PATH TRACKING ALGORITHM FOR AN AUTONOMOUS GROUND ROBOT

NOORAIN BT MOHD JOHARY

A thesis submitted in
fulfillment of the requirement for the award of the
Degree of Master of Electrical Engineering

FACULTY OF ELECTRICAL AND ELECTRONIC ENGINEERING
UNIVERSITI TUN HUSSEIN ONN MALAYSIA

JAN 2014

ABSTRACT

This paper presents a path tracking algorithm using proportional-integral-derivative (PID) controller. In this paper the tracking problem of an autonomous ground robot is considered. A new control strategy is proposed to determine the tracking algorithm can deal with very large tracking error or not. Besides that, autonomous ground robots maybe have an overshoot and deviation occurs when following the path as the path are made of linear segments which lead to abrupt change at waypoint. Therefore, tracking control algorithm need to be implement so that the robot will able to track the planned path smoothly. Simulation results show that the proposed path tracking algorithms are computationally efficient and the path tracking using PID controller are capable of tracking the path.

ABSTRAK

Kertas ini membentangkan algoritma mengesan laluan menggunakan pengawal *proportional-integral-derivative* (PID). Dalam kertas ini masalah pengesanan untuk robot tanah autonomi dipertimbangkan. Satu strategi kawalan baru dicadangkan untuk menentukan algoritma pengesanan yang boleh menangani kesilapan pengesanan yang sangat besar atau tidak. Di samping itu, robot tanah autonomi mungkin mempunyai lajakan dan sisihan berlaku apabila mengikuti jalan yang diperbuat daripada segmen linear yang membawa kepada perubahan yang mendadak pada titik laluan. Oleh itu, pengesanan algoritma kawalan perlu dilaksanakan supaya robot akan dapat mengesan jalan yang dirancang dengan lancar. Keputusan simulasi menunjukkan bahawa algoritma pengesanan jalan dicadangkan adalah efisyen dan cekap dan pengesanan jalan menggunakan pengawal PID berkebolehan mengesan jalan yang dirancang.

CONTENTS

CHAPTER 2 LITERATURE REVIEW

CHAPTER 3 METHODOLOGY

CHAPTER 4 RESULT AND ANALYSIS

LIST OF FIGURES

LIST OF TABLE

LIST OF ABBREVIATIONS

PID	-	Proportional-Integral-Derivative
VWO	-	Variable waypoint offset
WP	-	Waypoint
SMC	-	Sliding Mode Control

LIST OF APPENDIX

CHAPTER 1

INTRODUCTION

1.1 Project Background

In the era of globalization, developing of autonomous robot has been a hot topic. Many ideas have been proposed and applied to the autonomous robots. Today, autonomous robots are attracting more and more attention worldwide in both academia and industry. The study of robotics has increased in the last decades and robots have become commercially available to the general public. There are many definition of the robot. Some defined a robot though the aspect of reprogram ability whiles other concerning on manipulation of the robot, behaviors, intelligence and so on. For path planning, the path is create for robot and robot need to find the fastest and safest way to get from starting point to goal point. While path tracking is a capability of a robot to followed the existing or reference path with respect to the path and path information. Path tracking is an important issue and one of the most fundamental problems in autonomous robots. Path tracking in robotics is the capability of a robot to follow a path that already exists considering the motion smoothness or the dynamic and kinematic constraints. Therefore a path tracking is necessary. In this project, the path tracker is realised using controllers to keep the robot on track and reduce the overshoot or deviation when following the path. There is various ways to control the movements of the robot. One of these methods is by using the PID controller. PID controller has many features that make it commonly use in close loop system. Among these features are its simple functionality and reliability.

1.2 Problem Statement

Most of path planning methods produce paths which consist of piece-wise linear segments. This in turn causes the paths have sharp corners, which are not feasible for mobile robot. While most of mobile robots are non-holonomic with kinematic and dynamic constraints, they are unable to traverse such paths. This needs the path to be smoothen considering the robots kinematic constraints such as minimum turning radius. Besides that, autonomous robot may experiences overshoot and deviation when following the path. Hence path smoothing or path tracking is needed to make the path satisfies the constraints. Therefore, path tracking algorithm for an autonomous ground robot need to be developed in order to ensure the robot are able to follow the reference path.

