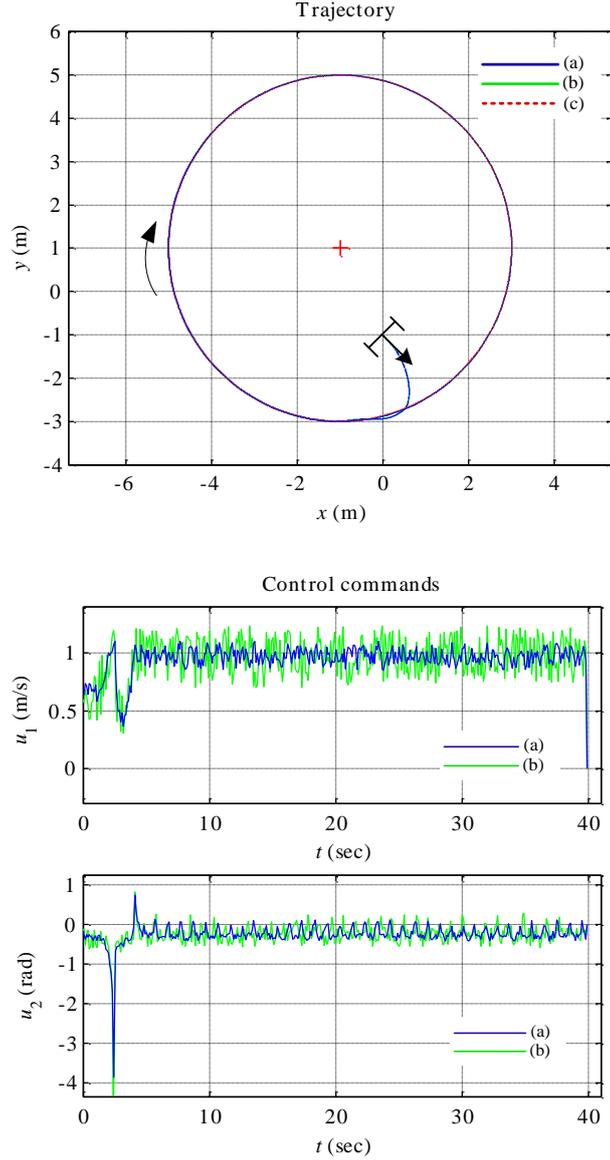


Fig. 3. 17: Trajectories in the  $x$ - $y$  plain along the clockwise direction and time responses of control commands by improved PGR-A with the adjustable look-ahead method for robot at different disturbance values of  $e_1$  and  $e_2$ , from (a) to (b).

Similarly, Figure 3.17 depicts the robot's trajectories along the clockwise direction and control commands by the improved PGR-A with the adjustable look-ahead method at different disturbances. The results point out that the trajectories converge to the target perfectly with the two different disturbances, which indicates that the improved PGR-A has strong capacity of resisting disturbance as well. The fluctuation range of control commands tends to be wide with the increasement of disturbance.

### 3.3.4. The PGR-A and Improved PGR-A with Adjustable Look-ahead Method to Track the S-shaped Passage.

In this Section we apply that the PGR-A and the improved PGR-A with the adjustable look-ahead

method applied to track the S-shaped passage.

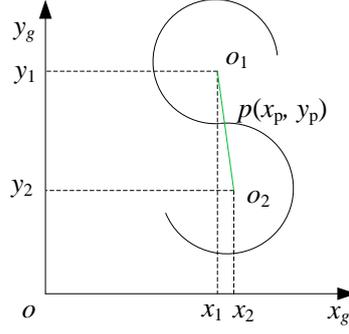


Fig. 3. 18: The S-shaped model

Shown as Fig. 3.18, the S-shaped passage consists of two arcs. The centers of two arcs are  $o_1(x_1, y_1)$  and  $o_2(x_2, y_2)$ , the radiuses are  $r_1$  and  $r_2$ , the intersection point of two arcs is  $p(x_p, y_p)$  that can be solved by (3-15) and (3-16).

$$x_p = \frac{1}{2}(x_1 + x_2) \quad (3-15)$$

$$y_p = \frac{1}{2}(y_1 + y_2) \quad (3-16)$$

Firstly, the robot moves along the arc  $o_1$  countclockwise, when it reaches the intersection point  $p$ , then starts from  $p$  and moves along the arc  $o_2$  clockwise.

The condition (a) does not consider error in input, the condition (b) imposes the disturbance on the inputs of  $u_1$  and  $u_2$ . The default parameter values for numerical simulations are shown in Table 3. 5.

Table 3. 5: Parameter values for numerical simulations for S-shaped passage

No.	Method	$\lambda$	$d_t$	$t$	Color
(a)	The PGR-A	0.35	0.1s	50s	blue
(b)	The improved PGR-A	0.35	0.1s	50s	green

Figure 3.19 depicts the robot's trajectories and control commands by the PGR-A and the improved PGR-A with adjustable look-ahead method to track the arc passage. As can be seen in the upper graph, the trajectory by the PGR-A is distributed in the outside of the target S-shaped passage, the trajectories converge to the target perfectly by the improved PGR-A with adjustable look-ahead method. In the below graph, around the intersection point  $p$ , the steering angle have two sudden changes from 29 seconds to 31 seconds. The first sudden change depicts that the trajectory switches from arc  $o_1$  to  $o_2$ , the second sudden change depicts that the trajectory changes from inside to outside of  $o_2$ . The steering angle of  $u_2$  has only a sudden change in 28 seconds at intersection point  $p$ , which indicates that the improved PGR-A method with adjustable look-ahead method has well performance to converge the target S-shape passage. Similarly, the PGR-A and the improved PGR-A also can be applied in the complicate passage which consists of many arcs.

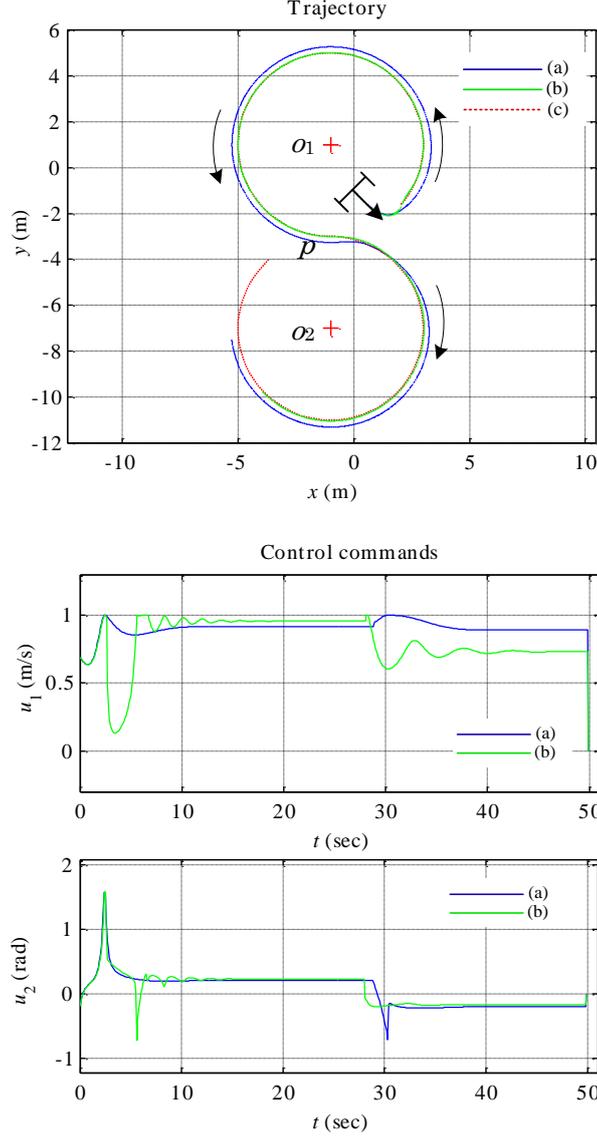


Fig. 3. 19: Trajectories in the  $x$ - $y$  plain and time responses of control commands by the PGR-A in (a) and the improved PGR-A with the adjustable look-ahead method in (b).

### 3.3.5 Discussion on Numerical Simulation Results

According to the above simulation results, we discuss the convergence property of the two methods for two-wheeled robot in this Section.

The first issue concerns the influence of  $\lambda$  in the two methods. By the PGR-A method in Section 3.1, the trajectory is closer to the target circle if the value of  $\lambda$  is larger, but the trajectory does not converge to the target arc. However, by the improved PGR-A with adjustable look-ahead method, even if  $\lambda$  is large or small, the robot trajectory is able to converge to the target arc perfectly. The change of the value of  $\lambda$  has no influence to the convergence property. The improved PGR-A with the adjustable look-ahead method for two-wheeled robot has better performance.

The second issue concerns the influence of time interval  $d_t$ . By the PGR-A method in Section 3.1, convergence property has some improvement as  $d_t$  becomes short, but the trajectory isn't able to

converge to the target arc. However, by the improved PGR-A with adjustable look-ahead method, when  $d_t$  becomes long, the trajectory fluctuates around the target arc. If we choose the proper  $d_t$  value, the robot trajectory is able to converge to the target arc perfectly.

The third issue concerns the capacity of resisting disturbance by the two methods. The two sizes of bound random disturbances are imposed, the results point out two methods have strong capacity of resisting disturbance.

The fourth issue concerns the application in S-shaped passage. The simulations are performed on the PGR-A and the PGR-A with the adjustable look-ahead method. The results show that robot trajectory is consistent with the target S-shaped passage on the improved PGR-A with adjustable look-ahead method.

In general, the improved PGR-A with adjustable look-ahead method has better performance along arc passage. Compared with the other research, like references [13-20], the improved PGR-A with the adjustable look-ahead method has not only the merits of convergence perfectly, but also the strong capacity of resisting disturbance. However, this method has the limitation in application that is just for arc passage problem. It will be extended to the complicate passage environment in the future work.

### **3.4 Conclusion**

This section presented the PGR-A and the improved PGR-A with the adjustable look-ahead method to track the arc passage for two-wheeled robots. The robot's trajectory tended to deviate outward from the target circle in the first method. To improve this weakness, the adjustable coefficient angle  $\beta$  is introduced. The robot adjusted the local coordinate system constantly according to the deviate between the trajectory and the target arc, which made the trajectory converge to the target arc perfectly. The simulation results also verified the improved PGR-A with adjustable look-ahead method had better performance on path tracking along the arc passage. This approach was also validated on the S-shaped passage, which indicated that it could be applied to multi-arc passage tracing problem.

We focused on the control method to track the arc passage and verify the efficiency on simulation environment without considering obstacles. However it does not mean that we neglect the obstacle avoidance problem. Actually, some members in our research group have proposed and are testing the obstacle avoidance algorithms with the PGR-A. In the future work, we will perform the experiments to verify the effectiveness of the method proposed in this paper in the real environment in the future.

## **Chapter 4**

### **The Extended PGR Converge to the Origin (PGR-O)**

#### **for Car-like Robots**



In this Section, we briefly introduce the PGR for car-like robot converge to origin (PGR-O). The purpose of PGR algorithm is to make main direction angle converge to the target direction angle and location converge to origin.

#### 4.1 Car-like Robots State Equations

The car-like robot is steering by the front wheel, and driving by the rear wheels. From the driver's point of view, a car-like robot has two controls, the velocity and the steering angle. The reference point with coordinates  $(x, y)$  is the midpoint of the rear wheels. The distance between rear and front axles was assumed as  $L$  and denote  $u_2$  as the steering angle of the front wheels and the main direction angle  $\theta$  of the car-like robot in global coordinates. Moreover a mechanical constraint imposes  $|u_2| < u_{2\max}$ . In the PGR-O, the steering angle theoretical maximum value is  $u_{2\max} = \pi/2$ .