1.3 Objective

There are few objectives that need to be achieved at the end of this project. The objectives of this project are:

i. To propose a tracking algorithm for a car-like robot that can deal with very large tracking error.
ii. To implement tracking control algorithm so that the robot is able to track the planned path smoothly.
iii. To enhance the performance of the path tracking robot by using the PID controller algorithms.

1.4 Project Scope

In order to achieve the objectives of the project, several scopes have been outline. The following are the scopes of the project.

i. Use a suitable algorithm and controller to keep the robot on track and follow the path smoothly.
ii. The robot is able to follow the existing planned path with less overshoot at every corner of path.
iii. The algorithm performance is simulated using MATLAB IDE Software.
iv. The algorithm is implemented off-line.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Literature review is a process of collecting and analyse data and information which are relevant to this study. The required data and information can be collected through variable sources such as journals, articles, reference books, online database and others. This chapter consists or two parts. The first part will be a case study on previous projects that relates to this project while the second part will focus on the theory aspects of this project.

2.2 Path Planning

The control system of autonomous robot generally comprises a path planner and path tracking controller. The path error in the robot navigation primarily depends on the smoothness of the references in the planning stages. The various path planning method have been studies in previous work [3, 5, 6]. Path planning in robotics consists in the design of the best path between two given configurations. The path must avoid all obstacles present in the physical space, as well as satisfy any kinematical or dynamical defined constraint of the motion and decide the shortest path from the starting position to the target position.

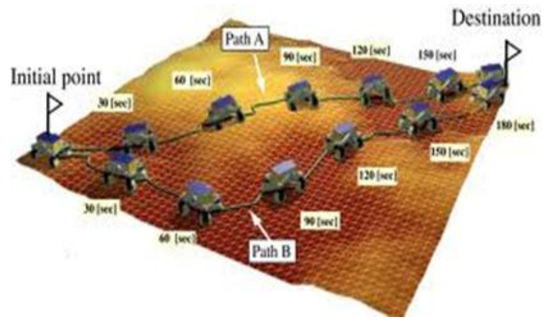

Figure 2.1: Path planning configuration

2.3 Path Tracking

Path tracking in robotics is capabilities of the robots to follow the path that already exist with considering the motion smoothness or the dynamic constraints. Path tracking controller can be classified into four categories which is linear, non-linear, geometrical and intelligent approaches [1]. The linear approaches are computationally simple, but the path tracking motion is inconsistent with size of the path errors. To cope with the problem, several authors have proposed non-linear path tracking algorithm. Those include a path tracking method based on Lyapunov function [2, 3] and a non-linear steering control law [4] considering the driving speed. The non-linear approaches, however consider only path error convergence or system stability, but neglect the smoothness of the transient trajectory or dynamic constraints. The geometric approaches are considered as attempts to connect the path tracking to path planning. Tsugawa [7] has presented a target point following algorithm in which a cubic spline curve is used to determine the steering angle or rotational velocity of the robot. This geometric scheme show the smooth tracking motion to guide the mobile robot towards the reference path, but neglect the dynamic constraints, such as the curvature or acceleration limits which are important factors for avoiding robot or wheel slippage or stray away from the path.

2.3.1 Kinematic Constraints

In this section, the error dynamic and kinematical constraints of robot are defined. For a mobile robot driven by two differential wheels, the center of motion, denoted by C is located at the midpoint between the left and right driving wheels. Assuming that the robot moves on the planner surface without slipping, the tangential velocity v_c and angular velocity ω_c at the center C can be written as [1]

$$v_c = \frac{r_w}{2} (\omega_r + \omega_1) \tag{2.1}$$

$$v_c = \frac{r_w}{d_w} (\omega_r - \omega_1) \tag{2.2}$$

where ω_r and ω_1 denote the rotational velocities of the right and left driving wheels, respectively, r_w is the radius of the wheels, and d_w is the azimuth length between the wheels. The kinematic equation of the mobile robot is given by

$$\dot{x}_c = v_c \cos\theta_c \tag{2.3}$$

$$\dot{y}_c = v_c \sin\theta_c \tag{2.4}$$

$$\theta_c = \omega_c \tag{2.5}$$

where coordinates (x_c, y_c) indicate the position of the robot with respect to the world coordinate system and θ_c is the heading angle of the robot. The triplet (x_c, y_c, θ_c) is used for defining the robot posture and represented by vector P. The posture of the robot can be estimated from integration of Equations (3)–(5). The integration is implemented by the following iterative algorithm called dead reckoning:

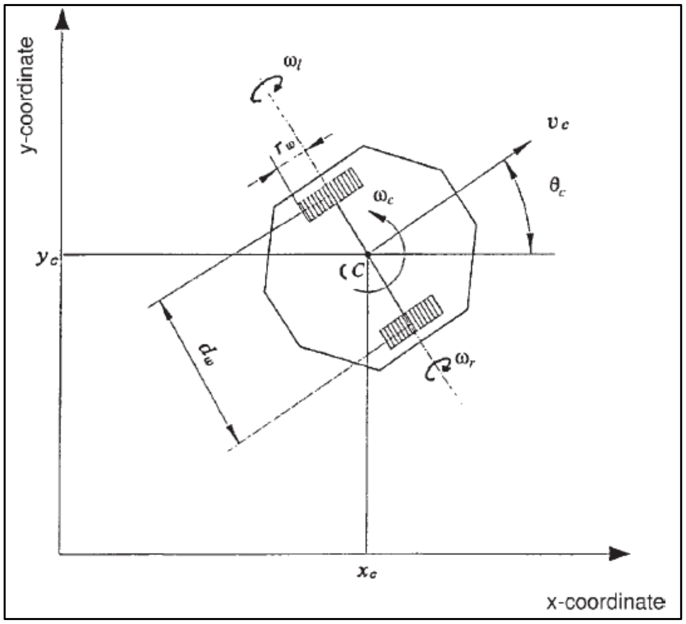

Figure 2.2: Definition of posture and velocities of two-wheeled mobile robot.

(i) In case $\omega_c \neq 0$

$$x_c^{k+1} = x_c^k + \frac{v_c}{\omega_c}[\sin(\theta_c^{k+1}) - \sin\theta_c^k], \tag{2.6}$$

$$y_c^{k+1} = y_c^k - \frac{v_c}{\omega_c}[\sin(\theta_c^{k+1}) - \cos\theta_c^k], \tag{2.7}$$

$$\theta_c^{k+1} = \theta_c^k + \omega_c t_s \tag{2.8}$$

(ii) In case $\omega_c = 0$

$$x_c^{k+1} = x_c^k + v_c t_s \cos(\theta_c^k), \tag{2.9}$$

$$y_c^{k+1} = y_c^k + v_c t_s \sin(\theta_c^k), \tag{2.10}$$

$$\theta_c^{k+1} = \theta_c^k \tag{2.11}$$

where k denotes the sampling index and t_s is the sampling time

2.3.2 Dynamic Constraints

Any abrupt change in the robot motion may cause the slippage or mechanical damage to the mobile robot [1]. If the angular acceleration of each driving wheel is limited by $\dot{\omega}_{\max}$

$$|\dot{\omega}_{r,1}| \leq \dot{\omega}_{\max} \tag{2.12}$$

then, from (1) and (2), the tangential and angular accelerations of the robot are bounded by

$$|a_c| + \frac{d_w}{2}|\alpha_c| \leq r_w \dot{\omega_{max}} \tag{2.13}$$

The above equation means that the maximum allowable bounds on tangential and angular accelerations of the robot are coupled with each other. The ranges of each value to be independently considered are obtained by taking half the value of each maximum as

$$|a_c| \leq a_{max} = \frac{r_w \dot{\omega_{max}}}{2} \tag{2.14}$$

$$|\alpha_c| \leq \alpha_{max} = \frac{r_w \dot{\omega_{max}}}{d_w} \tag{2.15}$$

where a_{max} and α_{max} are the tangential and angular acceleration limits of the robot, respectively.

2.4 Related Work

There have been several studies done before to develop the path tracking algorithm for an autonomous ground robot. The researched that had been studied is related to this project such as the method used to tracking the path.

K.C. Koh [1] has presented that dynamic constraints of the mobile robot should be considered in the design of path tracking algorithm. The driving velocity control law has been designed based on bang-bang control and the acceleration bounds of driving wheels need to be considered. The landing curve has been introduced as it works as an intermediate path smoothly steering the rotation of the robot towards the reference path. The target tracking algorithm used in this project is composed of two independent laws which is steering control law and velocity control law.

Sanhyuk [8] Park has studied the new guidance logic which is able to select a reference point on the desired trajectory and lateral acceleration command was generated by using the reference point. The several guidance logic have been developed which is the proportional derivatives controller (PD) has been used on the cross-track error, has an element of anticipation for the upcoming local desired flight path, and instantaneous vehicle speed was used in the algorithm. This kinematic factor adds an adaptive capability with respect to changes in vehicle inertial speed, due to the external disturbance.

Jeff Wit [9] has presented a new path tracking technique called ‘‘vector pursuit’’. This new technique is based on the theory of screws, which was developed by Sir Robert Ball in 1900. It generated a desired vehicle turning radius based on the vehicle’s current position and orientation relative to the position of a point ahead on the planned path and the desired orientation along the path at that point. The vector pursuit algorithm is compared to other geometrical approaches, and it is shown to be more robust, resulting in more accurate path tracking.

J. Giesbrecht [10] has presented the pure pursuit algorithm implementation and adaptation. Pure Pursuit algorithm was chosen for its accuracy, simplicity, adaptability and robustness. The Pure Pursuit algorithm was implemented in four different ways which is as a path tracker to follow the straight line between high level waypoints on a patrol mission, goal directedness has been provided to obstacle avoidance behaviour, a follower vehicle to pursue a lead vehicle via GPS breadcrumbs is allowed, and as a path tracker for a detailed on-line autonomous planner. This algorithm was devised to compute the arc necessary to return a vehicle back onto a path. It computes the curvature of an arc that a vehicle must follow to bring it from its current position to some goal position, where the goal is chosen as some point along the path to be tracked and the algorithm is extremely robust to poor sensing, poor actuation, combination with other control mechanisms, and is easily adapted for changing functionality. Another adjustable parameter has been implement is the radial tolerance that assigned to each waypoint. When the radial tolerance is set too low for the vehicle, the path is overshot at the corners. With a more realistic radial tolerance, the robot does not approach the waypoint itself as closely, but is able to adhere to the path more accurately.

R. Craig Conlter [11] has studied the implementation of the Pure Pursuit Path tracking Algorithm. The pure pursuit approached a method of geometrically determining the curvature that will drive the vehicle to a chosen path point, termed the goal point. The method itself is fairly straightforward. The only real implementation problems lie in deciding how to deal with the path information (communication, graphics, updating the path with new information from the planner). There is one parameter in the pure pursuit algorithm which is a lookahead distance. The effects of changing the lookahead distance must be considered within the problem faced such as regaining the path and maintaining the path.

Tao Dong [12] has presented path tracking and obstacle avoiding based on fuzzy logic approached. Fuzzy logic control algorithms are developed to achieve close path tracking while avoiding obstacles. The Fuzzy Logic Controller is activated when the obstacle sensor detects any obstacle. UAV velocity and heading angle change into corresponding different situations will be generated by the FLC. A two-layered FLC was used to make the UAV track its path while avoiding the fixed, but unexpected obstacles.

Jean-Matthieu bourgeot [13] has presented the path planner designed. It contains of two parts which is the references path need to be determined and tracking algorithm was applied which the robot followed the reference track. A 3D path planning method has been developed by using A* algorithm to find the easiest track biped robot, then the low level path tracking followed the path. For the path tracking strategy, heading and literal offset has been measured in the path tracking assignment.

G. Ambrosino [14] has studied the path generation and tracking algorithm for 3D UAV. The 3D path has been obtained by using Dubins Algorithm. One of the characteristic of the path generated by the proposed algorithm is composed by straight lines and circles/arcs of constant radii. The line-of-sight guidance algorithm has been used for path tracking algorithm. This algorithm is based only on the kinematic equations of motion. The algorithm for the path tracking guarantees, under specified assumptions the tracking error, both in position and in attitude, asymptotically tends to zero.

Jacky Baltes [15] has presented the used of reinforcement learning in solving the path-tracking problem for car-like robots. The most important concept in reinforcement learning is the agent and environment. In the path tracking problem, the reward is based on how well the agent tracked the given path. Reinforcement learning can be adapted to control a car in path tracking. The controller is needed to keeps the car on the track. The reinforcement controller is the only controller that has been used successfully to drive cars with and without linear steering behaviour.