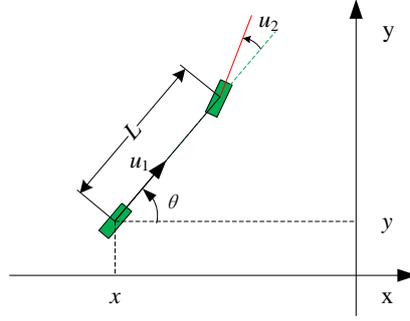


Fig. 4. 1: The two-wheel model is equivalent to the four-wheel car-like robot

We assume the two wheel model that is equivalent to the four-wheel car-like robot was shown in Fig. 4.1. The state equations of the car-like robot are expressed as (4-1), (4-2) and (4-3).

$$\dot{x} = u_1 \cos(\theta) \quad (4-1)$$

$$\dot{y} = u_1 \sin(\theta) \quad (4-2)$$

$$\dot{\theta} = \frac{\tan(u_2)}{L} u_1 \quad (4-3)$$

#### 4.2 Steering Angle and Velocity Control

The error between direction angle and target direction angle was defined as (4-4)

$$e = \theta - \theta_r \quad (4-4)$$

For regulation on the target direction angle, the  $e$  has to converge to 0. The steering angle  $u_2$  use the follow (4-5) in control law.

$$u_2 = \tan^{-1} \left\{ \frac{L}{u_1} \left[ -\lambda e + \frac{\partial \theta_r}{\partial x} u_1 \cos(e + \theta_r) + \frac{\partial \theta_r}{\partial y} u_1 \sin(e + \theta_r) \right] \right\} \quad (u_1 \neq 0) \quad (4-5)$$

Where,  $\lambda > 0$ , based on the (4-5), the first delay system was built as (4-6)

$$\dot{e} = \begin{cases} -\lambda e & u_1 \neq 0 \\ 0 & u_1 = 0 \end{cases} \quad (4-6)$$

A Lyapunov function with parameters of  $\lambda_1$  and  $\lambda_2$  was used which guarantees the stability of car-

like robot shows as (4-7).

$$V = \frac{1}{2}(e^2 + \lambda_1 x^2 + \lambda_2 y^2) \quad (4-7)$$

Where,  $\lambda_1, \lambda_2 > 0$ , the system trajectory perform of differentiation to time  $t$  and substitute to (4-6).

$$\frac{d}{dt}V = \begin{cases} -\lambda e^2 + (\lambda_1 x \cos(e + \theta_r) + \lambda_2 y \sin(e + \theta_r))u_1 & u_1 \neq 0 \\ 0 & u_1 = 0 \end{cases} \quad (4-8)$$

where, we defines the  $u_1$  as (4-9),

$$u_1 = -\lambda_1 x \cos(e + \theta_r) - \lambda_2 y \sin(e + \theta_r) \quad (4-9)$$

(4-9) was substituted to (4-8),

$$\frac{d}{dt}V = \begin{cases} -\lambda e^2 - u_1^2 & u_1 \neq 0 \\ 0 & u_1 = 0 \end{cases} \quad (4-10)$$

The value of the (4-10) were negative expect origin. Therefore, control system is stable with the Lyapunov function, and the robot is converging no matter in anywhere.

### 4.3 Path Function Group

Conference the paper [9], (4-11) is used for the path function group in PGR-O.

$$y = ax^n \quad (4-11)$$

Where,  $a$  is a positive real constant number,  $n$  is a positive integer.

Differential to the both side of (4-11) on the  $x$  can get the (4-12)

$$\frac{dy}{dx} = anx^{n-1} \quad (4-12)$$

The target direction angle shows in (4-12) was derived from (4-11).

$$\theta_r = \tan^{-1}\left(\frac{ny}{x}\right) \quad (4-13)$$

Fig. 4.2 is the target pose angle at lattice points when  $n=2$ .

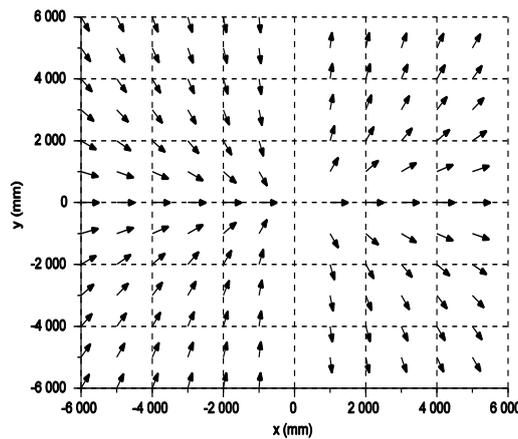


Fig. 4. 2: Target pose angle  $\theta_r$  at lattice points when  $n=2$

Based on (4-13), the (4-9) and (4-5) can be changed to (4-14) and (4-15)

$$u_1 = -(\lambda_1 x \cos(\theta) + \lambda_2 y \sin(\theta)) \quad (4-14)$$

$$u_2 = \tan^{-1} \left[ -\frac{L\lambda}{u_1} \left\{ \theta - \tan^{-1} \left( \frac{ny}{x} \right) \right\} + \frac{Ln(x \sin(\theta) - y \cos(\theta))}{x^2 + n^2 y^2} \right] \quad (4-15)$$

The robot in any poses converges to the origin with the  $u_1$  as the velocity input, and  $u_2$  as the steering angle input.



## **Chapter 5**

### **Singular Points and Feedback Gain Switching Strategy**

## 5.1 Singular Points

When  $u_1$  is equal to 0,  $\dot{x}$ ,  $\dot{y}$  and  $\dot{\theta}$  are respectively equal to 0 from Eqs. (4-1), (4-2) and (4-3). Therefore the point which satisfies (5-1) is regarded as the equilibrium point.

$$\lambda_1 x \cos \theta + \lambda_2 y \sin \theta = 0 \quad (5-1)$$

Recalling (4-14), when  $u_1$  approaches zero, inside of  $\tan^{-1}\{\}$  trends towards  $\pm\infty$ . These points other than the origin are also regarded as singular points. (5-1) can be rewritten as (5-2).

$$\theta = \tan^{-1}\left(-\frac{\lambda_1 x}{\lambda_2 y}\right) \quad (5-2)$$

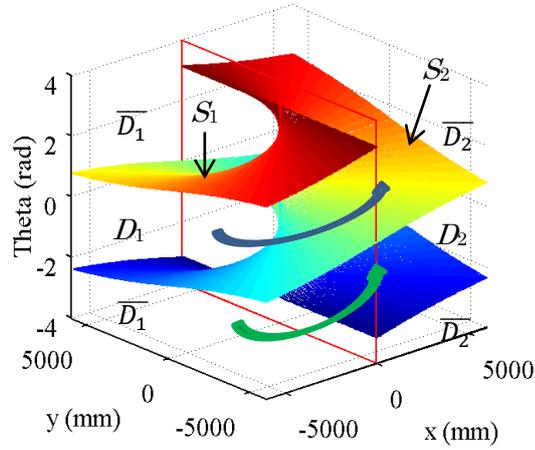


Fig. 5. 1: Singular surfaces when  $\lambda_1 = \lambda_2$

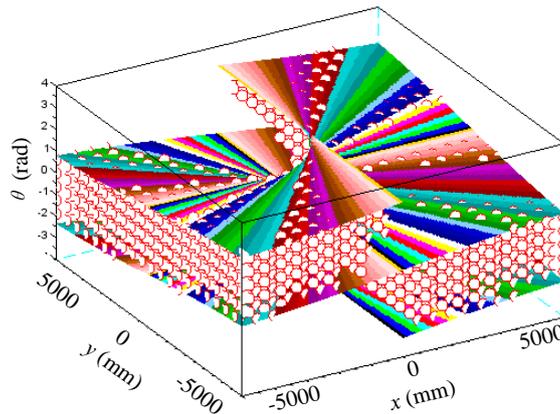


Fig. 5. 2: Initial poses which do not pass singular points

When  $\lambda_1 = \lambda_2$ , helical surfaces formed by the singular points satisfying (5-2) in the  $x$ - $y$ - $\theta$  state

space are presented in Fig. 5.1. Two surfaces can be formed by the straight line  $y = -x\cos\theta/\sin\theta$ , which rotates counterclockwise around the  $\theta$ -axis from  $-\pi$  to  $\pi$ . Upper and lower surfaces are designated respectively as  $S_1$  and  $S_2$ . The plane  $x = 0$  is expressed by the red rectangle. When  $x < 0$ , the space between the two surfaces is defined as  $D_1$ . The outside of  $D_1$  is defined as  $\overline{D}_1$ . When  $x > 0$ , the space between the two surfaces is defined as  $D_2$ . The outside of  $D_2$  is defined as  $\overline{D}_2$ . According to a previous report [9]. When the initial pose is in  $D_1$ , the robot moves forward and converges to the origin, when the initial pose is in  $D_2$ , the robot moves backward and converges to the origin. Shown in Fig. 5.2, the small red circles stand for the robot initial poses, which do not pass the singular points. When the initial pose is in  $\overline{D}_1$  or  $\overline{D}_2$ , the robot intersects with  $S_1$  or  $S_2$  and stops at the singular point. In order to make robot restart at the singular point, the feedback gain switching strategy has been proposed.

The simulation including three parts, the first is the initial poses in the outside of  $D_1$  and  $D_2$ . The second is applied for the feedback gain switching strategy without steering angle saturation. The third is the applied for the feedback gain switching strategy with steering angle saturation.

The simulation results are shown in Fig. 5.3 and Fig. 5.4. The trajectories when the initial poses in the outside of  $D_1$  and  $D_2$ .

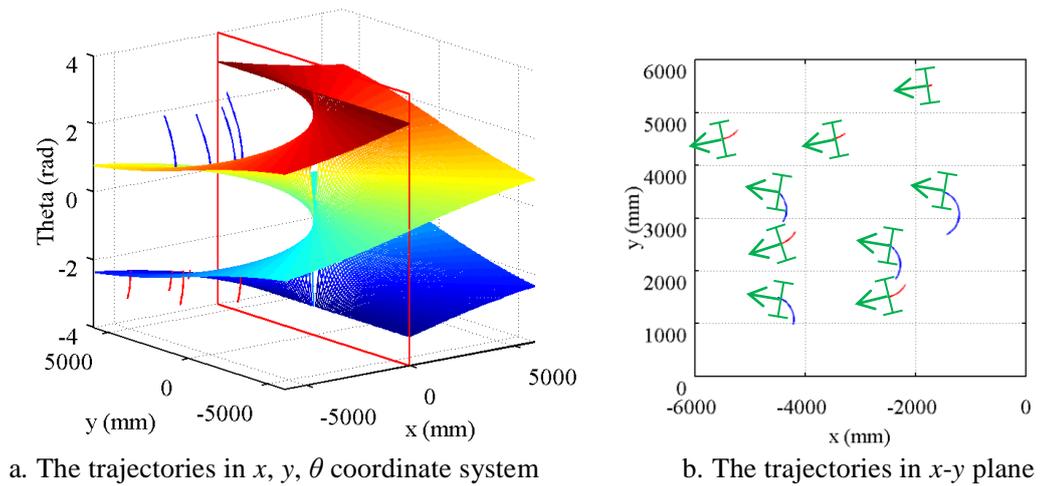


Fig. 5. 3: The trajectories of initial poses in the outside of  $D_1$

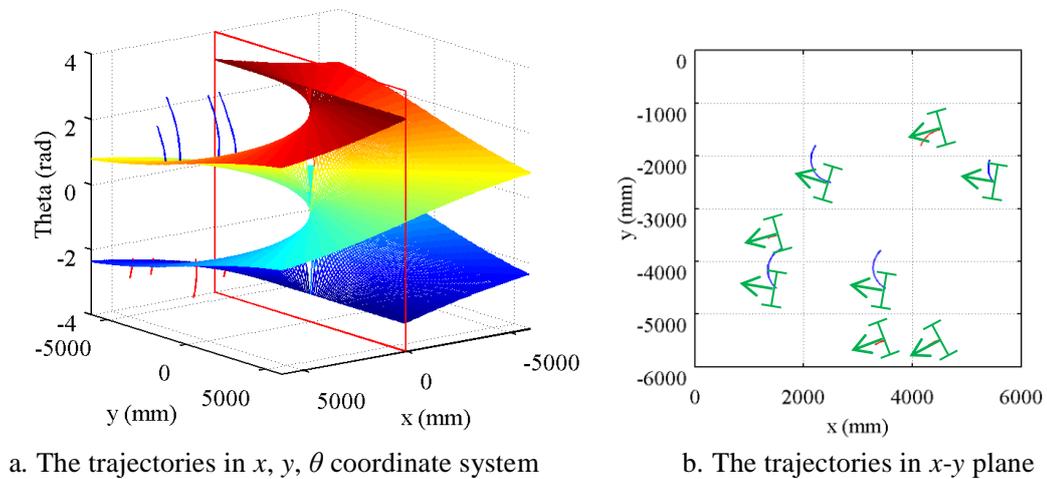


Fig. 5. 4: The trajectories of initial poses in the outside of  $D_2$

## 5.2 Feedback Gain Switching Strategy

If the robot starts from  $\overline{D}_1$  or  $\overline{D}_2$ , then it reaches  $S_1$  or  $S_2$  on the way to the origin. At that instant,  $u_1$  of the PGR-O becomes zero. In Fig. 5.5,  $P_0$  is defined as start point,  $P_1$  is defined as a singular point on  $S_1$ , and the target point is the origin. When the robot reaches  $S_1$ , it stops at  $P_1$ . To restart the robot,  $\lambda_1$  or  $\lambda_2$  can be changed to an appropriate value to satisfy  $u_1 \neq 0$ . The new velocity  $u_1'$  is obtainable by changing  $\lambda_2$  to  $\lambda_2' > 0$  from (5-3). Similarly, the new velocity  $u_1'$  is obtainable by changing  $\lambda_1$  to  $\lambda_1' > 0$ .

$$u_1' = -\lambda_1' x \cos \theta - \lambda_2' y \sin \theta \quad (5-3)$$

However in the practical application, when the robot stops at a singular point, a small new velocity  $u_1'$  is given. Its sign is opposite that of velocity  $u_1$  before stopping at the singular point. Then the new feedback gain  $\lambda_2'$  is obtained from (5-4).

$$\lambda_2' = \frac{-u_1' - \lambda_1' x \cos \theta}{y \sin \theta} \quad (5-4)$$

The singular surfaces  $S_1$  and  $S_2$  are changed to the new singular surfaces  $S_1'$  and  $S_2'$ , as depicted in Fig. 5.5 from (5-5).

$$\theta_{\lambda_2'} = \tan^{-1}\left(-\frac{\lambda_1' x}{\lambda_2' y}\right) \quad (5-5)$$

In the numerical example presented in Fig. 5.5, the robot starts from  $P_0 = (-4000 \text{ mm}, 3000 \text{ mm}, 5\pi/6 \text{ rad})$ . Then it intersects with  $S_1$  at  $P_1 = (-3834.5 \text{ mm}, 2399.9 \text{ mm}, 1.1 \text{ rad})$ . A new velocity  $u_1' = 30 \text{ mm/s}$  is given. Then  $\lambda_2 = 0.1$  is changed to  $\lambda_2' = 0.0701$  from (5-5). The robot restarts and converges to the origin.

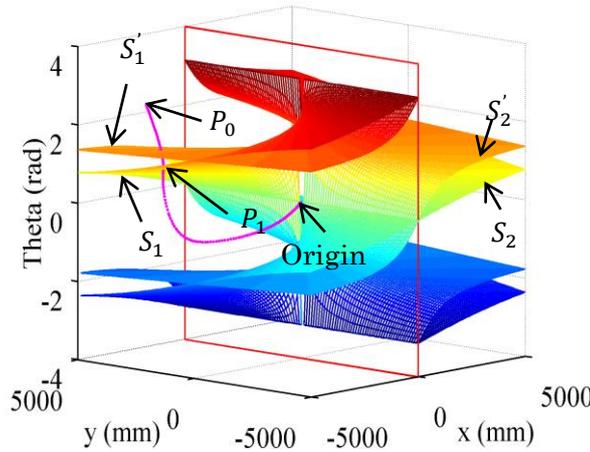


Fig. 5. 5: Robot restarts after  $\lambda_2$  is changed equivalent to the car-like robot to  $\lambda_2'$

The trajectories in Fig. 5.6 and Fig. 5.7 were adopted the feedback gain switching strategy.

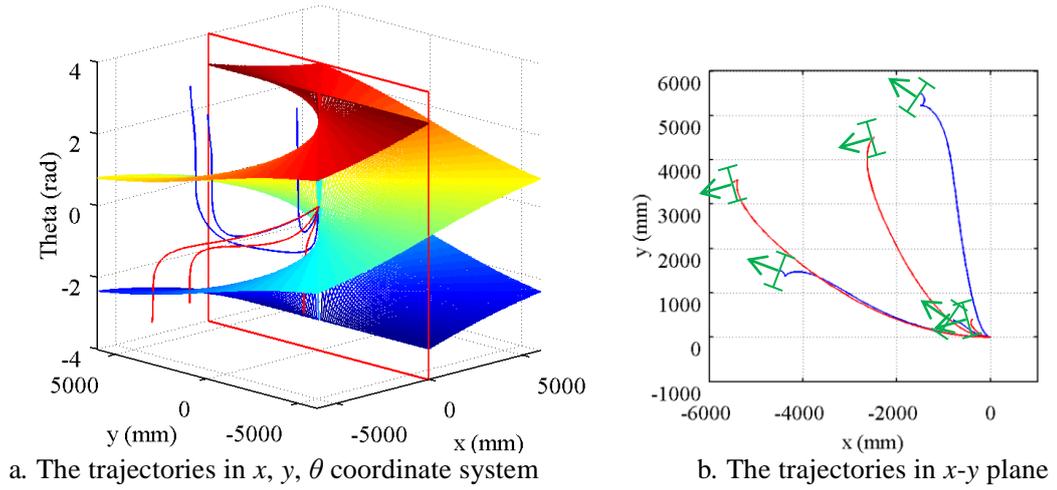


Fig. 5. 6: The trajectories was adopted the feedback gain switching strategy when the initial poses outside of  $D_1$

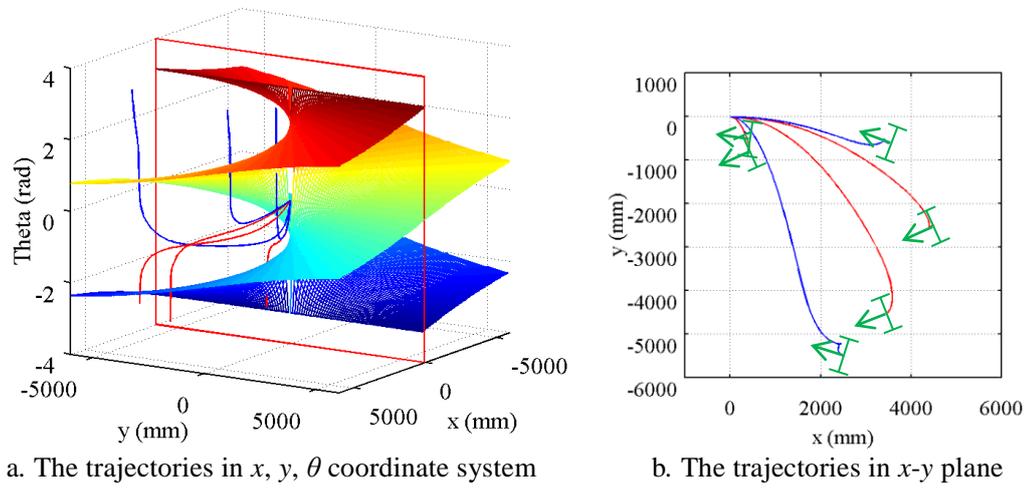


Fig. 5. 7: The trajectories was adopted the feedback gain switching strategy when the initial poses outside of  $D_2$

## **Chapter 6**

# **The Influence of Convergent Property on PGR-O with Steering Angle Saturation**

## 6.1 Running Area Division based on the Convergence

In this section, we present an investigation of the behavior of the robot under steering angle saturation. When the robot restarts with a small value of  $u_1$  after gain switches,  $u_2$  is close to  $\pm\pi/2$  rad, which might exceed  $u_{2\max}$ . This phenomenon causes steering angle saturation, where the robot drives along the minimum turning circle.

First, the minimum turning circle is defined. Next, the process of robot restart is discussed in detail under steering angle saturation. Then, we point out that the convergent property is related strongly to the number of the singular points. The running area is divided into four regions to clarify the convergent properties at different locations.

### 6.1.1 Minimum Turning Circle under Steering Angle Saturation

As portrayed in Fig. 6.1, assuming that the robot turns without slipping of the tires, the minimum turning circles are defined on both sides of it. The inner of the two circles is the area where the robot can not reach because of the steering angle saturation. The minimum turning radius  $R$  of the robot is expressed as (6-1).

$$R = \frac{L}{\tan u_{2\max}} \quad (6-1)$$

The center of the left minimum turning circle is expressed as (6-2) and (6-3)

$$x_l = x - R \sin \theta \quad (6-2)$$

$$y_l = y + R \cos \theta \quad (6-3)$$

The center of the right minimum turning circle is expressed as (6-4) and (6-5)

$$x_r = x + R \sin \theta \quad (6-4)$$

$$y_r = y - R \cos \theta \quad (6-5)$$

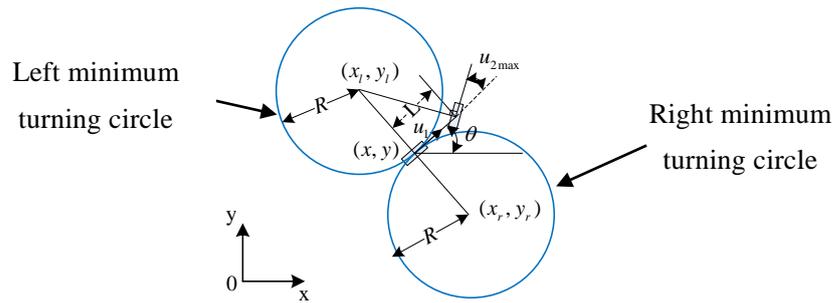


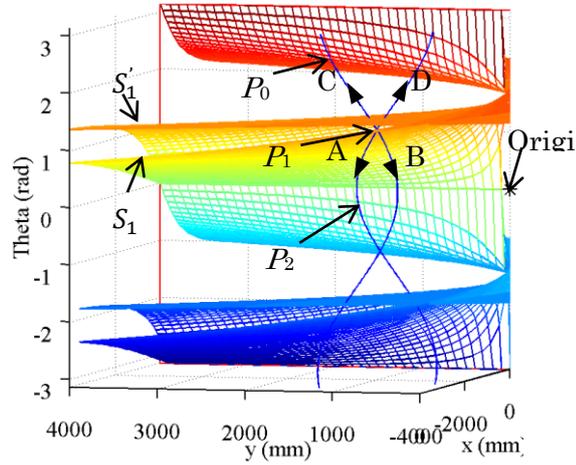
Fig. 6. 1: Minimum turning circles in  $x$ - $y$  plane

### 6.1.2 Robot Restart after Feedback Gain Switching Strategy

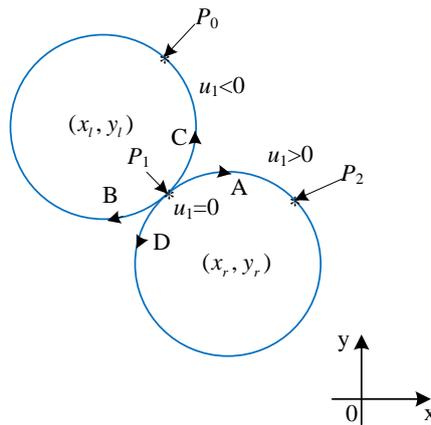
Figure 6.2a shows the trajectory which the robot drives along the minimum turning circles in the  $x$ - $y$ - $\theta$  state space, whereas Figure 6.2b shows the trajectory in the  $x$ - $y$  plane. The two blue lines represent the minimum turning circles. Here,  $P_0$  is the start point,  $P_1$  is the singular point or restart

point,  $P_2$  is defined as the point of leaving the minimum turning circle. Here, A, B, C, and D respectively represent the directions along minimum turning circles at the singular point. We assume that the center of the right minimum turning circle locates in the second quadrant and that  $P_0$  locates in  $\overline{D_1}$ . The robot drives along the left minimum turning circle with two singular points from the start pose. When the car-like robot stops at  $P_1$ , the new velocity  $u'_1$  is obtained using the feedback gain switching strategy. In this case,  $x < 0$ ,  $y > 0$ ,  $0 < \theta < \pi/2$ ,  $\lambda > 0$ ,  $n > 0$ ,  $L > 0$ ,  $u'_1 > 0$ , and  $u'_1$  is close to zero. Therefore,  $u_2$  tends to be  $-\pi/2$  rad from (4-15), under steering angle saturation,  $u_2 = -u_{2\max}$ . Because  $u_1 > 0$  and  $u_2 = -u_{2\max}$ , the robot restarts and drives to the A direction along the right minimum turning circles until it reaches  $P_2$ .