Takeshi Yamasaki [16] has studied about a guidance and control system for a trajectory-tracking unmanned aerial vehicle (UAV). A proportional navigation guidance law is applied to a trajectory-tracking flight to achieve the robust trajectory-tracking guidance and control system. The system employed a dynamic inversion technique for the guidance force generation, which allows the UAV to maintain high maneuverability, and a simple velocity control to obtain a desired velocity. With the proportional navigation guidance, UAV may avoid its control saturation or divergence even in large tracking-error situations.

Guangfeng YUAN [17] has developed tracking control approach for a car-like robot based on backstepping techniques and a neural dynamics model. The proposed control algorithm can generate smooth and reasonable velocity commands and deal with arbitrarily large tracking error. The advantage using backstepping is simple and stable. While the disadvantage is has a speed jump that has caused huge acceleration. The tracking control model can produced a smoothly changing velocity curve with time. The stability of the control systems are analysed and proved using a Lyapunov stability theory.

A.Hemami [18] has proposed a new control strategy to determine the steering angle at each instant based on measured errors, the offset from the path and the deviation in orientation. The steering system is considered to control the angle of the steering wheel so that any deviation from the path is corrected in a stable manner and as fast as possible, and without oscillations about the path. Besides that, the dynamic equation of the vehicle is formulated to study the effect of a control strategy.

André KAGMA [19] has presented a method to track straight lines path with a car-like tricycle vehicle. Straight line tracking controller is used as a control strategy to track the path as a robot is moves. The aim of this project is to design a controller which makes the vehicle follow the X – axis. Kinematics of the tricycle robot has been considered in this project.

Arturo L.Rankin [20] has developed a path following strategy for autonomous steered-wheeled robotics vehicle where accuracy is easily attained. The specified path should not only be safe and efficient, it should be achievable. The vehicle must be able to evaluate how well it performed the task and the manual tuning of control parameters should be replaced by autonomous robot. Three types of steering control were autonomously tuned and evaluated which are proportional, integral, and derivative (PID) controls applied to the error in the vehicle heading, pure pursuit control, and a weighted PID/pure pursuit solution. Both PID and pure pursuit method of high-level steering control have advantages and disadvantages. PID method is stable when the vehicle velocity is small and the distance to the carrot is large. While the disadvantage is difficult to accurately tune its control parameter, unstable at high velocity, and causes corners to be cut due to an inherent error. The pure pursuit controller is easy to tune and performed well but if the literal error is large, this method becomes unstable. Stability can be improved by using an adaptive rather than standard pure pursuit controller.

Table 2.1 shows the list of related works and project description that have been done before.

Table 2.1: List of Related Works

Project Title	Year	Author	Project Description
Path-tracking Control of Non-holonomic Car-like Robot With Reinforcement Learning	-	Jacky Baltes and Yuming Lin Centre for image Technology and Robotics University of Auckland, Auckland New Zealand.	-Used reinforcement Learning and Reinforcement Controller.
A Simple Path Tracking Controller For Car-Like Mobile Robots	-	André Kamga and Ahmed Rachid System automatic Laboratory France.	-Design and used straight line tracking controller as a control strategy to track the path. -Consider kinematic of the tricycle robot.
A New Control Strategy for Tracking in Mobile Robots and AGV's	1990	A. Hemami, M.G. Mehrabi, and R.M.H. Cheng Department of Mechanical Engineering Concordia University, Canada	-Proposed a new control strategy that considers steering system. -Formulated dynamic equation of the vehicle.
Implementation of the Pure Pursuit Path tracking Algorithm	1992	R. Craig Conlter The Robotics Institute Camegie Mellon University Pittsburgh, Pennsylvania	-Implement pure pursuit path tracking Algorithm.
Development of path tracking software for an autonomous steered-wheeled robotic vehicle and its trailer	1997	Arturo L.Rankin University of Florida	-Developed a path following strategy where accuracy is easily attained -Used three types of steering control 1.PID Controller 2.Pure pursuit control 3.Weighted PID/pure pursuit solution
A Smooth Path Tracking Algorithm for Wheeled Mobile Robots With Dynamic Constraints	1999	K. C. KOH Division of mechanical and control engineering Sun Moon University Korea H. S. CHO Department of Mechanical Engineering, KAIST	-Used time optimal bang-bang control for path tracking algorithm. -Introduced a landing curve so that mobile robot could softly land on the target line. -Target tracking algorithm used is steering control law and driving velocity control law.

Autonomous Ground Vehicle Path Tracking	2000	Jeffrey S. Wit University of Florida	-Used vector pursuit tracking technique.
Tracking Control of a Mobile Robot Using Neural Dynamics Based Approaches	2001	Guanfeng Yuan The Faculty of Graduate Studies of The University of Guelph	-Used backstepping techniques and a neural dynamics model. - The stability of the control systems are analysed using a Lyapunov
Path Planning And Tracking in a 3D Complex Environment for an Anthropomorphic Biped Robot	2002	Jean – Matthieu Bourgeot, Nathalie Cislo, Bernand INRIA Rhone-Alpes, BIP Project, Montbonnot, France.	-3D path planning method develop by using A* algorithm. - Path tracking strategy measure heading and literal offset.
A new nonlinear Guidance Logic for Trajectory Tracking.	2004	Sanhyuk Park, John Deyst and Jonathan P.How Massachusetts Institute of Technology, Cambridge, MA,USA	-Used proportional derivative (PD) controllers. -Three purposes angle η used in the guidance logic 1. Provides a heading correction 2. Provides PD control on cross track 3. Provides an anticipatory acceleration command to exactly follow a circular reference trajectory. -Used instantaneous vehicle speed in the algorithm.
Path Tracking for Unmanned Ground Vehicle Navigation	2005	J. Giesbrecht, D. Mackay, J. Collier, S. Verret DRDC Suffield Defence Research and Development Canada	-Used pure pursuit algorithm. -Implement the radial tolerance waypoints.
Path Tracking and Obstacle Avoidance of UAVs - Fuzzy Logic Approach	2005	Tao Dong, X. H. Liao, R. Zhang, Zhao Sun and Y. D. Song Department of Electrical and Computer Engineering North Carolina A&T State University, USA	-Used fuzzy logic based approach to path tracking and obstacle avoiding. -Used fuzzy logic controller.
Algorithms for 3D UAV Path Generation and Tracking	2006	G. Ambrosino, M. Ariola, U. Ciniglio, F. Corraro, A. Pironti and M. Virgilio Proceedings of the 45th IEEE Conference on Decision & Control USA,	-Used Dubins Algorithm for 3D Path. -Used line-of-sight guidance algorithm for tracking path.

Robust Trajectory-Tracking Method for UAV Guidance Using Proportional Navigation	2007	Takeshi Yamasaki, Hirotoshi Sakaida, Keisuke Enomoto, Hiroyuki Takano and Yoriaki Baba Department of Aerospace Engineering, National Defense Academy, Kanagawa, Japan	- Used proportional navigation guidance law. - Employed a dynamic inversion technique for the guidance force generation.

From the project reviews, many paths tracking algorithm method can be used in order to track the path. But for this project, the path tracker is realised using controllers to keep the robot on track and reduce the overshoot or deviation when following the path.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter will describe the overall process to develop this research project, method and technique approach to complete the project. To accomplish it successfully, the method and technical strategy implied is the most important disciplined need to be concerned.

3.2 Project Methodology Flowchart

Figure 3.1 shows the flowchart of the methodology to conduct the project. Firstly, it is needed to search and study about the literature review that related from journal, relevant paper and publication.