Similarly, when the center of minimum turning circle locates in the other quadrants, the robot can also restart. If the number of the singular points is two, the robot can leave the circle at the leaving point between them. However, if the number of the singular points is more than two, the robot might reach the second singular point before the leaving point. This is a main result of this paper.



a. x-y- $\theta$  state space



b. x-y plane

Fig. 6. 2: Trajectory which the robot drives along the minimum turning circles

### 6.1.3 Region Division based on the Center Location of Minimum Turning Circle

At the foundation of the analysis about the center location of the minimum turning circle, we divide the running area of the robot into four regions.

First, the number and location of the singular points are analyzed when the center of the right minimum turning circle is in the third quadrant. The robot is assumed to drive along the right minimum turning circle.  $u_1$  is written as (6-6).

$$u_1 = -\lambda_1(x_r - R \sin \theta) \cos \theta - \lambda_2(y_r + R \cos \theta) \sin \theta \quad (6-6)$$

The two sides of (6-7) are multiplied by  $\sec(\theta)\csc(\theta)$  to become  $F(\theta)$ .

$$F(\theta) = \lambda_1(x_r \csc \theta - R) + \lambda_2(y_r \sec \theta + R) \quad (\sin \theta \neq 0, \cos \theta \neq 0) \quad (6-7)$$

Defining  $F_1(\theta) = \lambda_1(x_r \csc \theta - R)$ ,  $F_2(\theta) = \lambda_2(y_r \sec \theta + R)$ ,  $F(\theta)$  is the sum of  $F_1(\theta)$  and  $F_2(\theta)$ . Figure 6.3 presents the outlines of  $F_1(\theta)$ ,  $F_2(\theta)$ , and  $F(\theta)$ . We define  $a$  as the value of  $F_1(\theta)$  when  $\theta$  is equal to  $-\pi/2$ ,  $b$  as the value of  $F_1(\theta)$  when  $\theta$  is equal to  $\pi/2$ ,  $c$  as the value of  $F_2(\theta)$  when  $\theta$  is equal to 0,  $d$  as the value of  $F_2(\theta)$  when  $\theta$  is equal to  $\pm\pi$ ,  $e$  as the minimum value of the  $F(\theta)$  when  $-\pi < \theta < -\pi/2$ , and  $f$  as the maximum value of  $F(\theta)$  when  $0 < \theta < \pi/2$ . If  $F(\theta) = 0$ ,  $u_1 = 0$ , then the number of the singular points can be expressed by the number of interaction points of the curve of  $F(\theta)$  with the horizontal axis.

When  $x_r < -R$ ,  $y_r < -R$ ,  $a = -x_r - R > 0$ ,  $b = x_r - R < -2R$ ,  $c = y_r + R < 0$ , and  $d = -y_r + R > 2R$ . Therefore,  $e > 0$ ,  $f < 0$ . The outline of  $F(\theta)$  is portrayed in Fig. 6.4. There is only one interaction point (singular point) separately when  $\theta$  is within the two intervals of  $-\pi < \theta < 0$  and  $0 < \theta < \pi$ .

When  $-R < x_r < 0$ ,  $-R < y_r < 0$ ,  $a = -x_r - R$ ,  $-R < a < 0$ ,  $b = x_r - R$ ,  $-2R < b < -R$ ,  $c = y_r + R$ ,  $0 < c < R$ ,  $d = -y_r + R$ ,  $R < d < 2R$ . Therefore,  $e$  might be less than 0 and  $f$  might be greater than 0. The outline of  $F(\theta)$  is presented in Fig. 6.5. One or three interaction points (singular points) might exist separately when  $\theta$  is within the two intervals of  $-\pi < \theta < 0$  and  $0 < \theta < \pi$ .

When  $x_r < -R$ ,  $-R < y_r < 0$ ,  $a = -x_r - R > 0$ ,  $b = x_r - R < -2R$ ,  $c = y_r + R$ ,  $0 < c < R$ ,  $d = -y_r + R$ , and  $R < d < 2R$ . Therefore,  $e > 0$  and  $f$  might be greater than 0. The outline of  $F(\theta)$  is shown in Fig. 6.6. There might be one or three interaction points (singular points) when  $\theta$  is within the interval of  $0 < \theta < \pi$  or only one interaction point (singular point) within the interval of  $-\pi < \theta < 0$ .

When  $-R < x_r < 0$ ,  $y_r < -R$ ,  $a = -x_r - R$ ,  $-R < a < 0$ ,  $b = x_r - R$ ,  $-2R < b < -R$ ,  $c = y_r + R < 0$ ,  $d = -y_r + R > 2R$ . Therefore  $e$  might be less than 0, and  $f < 0$ . The outline of  $F(\theta)$  is presented in Fig. 6.7. One or three interaction points (singular points) might exist when  $\theta$  is within the interval of  $-\pi < \theta < 0$  and only one interaction point (singular point) within the interval of  $0 < \theta < \pi$ .

Similarly, in the other quadrants, the number and location of the singular points have the same property in the same region. To clarify the convergent property at different locations, we divide the running area into four regions presented in Fig. 6.8 based on the analysis above. Among them,  $|x_r| > R$  and  $|y_r| > R$  are defined as region I.  $|x_r| < R$  and  $|y_r| > R$  are defined as region II.  $|x_r| > R$  and  $|y_r| < R$  are defined as region III.  $|x_r| < R$  and  $|y_r| < R$  are defined as region IV. The subscript denotes the quadrant in Fig. 6.8.

This section explains the simulation and experiment conducted to verify the reasonability of region division on the convergent property. In the resulting discussion, we also address an example in which

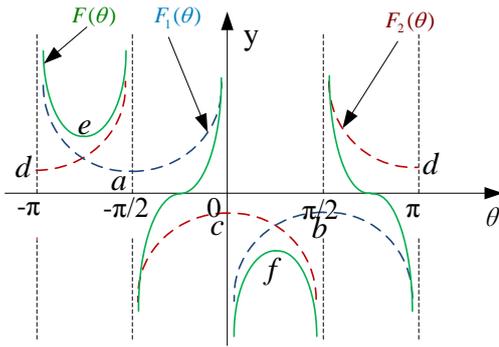


Fig. 6. 3: Outline of  $F_1$  and  $F_2$

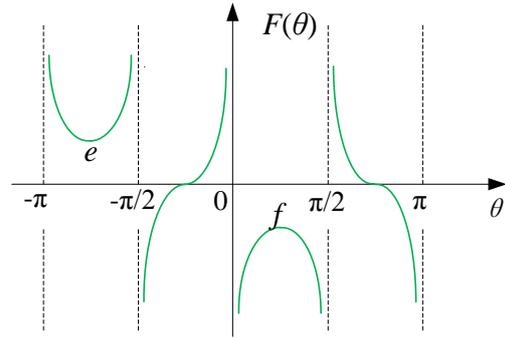


Fig. 6. 4:  $F(\theta)$  ( $x_r < -R, y_r < -R$ )

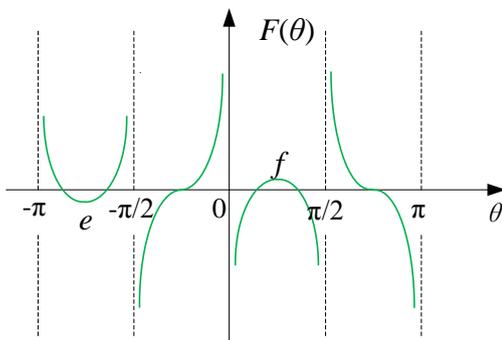


Fig. 6. 5:  $F(\theta)$  ( $-R < x_r < 0, -R < y_r < 0$ )

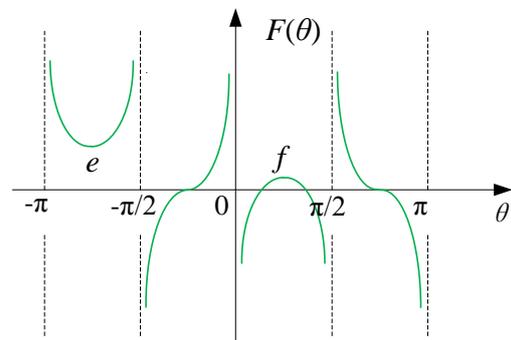


Fig. 6. 6:  $F(\theta)$  ( $x_r < -R, -R < y_r < 0$ )

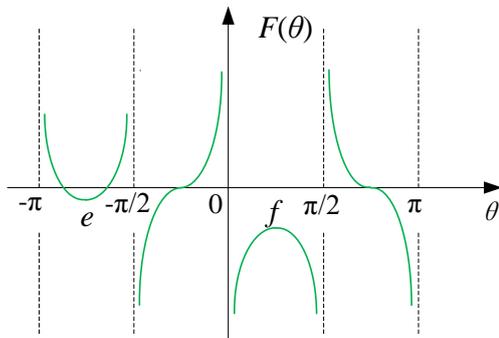


Fig. 6. 7:  $F(\theta)$  ( $-R < x_r < 0, y_r < -R$ )

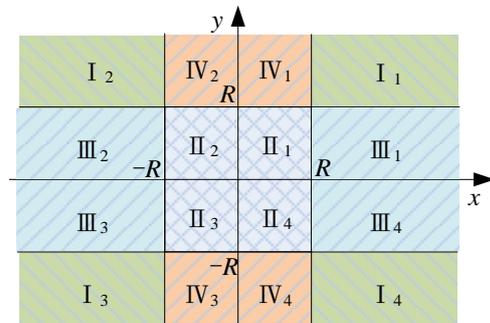


Fig. 6. 8: Region division based on the center

location of minimum turning circle

the number of the singular points and convergent property can be changed by adjusting  $\lambda_1$  and  $\lambda_2$  in region III.

## 6.2 Equipment

As presented in Fig. 6.9, a robot car (RoboCar(R) 1/10; ZMP Inc.) with length  $\times$  width  $\times$  height =

429.0 mm × 195.0 mm × 212.2 mm is used. Its kinematic parameters are  $L = 256$  mm, and  $u_{2\max} = \pi/6$  rad. Its minimum turning radius  $R = 443.4$  mm from (6-1). It is equipped with LRF (URG-04LX; Hokuyo Electric Inc.) to identify landmarks for autonomous navigation, a wireless LAN adapter to allow a remote access to the car from a distant computer. The MATLAB/Simulink is used for programming. Figure 6.10 presents the experiment environment. A pipe, used as a landmark, has diameter of 115 mm and height of 400 mm.