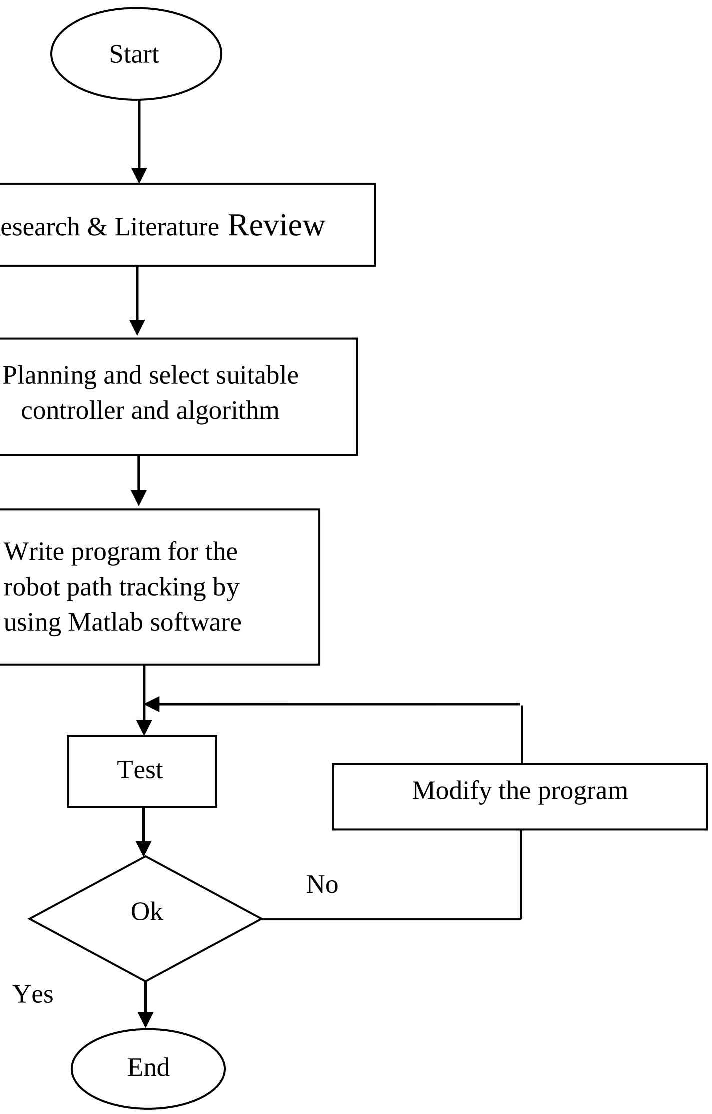

Figure 3.1: Methodology of Project

3.2.1 Project flowchart

Figure 3.2 shows the flow chart for this project. It consist of the process to tracking the path by considering the prior information and path tracking algorithm is applied in order to track the existing path. Path tracker is realised using controllers to keep the robot on track and reduce the overshoot or deviation when following the path.

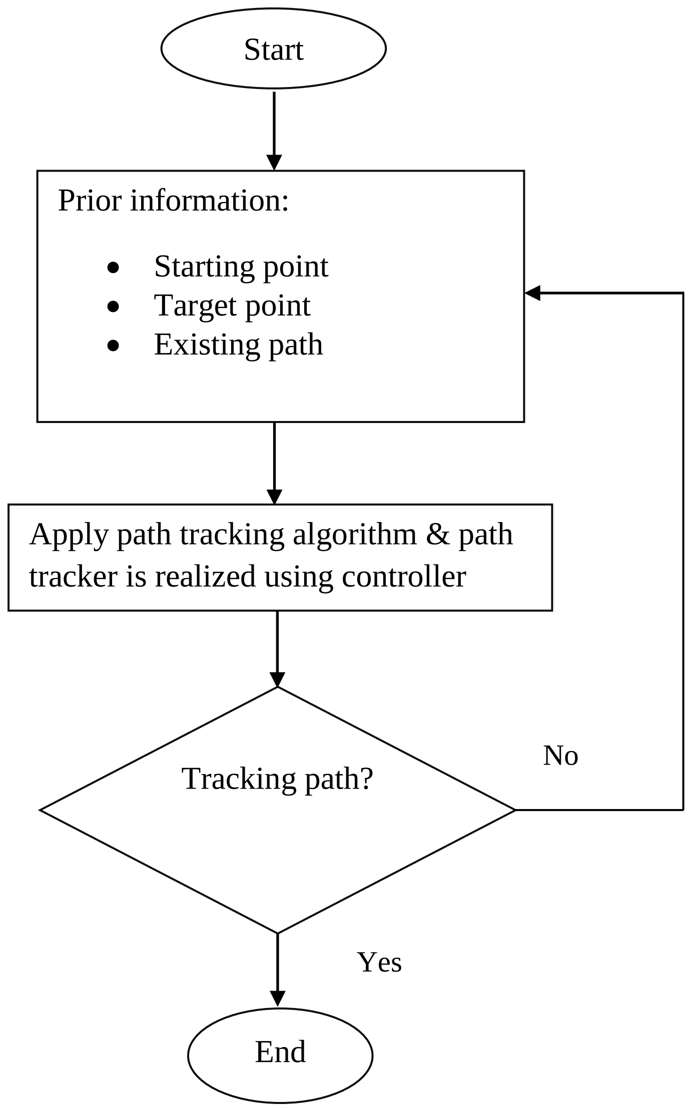

Figure 3.2: Flowchart of the project

3.3 Path Tracking

Path tracking is a capability of robot to follow the existing or reference path with respect to the path and path information. Path tracking is an important issue and one of the most fundamental problems in mobile robots. The purpose of path tracking is to follow the planned path smoothly by considering the kinematic and dynamic constraint. Therefore the controller is needed which keep the robot on track and reduce the overshoot or deviation when following the path.