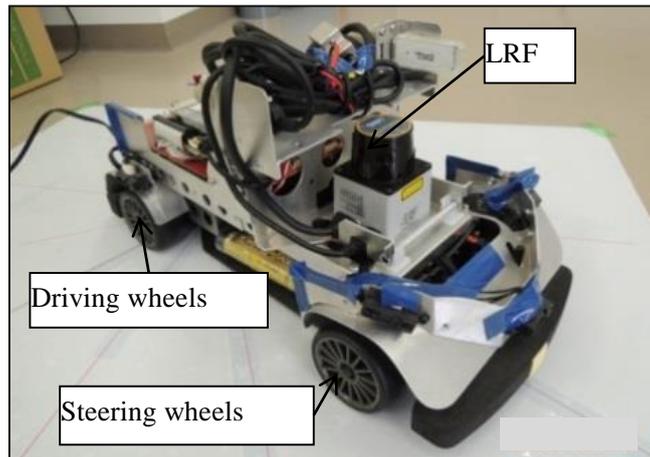


Fig. 6. 9: Robot car

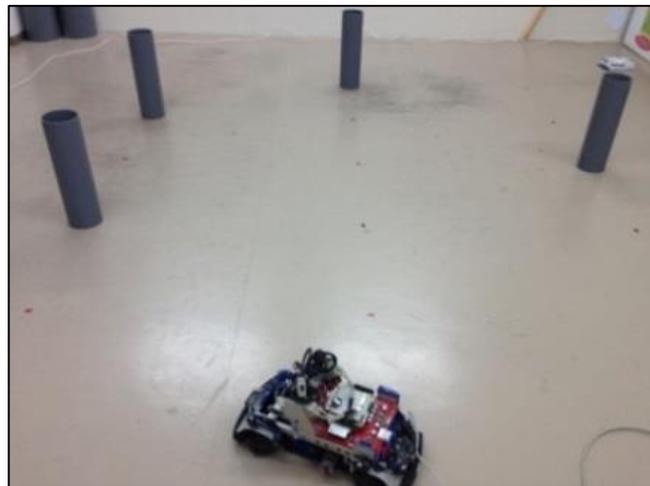


Fig. 6. 10: Experiment environment

## 6.3 Simulation and Experiment Method

### 6.3.1 The Theory of Extended Kalman Filter SLAM

The extended Kalman filter (EKF) is as the name suggests an extension of the basic Kalman filter. The basic Kalman filter is a recursive algorithm used for estimation of state variables in a linear system with noisy measurements. As most real systems are far from being linear, the basic Kalman filter is inadequate for estimation of states in these systems. The extended Kalman filter can be used instead as

it allows state transitions and state observations to be non-linear functions. The EKF linearizes the state transition and observations functions around the estimated state mean. The covariance estimate of the states is thus dependent of the mean estimate of the state in contrast to the basic Kalman filter.

The EKF algorithm has as earlier mentioned two phases, predict and update. The prediction phase projects new estimations to the state vector and the covariance matrix, based on the previous estimations.

This algorithm is applied for sensor model with known correspondence. To implement this measurement model, we need to define a variable that establishes correspondence between the feature  $f_t^i$  and the landmark  $m_j$  in the map. This variable will be denoted by  $c_t^i$  with  $c_t^i \in \{1, 2, \dots, N+1\}$ ;  $N$  is the number of landmarks in the map  $m$ . If  $c_t^i < j \leq N$ , then the  $i$ -th feature observed at time  $t$  corresponds to the  $j$ -th landmark in the map. In other words,  $c_t^i$  is the true identity of an observed feature. The only exception occurs with  $c_t^i = N+1$ : Here a feature observation does not correspond to any feature in the map  $m$ . This case is important for handling spurious landmarks; it is also of great relevance for the topic of robotic mapping, in which the robot regularly encounters previously unobserved landmarks.

In SLAM, the initial pose is taken to be to origin of the coordinate system. This definition is somewhat arbitrary, in that it can be replaced by any coordinate. None of the landmark locations are known initially. The following initial mean and covariance express this belief:

$$\mu_0 = (0 \ 0 \ 0 \ \dots \ 0)^T \quad (6-8)$$

$$\Sigma_0 = \begin{pmatrix} 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 \end{pmatrix} \quad (6-9)$$

The covariance matrix is of size  $(3N+3) \times (3N+3)$ . It is composed of a small  $3 \times 3$  matrix of zeros for the robot pose variables. All other covariance values are infinite.

As the robot moves, the state vector changes according to the standard noise-free (noise is zero) velocity model. In SLAM, this motion model is extended to the augmented state vector:

$$X_{t+1} = X_t + \begin{pmatrix} \Delta t \cdot v_{t+1} \cos \phi_{t+1} - \Delta t \cdot \frac{v_{t+1}}{L} \cdot \tan \alpha_{t+1} \cdot a \sin \phi_{t+1} \\ \Delta t \cdot v_{t+1} \sin \phi_{t+1} + \Delta t \cdot \frac{v_{t+1}}{L} \cdot \tan \alpha_{t+1} \cdot a \cos \phi_{t+1} \\ \Delta t \cdot \frac{v_{t+1}}{L} \cdot \tan \alpha_{t+1} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (6-10)$$

Because the motion only affects the robot's pose and all landmarks remain where they are, only the first three elements in the update are non-zero. This enables us to write the same equation more compactly:

$$X_{t+1} = X_t + F_x^T \begin{pmatrix} \Delta t \cdot v_{t+1} \cos \phi_{t+1} - \Delta t \cdot \frac{v_{t+1}}{L} \cdot \tan \alpha_{t+1} \cdot a \sin \phi_{t+1} \\ \Delta t \cdot v_{t+1} \sin \phi_{t+1} + \Delta t \cdot \frac{v_{t+1}}{L} \cdot \tan \alpha_{t+1} \cdot a \cos \phi_{t+1} \\ \Delta t \cdot \frac{v_{t+1}}{L} \cdot \tan \alpha_{t+1} \end{pmatrix} \quad (6-11)$$

Here  $F_x$  is a matrix that maps the 3-dimensional state vector into a vector of dimension  $3N + 3$ .

$$F_x = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \underbrace{0 \cdots 0}_{3N \text{ column}} \end{pmatrix} \quad (6-12)$$

The full motion model with noise is then as follows

$$X_{t+1} = X_t + F_x^T \underbrace{\begin{pmatrix} \Delta t \cdot v_{t+1} \cos \phi_{t+1} - \Delta t \cdot \frac{v_{t+1}}{L} \cdot \tan \alpha_{t+1} \cdot a \sin \phi_{t+1} \\ \Delta t \cdot v_{t+1} \sin \phi_{t+1} + \Delta t \cdot \frac{v_{t+1}}{L} \cdot \tan \alpha_{t+1} \cdot a \cos \phi_{t+1} \\ \Delta t \cdot \frac{v_{t+1}}{L} \cdot \tan \alpha_{t+1} \end{pmatrix}}_{g(u_{t+1}, X_t)} + N(0, F_x^T R_t F_x) \quad (6-13)$$

where  $F_x^T R_t F_x$  extends the covariance matrix to the dimension of the full state vector squared.

As usual in EKFs, the motion function  $g$  is approximated using a first degree Taylor expansion

$$g(u_{t+1}, X_t) \approx g(u_{t+1}, \mu_t) + G_t (X_t - \mu_t) \quad (6-14)$$

where the Jacobian  $G_t = g'(u_{t+1}, \mu_t)$  is the derivative of  $g$  at  $X_t$ , as in Equation

$$G_t = \begin{pmatrix} 1 & 0 & -\Delta t \cdot v_{t+1} \sin \phi_{t+1} - \Delta t \cdot \frac{v_{t+1}}{L} \cdot \tan \alpha_{t+1} \cdot a \cos \phi_{t+1} \\ 0 & 1 & \Delta t \cdot v_{t+1} \cos \phi_{t+1} - \Delta t \cdot \frac{v_{t+1}}{L} \cdot \tan \alpha_{t+1} \cdot a \sin \phi_{t+1} \\ 0 & 0 & 1 \end{pmatrix} \quad (6-15)$$

Obviously, the additive form in (6-13) enables us to decompose this Jacobian into an identity matrix of dimension  $(3N + 3) \times (3N + 3)$  (the derivative of  $X_t$ ) plus a low-dimensional Jacobian  $g_t$  that characterizes the change of the robot pose:

$$G_t^i = I + F_x^T g_t F_x \quad (6-16)$$

With

$$g_t = \begin{pmatrix} 0 & 0 & -\Delta t \cdot v_{t+1} \sin \phi_{t+1} - \Delta t \cdot \frac{v_{t+1}}{L} \cdot \tan \alpha_{t+1} \cdot a \cos \phi_{t+1} \\ 0 & 0 & \Delta t \cdot v_{t+1} \cos \phi_{t+1} - \Delta t \cdot \frac{v_{t+1}}{L} \cdot \tan \alpha_{t+1} \cdot a \sin \phi_{t+1} \\ 0 & 0 & 0 \end{pmatrix} \quad (6-17)$$

Obviously, several of the matrices multiplied are sparse, which should be exploited when implementing this algorithm. The result of this update are the mean  $\bar{\mu}_t$  and the covariance  $\bar{\Sigma}_t$  of the estimate at time  $t$  after updating the filter with the control  $u_t$ , but before integrating the measurement  $z_t$ .

The derivation of the measurement update is similar to the parts in the localization algorithm. In particular, we are given the following measurement model

$$\begin{pmatrix} r_t^i \\ \theta_t^i \\ s_t^i \end{pmatrix}_{z_t^i} = \underbrace{\begin{pmatrix} \sqrt{(m_{j,x} - x_t)^2 + (m_{j,y} - y_t)^2} \\ \frac{\pi}{2} - \phi_t + \text{atan2}(m_{j,y} - y_t, m_{j,x} - x_t) \\ m_{j,s} \end{pmatrix}}_{h(x_t, j, m)} + N(0, \underbrace{\begin{pmatrix} \sigma_r & 0 & 0 \\ 0 & \sigma_\theta & 0 \\ 0 & 0 & \sigma_s \end{pmatrix}}_{\mathcal{Q}_t}) \quad (6-18)$$

$i$  is the index of an individual landmark observation in  $z_t$ , and  $j = c_t^i$  is the index of the observed landmark at time  $t$ . the variable  $r$  denotes the range to a landmark,  $\theta$  is the bearing to a landmark, and  $s$  is the landmark signature; the terms  $\sigma_r$ ,  $\sigma_\theta$  and  $\sigma_s$  are the corresponding measurement noise covariance.

This expression is approximated by the linear function

$$h(X_t, j) \approx h(\bar{\mu}, j) + H_t^i (X_t - \bar{\mu}_t) \quad (6-19)$$

Here  $H_t^i$  is the derivative of  $h$  with respect to the full state vector  $X_t$ . Since  $h$  depends only on two elements of that state vector, the robot pose  $X_t$  and the location of the  $j$ -th landmark  $m_j$ , the derivative factors into a low-dimensional Jacobian  $h_t^i$  and a matrix  $F_{x,j}$ , which maps  $h_t^i$  into a matrix of the dimension of the full state vector:

$$H_t^i = h_t^i F_{x,j} \quad (6-20)$$

Here  $h_t^i$  is the Jacobian of the function  $h(X_t, j)$  at  $\bar{\mu}_t$ , calculated with respect to the state variables  $x_t$  and  $m_j$ :

$$h_t^i = \begin{pmatrix} -\frac{m_{j,x} - \bar{\mu}_{t,x}}{\sqrt{q_t}} & -\frac{m_{j,y} - \bar{\mu}_{t,y}}{\sqrt{q_t}} & 0 & \frac{m_{j,x} - \bar{\mu}_{t,x}}{\sqrt{q_t}} & \frac{m_{j,y} - \bar{\mu}_{t,y}}{\sqrt{q_t}} & 0 \\ -\frac{m_{j,y} - \bar{\mu}_{t,y}}{q_t} & -\frac{m_{j,x} - \bar{\mu}_{t,x}}{q_t} & -1 & \frac{m_{j,y} - \bar{\mu}_{t,y}}{q_t} & \frac{m_{j,x} - \bar{\mu}_{t,x}}{q_t} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (6-21)$$

The scalar  $q_t = (m_{j,x} - \mu_{t,x})^2 + (m_{j,y} - \mu_{t,y})^2$ , and as before,  $j = c_t^i$  is the landmark that corresponds to the measurement  $z_t^i$ . The matrix  $F_{x,j}$  is of dimension  $6 \times (3N+3)$ . It maps the low-dimensional matrix  $h_t^i$  into a matrix of dimension  $3 \times (3N+3)$ .

$$F_{x,j} = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \end{pmatrix} \quad (6-22)$$

$\underbrace{\hspace{10em}}_{3j-3}$ 
 $\underbrace{\hspace{10em}}_{3N-3j}$

These expressions make up for the gist of the Kalman gain calculation in Lines 8 through 17 in our EKF SLAM algorithm, with one important extension. When a landmark is observed for the first time, its initial pose estimate in Equation (2-1) leads to a poor linearization. This is because with the

default initialization in (2-1), the point about which  $h$  is being linearized is  $(\hat{\mu}_{j,x} \ \hat{\mu}_{j,y} \ \hat{\mu}_{j,s})^T = (0 \ 0 \ 0)^T$ , which is a poor estimator of the actual landmark location. A better landmark estimator is given in Line 10. Here we initialize the landmark estimate  $(\hat{\mu}_{j,x} \ \hat{\mu}_{j,y} \ \hat{\mu}_{j,s})^T$  with the expected position. This expected position is derived from the expected robot pose and the measurement variables for this landmark

$$\begin{pmatrix} \bar{\mu}_{j,x} \\ \bar{\mu}_{j,y} \\ \bar{\mu}_{j,s} \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{t,x} \\ \bar{\mu}_{t,y} \\ s_t^i \end{pmatrix} + \begin{pmatrix} r_t^i \cos(\theta_t^i + \bar{\mu}_{t,\phi}) \\ r_t^i \sin(\theta_t^i + \bar{\mu}_{t,\phi}) \\ 0 \end{pmatrix} \quad (6-22)$$

We note that this initialization is only possible because the measurement function  $h$  is bijective. Measurements are two-dimensional, as are landmark locations. In cases where a measurement is of lower dimensionality than the coordinates of a landmark,  $h$  is a true projection and it is impossible to calculate a meaningful expectation for  $(\bar{\mu}_{j,x} \ \bar{\mu}_{j,y} \ \bar{\mu}_{j,s})^T$  from a single measurement only.

Finally, we note that the EKF algorithm requires memory that is quadratic in  $N$ , the number of landmarks in the map. Its update time is also quadratic in  $N$ . The quadratic update complexity stems from the matrix multiplications that take place at various locations in the EKF.

### 6.3.2 The Control of Car-like Robot with PGR-O and EKF-SLAM

The flow chart showing the robot car control system is presented in Fig. 6.11. The inputs of  $u_1$  and  $u_2$  are derived from the present pose based on the PGR-O. Using the EKF-SLAM algorithm [17, 19], the next pose is estimated according to the pose estimate model. When the robot receives scanning data from LRF, the landmarks can be identified based on circle feature extracting algorithm [35]. Then the data association is implemented with prediction and features. If the features are associated, then the robot adopts the current map and state model. If the features are not associated, then the robot ignores the fake features, adds new features, and updates the map.

## 6.4 Simulation and Experiment Conditions

The conditions of the simulation and experiment are that the center location of the right minimum turning circle is in the third quadrant. The default parameter values for numerical simulation and experiment are shown in Table 1. The robot initial poses in numerical simulation and experiment are shown in Table 2. Condition 1 is that the center of right minimum turning circle is in region  $I_3$ . Condition 2 is that the center of right minimum turning circle is in region  $II_3$ . Condition 3 is that the center of the right minimum turning circle is in region  $III_3$ . Condition 4 is that the center of right minimum turning circle is in region  $IV_3$ .



mm, -327.7 mm, 2.21 rad),  $u_1' = 20$  mm/s is given, then  $\lambda_2' = 0.2416$  from (5-4), the robot restarts at the first singular point, but it stops at the second singular point (-708.3 mm, 86.5 mm, 1.24 rad). Figure 6.14 depicts the results obtained for condition 3, when the robot stops at the first singular point (-1601.2 mm, -88.5 mm, 1.69 rad),  $u_1' = 20$  mm/s is given, then  $\lambda_2' = 0.5754$  from (5-4), the robot restarts at the first singular point, but it stops at the second singular point (-1589.3 mm, 38.4 mm, 1.39 rad). Figure 6.15 depicts the results obtained for condition 4, when the robot stops at the first singular point (-416.1 mm, -1677.2 mm, -0.32 rad),  $u_1' = 20$  mm/s is given, then  $\lambda_2' = 0.0334$  from (5-4). The robot restarts at the first singular point. Then the robot passes through the y-axis and stops at the second singular point (147.4 mm, -1504.2 mm, 0.26 rad) in  $S_2$ .

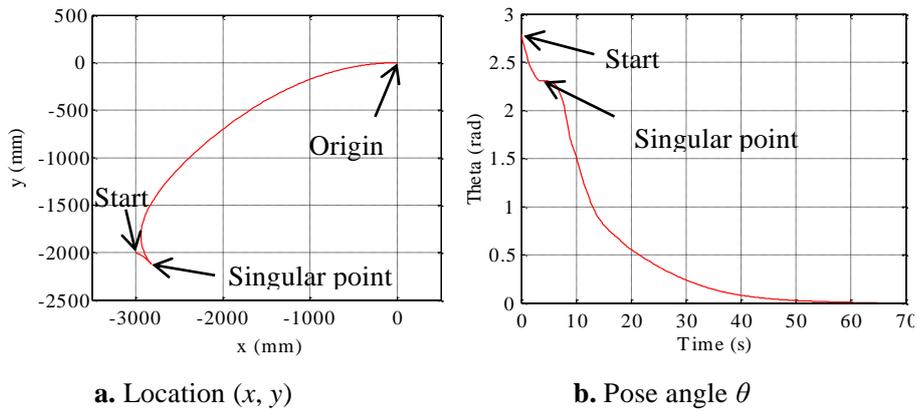


Fig. 6. 12: Simulation results for condition 1 ( $\lambda_2' = 0.1162$ )

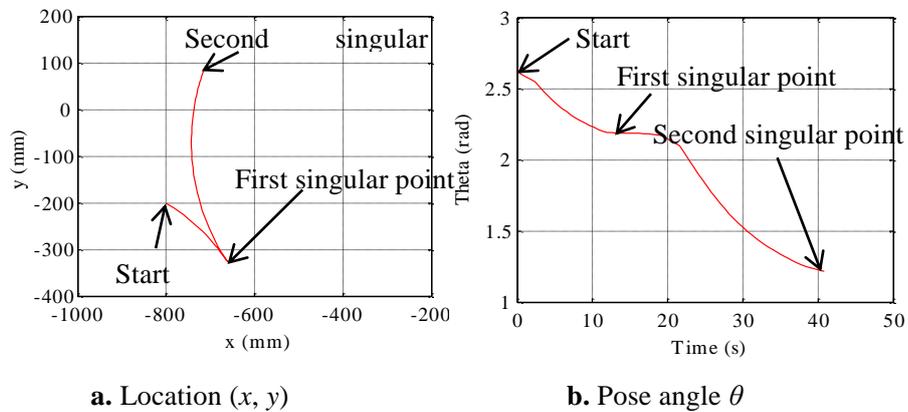


Fig. 6. 13: Simulation results for condition 2 ( $\lambda_2' = 0.2416$ )

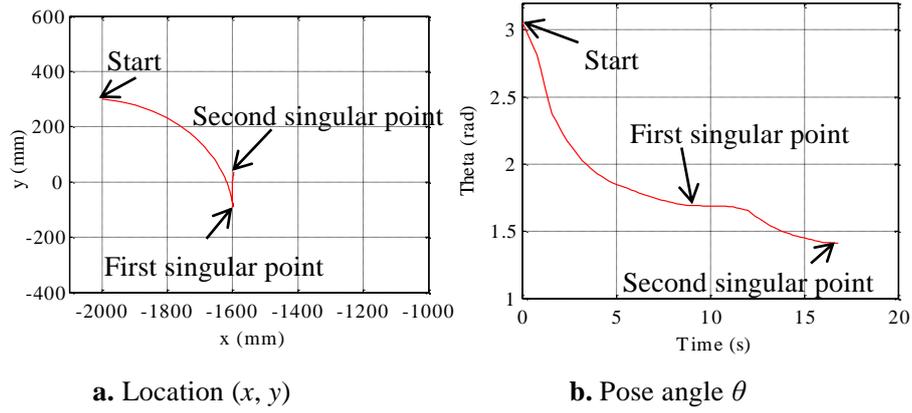


Fig. 6. 14: Simulation results for condition 3 ( $\lambda'_2 = 0.5754$ )

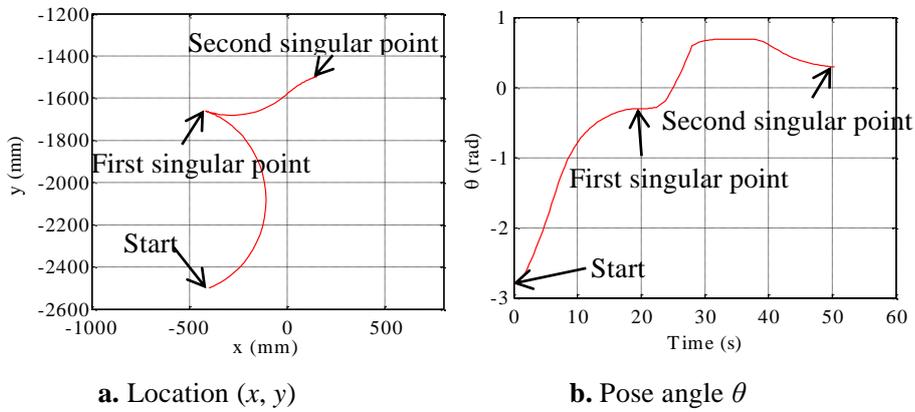


Fig. 6. 15: Simulation results for condition 4 ( $\lambda'_2 = 0.0334$ )

## 6.6 Experiment Results

The experiment is conducted with the robot car using the same configuration parameters. The location, pose angle, and the command value and measured value of  $u_1$ , and the command value and measuring value of  $u_2$  are shown in the experiment results. The command values of  $u_1$  and  $u_2$  are shown as the red line. The measuring values of  $u_1$  and  $u_2$  are shown as the blue line. For condition 1, the results are presented in Fig. 6.16. When the robot stops at the singular point of (-2767.4 mm, -2169.2 mm, 2.25 rad),  $u'_1 = 20$  mm/s is given. Then  $\lambda'_2 = 0.1231$ , it restarts and reaches the origin. For condition 2, the results are presented in Fig. 6.17. When the robot stops at the first singular point (-667.4 mm, -320.3 mm, 2.2 rad),  $u'_1 = 20$  mm/s is given. Then  $\lambda'_2 = 0.2577$ , it restarts at the first singular point, but it stops at the second singular point (-746.2 mm, 38.5 mm, 1.41 rad). For condition 3, the results are presented in Fig. 6.18. When the robot stops at the first singular point (-1575.3 mm, -81.2 mm, 1.71 rad),  $u'_1 = 20$  mm/s is given. Then  $\lambda'_2 = 0.5882$ , the robot restarts at the first singular point, but it stops at the second singular point (-1582.1 mm, 43.6 mm, 1.39 rad). Figure 6.19 depicts the results obtained for condition 4. When the robot stops at the first singular point (-392.3 mm, -1667.1 mm, -0.27 rad),  $u'_1 = 20$  mm/s is given. Then  $\lambda'_2 = 0.0311$ , the robot restarts at the first singular point, and it passes through the y-axis, then stops at the second singular point (120.1 mm, -1567.2 mm, 0.29 rad) in  $S_2$ . Moreover, the robot has a pause interval at the first singular point restart because the value of new

velocity  $u_1'$  is smaller. It has no effect on the purposes of this experiment.

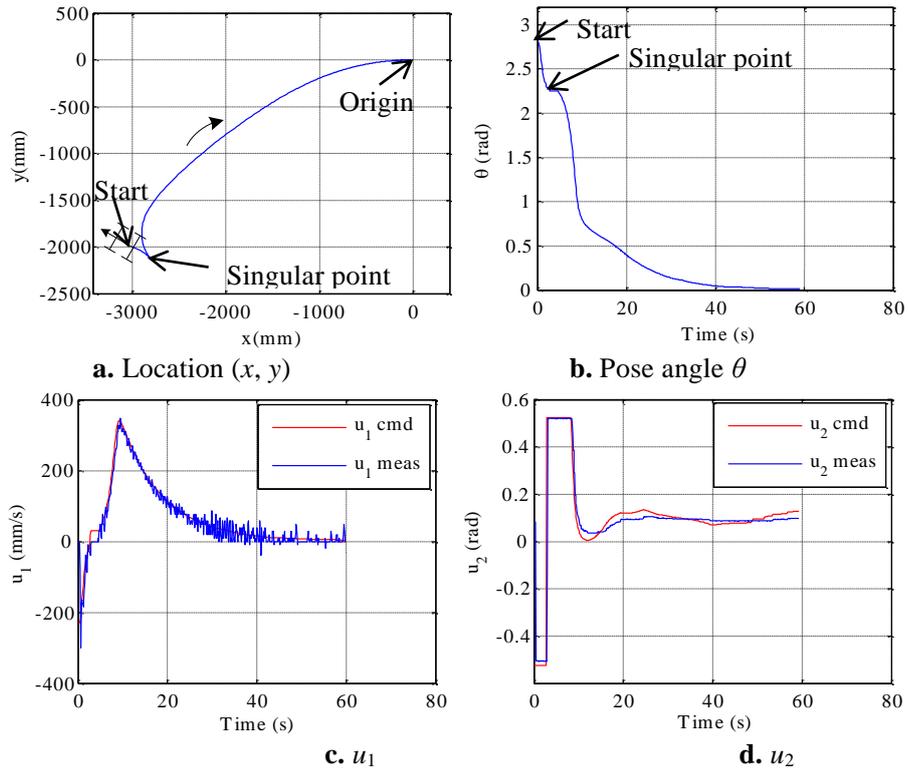


Fig. 6. 16: Experiment results for condition 1

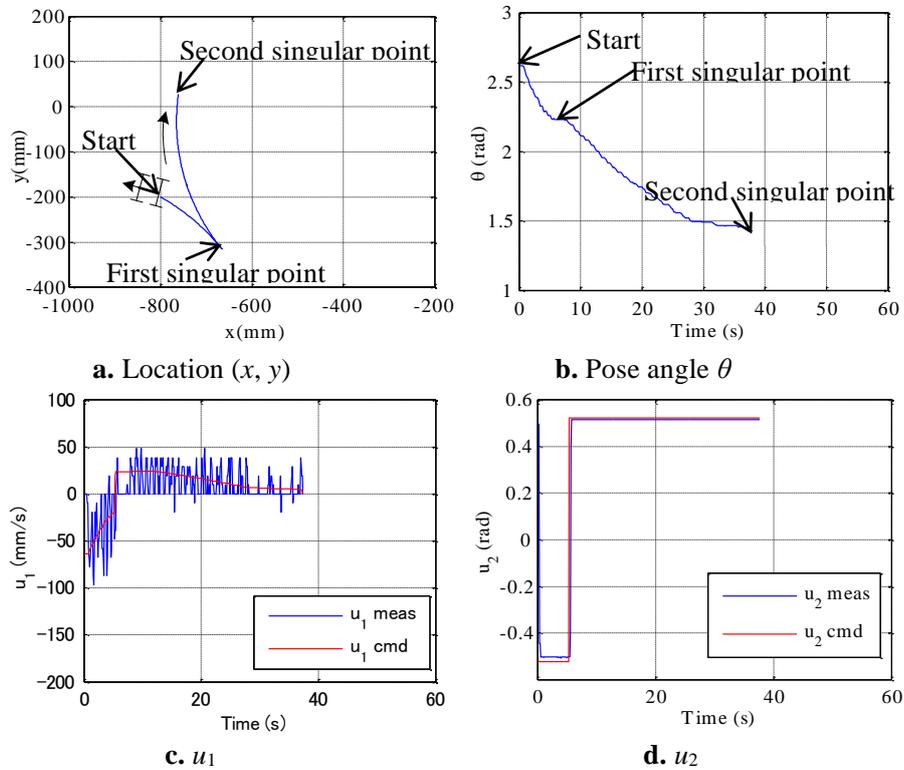


Fig. 6. 17: Experiment results for condition 2

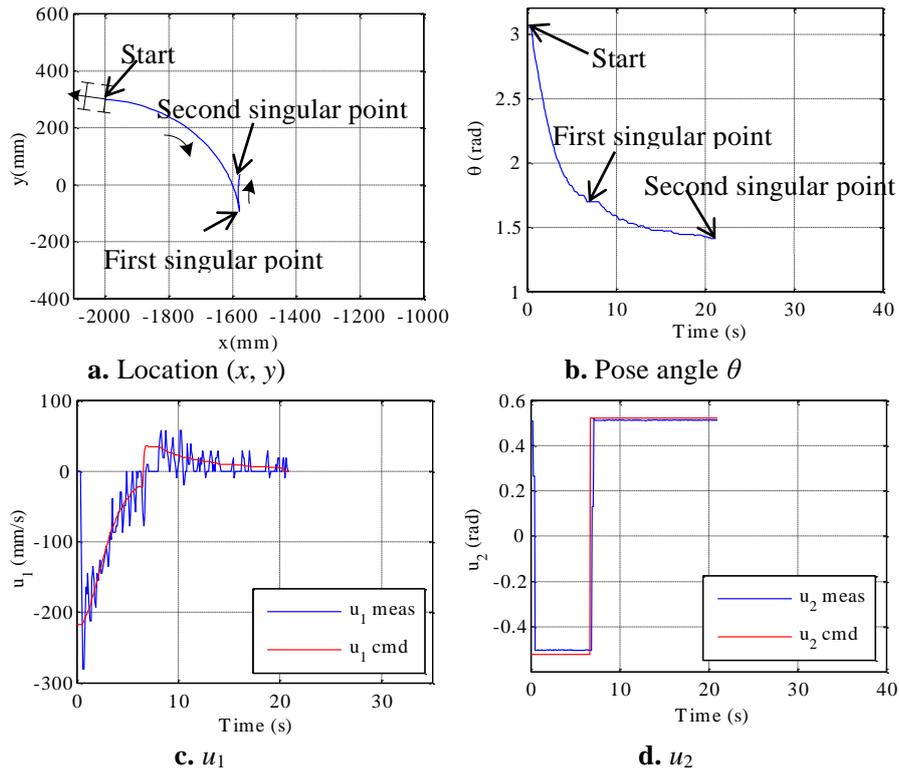


Fig. 6. 18: Experiment results for condition 3

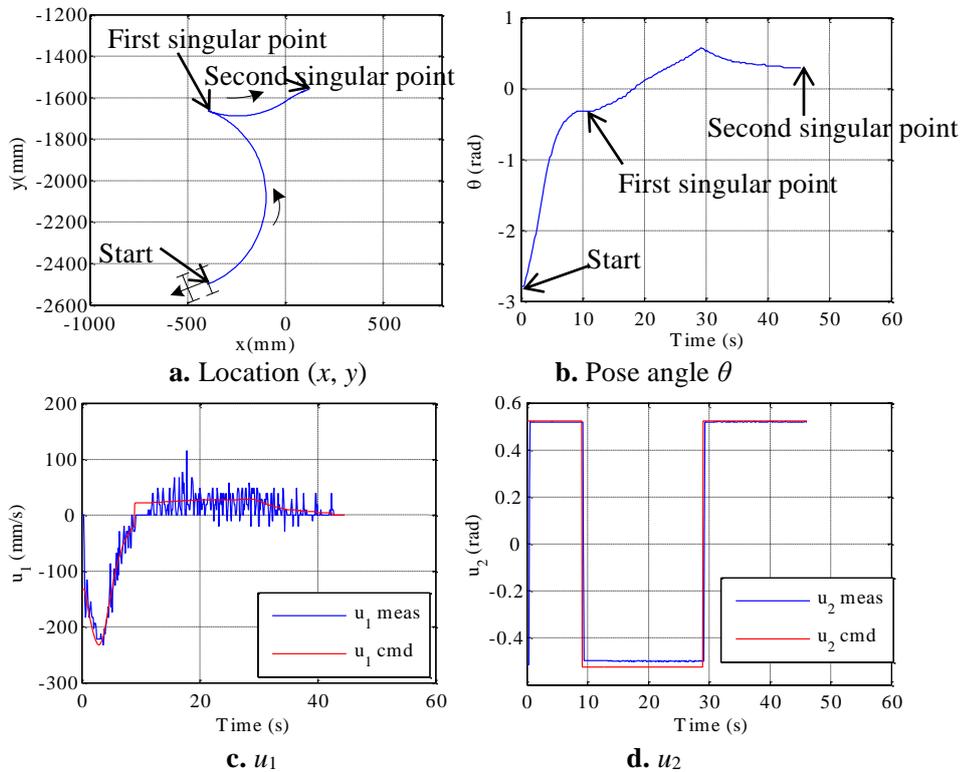


Fig. 6. 19: Experiment results for condition 4

## 6.7 Discussion

The results of simulations and experiments demonstrate the convergent properties of the four regions in the third quadrant under steering angle saturation.

In region  $I_3$ , singular points exist separately within the two intervals of  $-\pi < \theta < 0$  and  $0 < \theta < \pi$ . Consequently, the robot can converge to the origin.

In region  $II_3$ , three singular points might exist separately within the two intervals of  $-\pi < \theta < 0$  and  $0 < \theta < \pi$ . The robot restarts at the first singular point, but it might stop at the second singular point. Moreover, as presented in Fig. 6.20, the robot can not converge to the origin when the distance between the center of the minimum turning circle and the origin is less than  $R$ .

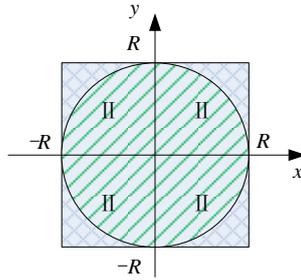


Fig. 6. 20: Running area can not converge to origin in region II

Region  $III_3$  has only a singular point within the interval of  $-\pi < \theta < 0$ . Three singular points might exist within the interval of  $0 < \theta < \pi$ . The robot restarts at the first singular points, but it might stop at the second singular point.

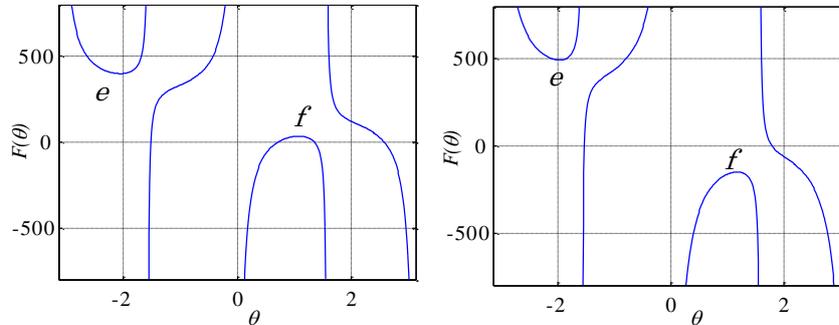


Fig. 6. 21:  $F(\theta)$  when  $\lambda_1 = 0.1$ ,  $\lambda_2' = 0.5346$  Fig. 6. 22:  $F(\theta)$  when  $\lambda_1' = 0.2$ ,  $\lambda_2' = 0.5346$

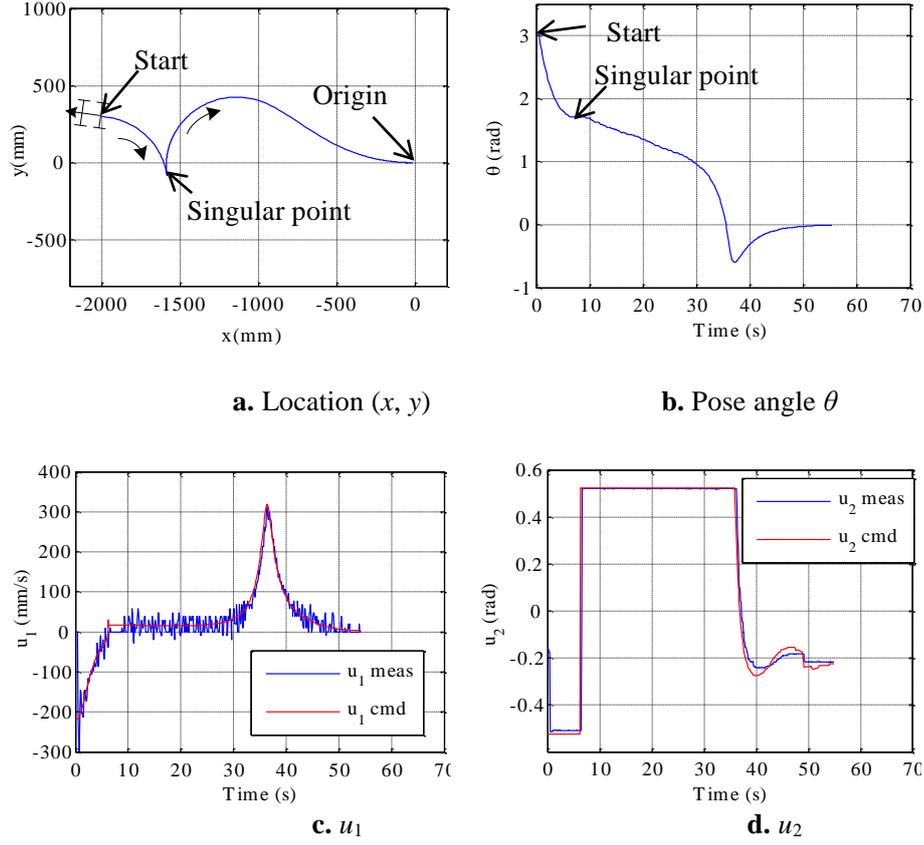


Fig. 6. 23: Experiment results for condition 3 with the extended feedback gain switching strategy

In condition 3, based on (6-6), it is able to obtain Fig. 6.21 after applying the feedback gain switching strategy. If  $f < 0$ , the two singular points within the interval of  $0 < \theta < \pi$  will vanish. Because the other parameters are immutable, except for  $\lambda_1$ , it must give larger  $\lambda'_1$  than  $\lambda_1$  such that  $f < 0$ . In this example, when  $\lambda_2$  is changed to  $\lambda'_2$ , the new  $\lambda'_1 = 0.2$  is given. The  $F(\theta)$  is presented in Fig. 6.22. Experiment results presented in Fig. 6.23 show that the robot can restart at the singular point and converge to the origin.

In region  $IV_3$ , there might be three singular points within the interval of  $-\pi < \theta < 0$  and only a singular point within the interval of  $0 < \theta < \pi$ . The robot restarts at the first singular point, but it might stop at the second singular point in  $S_1$  or might pass through the  $y$ -axis and then stop at the singular point in  $S_2$ .

Similar convergent properties are apparent when the center of minimum turning circles is in the other quadrants. The analysis clarifies the following situations: under steering angle saturation, when the center of the minimum turning circle is in region I, the car-like robot restarts at the singular point and converges to the origin; when the center location of minimum turning circle is in regions II, III and IV, the car-like robot might stop at the second singular point. Based on the number of singular points described in the summary presented above, the region division on the convergent property is reasonable. Moreover, if the value of  $\lambda_1$  or  $\lambda'_2$  at the singular point in region III is changed, then the robot might be able to converge to the origin.



## **Chapter 7**

### **Conclusion**

In this research, the two-wheeled and four-wheeled mobile robot are objects of study. The PGR was extended to track the arc passage and converge to the origin with steering angle saturation.

This research presented the PGR-A and the improved PGR-A with the adjustable look-ahead method to track the arc passage for two-wheeled robots first. The robot's trajectory tended to deviate outward from the target circle in the first method. To improve this weakness, the adjustable coefficient angle  $\beta$  is introduced. The robot adjusted the local coordinate system constantly according to the deviate between the trajectory and the target arc, which made the trajectory converge to the target arc perfectly. The simulation results also verified the improved PGR-A with adjustable look-ahead method had better performance on path tracking along the arc passage. This approach was also validated on the S-shaped passage, which indicated that it could be applied to multi-arc passage tracing problem.

We focused on the control method to track the arc passage and verify the efficiency on simulation environment without considering obstacles. However it does not mean that we neglect the obstacle avoidance problem. Actually, some members in our research group have proposed and are testing the obstacle avoidance algorithms with the PGR-A. In the future work, we will perform the experiments to verify the effectiveness of the method proposed in this paper in the real environment in the future.

Another issue of this study analyzes the influence of steering angle saturation to the convergent property in the PGR-O under the feedback gain switching strategy for car-like robots. The convergent property of the robot under steering angle saturation is investigated firstly. We report that the convergent property is related closely to the number of the singular points that are dependent on the center location of the minimum turning circle. Secondly, the convergent properties at different locations are clarified through region division. A method that extends the feedback gain switching strategy is proposed to change the convergent property in the specific region. Based on simulation and experiment results, we summarize convergent property related to the region and verify the validity of the extended method.

## **Acknowledgments**

I would first like to thank my advisor Prof. Naohiko HANAJIMA for his guidance and support over the course of my doctor's Program. His endless enthusiasm and patience is something to be admired and sought after in both academic settings as well as life. I also would like to thank Prof. Toshiharu KAZAMA, Prof. Koji TERAMOTO give me many support. I would also like to thank my wife for her support and encouragement.



## References

- (1) Brockett, R. W., Millman, R. S., Sussmann, H. J., Boston, Eds., and Birkhauser, MA., Asymptotic stability and feedback stabilization, *Differential Geometric Control Theory*, (1983), pp.181–191.
- (2) Kolmanovsky, I., and McClamroch, N. H., “Developments in nonholonomic control problems,” *IEEE Control. System. Mag.*, vol. 15 (1995), pp.20–36.
- (3) Jiang Z. P., and Pomet, J. B., Combining back stepping and time-varying techniques for a new set of adaptive controllers, *Proc. IEEE Int. Conf. Decision Control* (1994), pp.2207–2212.
- (4) Jiang, Z. P., Iterative design of time-varying stabilizers for multi-input systems in chained form, *System. Control. Letter.*, vol. 28 (1996), pp.255–262.
- (5) Hamel T. and Meizel, D., Robust control laws for wheeled mobile robots, *Int. J. Syst. Sci.*, vol. 27, no. 8 (1996), pp.695–704.
- (6) Aguilar, L. E., Soueres, P., Courdresses, M., and Fleury, S., Robust path following control with exponential stability for mobile robots, in *Proc. IEEE Int. Conf. Robot. Automat* (1998), pp. 3279–3284.
- (7) Michishita, T., Hanajima, N., Hikita, H., Miyashita, Y. and Yoshida, H., Development of a wire winding steering system for an autonomous all terrain vehicle, *Transactions of Japan Society of Mechanical Engineers, Series C, Vol.78, No.791* (2012), pp.2462–2473 (in Japanese).
- (8) Bowen, R. M., Jollineau, J., and Margheim, L., “iRobot corporation’s intellectual property: accounting for research and development under U.S. GAAP Versus IFRS,” *Journal of Business Case Studies*, vol. 9, no. 4 (2013), pp. 321-331.
- (9) Double Robotics, “Double,” *Mountain View CA*, <http://www.doublerobotics.com/>, 2014.
- (10) Petelin, J. B., Nelson, M. E., and Goodman, J., “Deployment and early experience with remote-presence patient care in a community hospital,” *Surg Endosc* (2006), pp. 53-56.
- (11) Ha, X. V., Ha, C., and Choi, H. S., “Path tracking control of a mobile robot by using dual estimation algorithm,” *Advances in Mechanical Engineering*, vol. 2013, Article ID 367127 (2013), 11 pages.
- (12) Saudi, A., Sulaiman, J., “Robot path planning using Four Point-Explicit Group Via Nine-Point Laplacian (4EG9L) iterative method,” *Procedia Engineering*, vol. 41 (2012), pp. 182-188.
- (13) Sampei, M., “Feedback control of nonholonomic systems,” *Journal of the Society of Instrument and Control Engineers*, vol. 39, no. 6 (1997), pp. 396-403.
- (14) Takashima, A., Hashimoto, Y., Hori, K., Hanajima, N., Hikita, H., and Yamashita, M., “Feedback control of non- holonomic mobile robots by Path-generating regulator,” *Trans. of JSME (C)*, vol. 70, no. 689 (2004), pp. 120-126, (in Japanese).
- (15) Taruumi, Y., Hanajima, N., Dai, J., and Takashima, A., “Driving control of car-like robots by Path-generating regulator,” *Trans. of JSME (C)*. vol. 79, no. 801 (2013), pp. 1693-1708, (in Japanese).
- (16) Dai, J., Hanajima, N., Taruumi, Y., and Kazama, T., “The improvement of Path-generating regulator for four-wheeled car-like robots by the EKF-SLAM,” *International Session in RSJ2012* (2012), pp. 103-106.
- (17) Dai, J., Hanajima, N., Taruumi, Y., and Kazama, T., “Path-generating regulator for four-wheeled car-like robot navigation combined with EKF-SLAM,” *RSJ-HRT2013* (2013), pp. 138-141.

- (18) Yang, B., Hanajima, N., Yamamoto, A., Ayamura, M., and Dai, J., "Path-generating regulator along a straight passage for two-wheeled mobile robots," *Proc. of 2013 IEEE IRS/RSJ International Conference on Intelligent Robots and Systems* (2013), pp. 4837-4844.
- (19) Laumond, J. P., Sekhavat, S., and Lamiraux, F., "Guidelines in nonholonomic motion planning for mobile robots," *Lectures Notes in Control and Information Sciences 229*, Springer (1998), ISBN 3-540-76219-1.
- (20) Ma, B. L., Tso, S. K., and Xu, W. L., "Adaptive/robust time -varying stabilization of second-order nonholonomic chained form with input uncertainties," *International Journal of Robust and Nonlinear Control*, 2002.
- (21) Khennouf, H., and Canudes, C., "Quasi-continuous exponential stabilizers for nonholonomic systems," *Proc. of 13th IFAC World Congress* (1996), pp. 49-54.
- (22) Primbs, J. A., Nevistic, V., and Doyle, J. C., "Nonlinear optimal control: a control Lyapunov function and receding horizon perspective," *Asian Journal of Control*, vol. 1, no. 1(1999), pp. 14-24.
- (23) Matveev, A. S., Hoy, M., Katupitiya, J., and Savkin, A. V., "Nonlinear sliding mode control of an unmanned agricultural tractor in the presence of sliding and control saturation," *Robotics and Autonomous Systems*, vol. 61, issue. 9 (2013), pp. 973-987.
- (24) Ding, L., Gao, H. B., Deng, Z. Q., Li, Z. J., Xia, K. R., and Duan, G. R., "Path-following control of wheeled planetary exploration robots moving on deformable rough terrain," *The Scientific World Journal*. vol. 2013, Article ID 793526 (2014), 13 pages.
- (25) Yang, J. M., Kim, J. H., "Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots," *IEEE Transactions on Robots and Automation*, vol. 15, issue. 3 (1999), pp. 578-587.
- (26) Miao, H., Tian, Y. C., "Dynamic robot path planning using an enhanced simulated annealing approach," *Applied Mathematics and Computation*, vol. 222 (2013), pp. 420-437.
- (27) Chen, H., Wang, C. L., Yang, L. and Zhang, D., Semiglobal stabilization for nonholonomic mobile robots based on dynamic feedback with inputs saturation, *Journal of Dynamic Systems, Measurement, and Control*, Vol.134, Issue 4 (2012), pp.041006.1-8.
- (28) Khennouf, H. and Canudes, C., Quasi-Continuous exponential stabilizers for nonholonomic systems, *Proc. of 13th IFAC World Congress* (1996), pp.49-54.
- (29) Li, Y., He, L. Y. and Yang, L., Path-following control for multi-axle car-like wheeled mobile robot with nonholonomic constraint, *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics* (2013), pp.268-273.
- (30) Ma, B. L. and Tso, S. K., Unified controller for both trajectory tracking and point regulation of second-order nonholonomic chained systems, *Robotics and Autonomous Systems*, Vol.56, Issue 4 (2008), pp.317-323.
- (31) Primbs, J. A., Nevistic, V. and Doyle, J. C., Nonlinear optimal control: A control lyapunov function and receding horizon perspective, *Asian Journal of Control*, Vol.1, No.1 (1999), pp.14-24.
- (32) Rehman, F., Discontinuous steering control for nonholonomic systems with drift, *Nonlinear Analysis: Theory, Methods & Applications*, Vol.63, Issue 3 (2005), pp.311-325.

- (33) Wang, J. J., Dai, J., Taruumi, Y. and Hanajima, N., Feedback gain switching strategy for car-like robots based on Path-generating regulator, 17th Intelligent Mechatronics Workshop (2012), M4-1.
- (34) Xavier, J., Pacheco, M., Castro, D. and Ruano, A., Fast Line, Arc/circle and leg detection from laser scan data in a player driver, Proceedings of the 2005 IEEE International Conference on Robotics and Automation (2005), pp.3930–3935.