3.3.1 PID Controller

A proportional-integral-derivative controller (PID controller) is a generic control loop feedback mechanism (controller) widely used in industrial control systems. A PID controller calculates an "error" value as the difference between a measured process variable and a desired set point. The controller attempts to minimize the error by adjusting the process control inputs.

The PID controller calculation algorithm involves three separate constant parameters, and is accordingly sometimes called three-term control: the proportional, the integral and derivative values, denoted P, I, and D. Simply put, these values can be interpreted in terms of time: P depends on the present error, I on the accumulation of past errors, and D is a prediction of future errors, based on current rate of change.

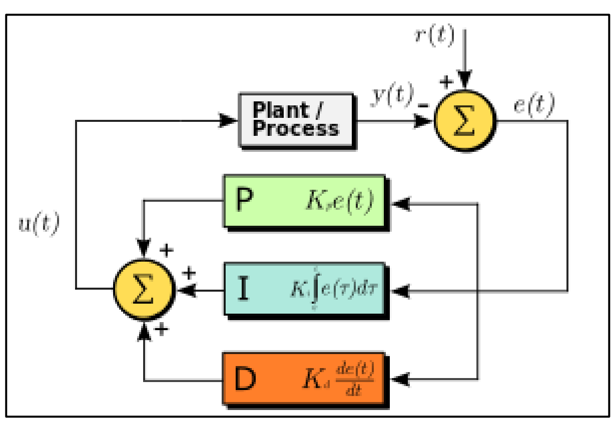

Figure 3.3: A block diagram of a PID controller in a feedback loop

The PID algorithm is described by

$$u(t) = K_P e(t) + K_i \int e(t)dt + K_d \frac{de}{dt} \tag{3.1}$$

where y is the measured process variable, r the reference variable, u is the control signal and e is the control error $e = y_{sp} - y$. The reference variable is often called the set point. The variable (e) represent the tracking error, the difference between the desired input value (γ) and the actual output(y). This error signal (e) will be sent to the PID controller, and the controller computes both the derivative and the integral of this error signal. The control signal (u) to the plant is equal to the proportional gain (K_P) times the magnitude of the error plus the integral gain (K_i) times the integral of the error plus the derivative gain (K_d) times the derivative of the error.

This control signal (u) is sent to the plant, and the new output (y) is obtained. The new output (y) is then fed back and compared to the reference to find the new error signal (e). The controller takes this new error signal and computes its derivative and its integral again until the output obtain is satisfied. The control signal is thus a sum of three terms, the P-term (which is proportional to the error), and the I-term (which is proportional to the integral of the error), and the D-term (which is proportional to the derivative of the error). The controller parameters are proportional gain K_p, integral time T_i, and derivative time T_d.

The transfer function of a PID controller is found by taking the Laplace transform of Eq. (1).

$$K_P + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_P s + K_i}{s} \tag{3.2}$$

K_P = Proportional gain

K_i= Integral gain

K_d= Derivative gain

3.3.2 Characteristic of P, I, and D controllers

A proportional controller (K_P) will have the effect of reducing the rise time and will reduce but never eliminate the steady-state error. An integral control (K_i) will have the effect of eliminating the steady-state error for a constant or step input, but it may make the transient response slower. A derivative control (K_d) will have the effect of increasing the stability of the system, reducing the overshoot, and improving the transient response. The effects of each of controller parameters K_P, K_i, and K_d on a closed-loop system are summarized in the table below.

Table 3.1: Effects of each of controller parameters

CL Response	Rise Time	Overshoot	Settling Time	S-S Error
K_p	Decrease	Increase	Small Change	Decrease
K_i	Decrease	Increase	Increase	Eliminate
K_d	Small Change	Decrease	Decrease	No Change

These correlation may not be exactly accurate, because K_P, K_i, and K_d are dependent on each other. In fact, changing one of these variables can change the effect of the two other. The table should only be used as a reference when determining the values for K_P, K_i, and K_d.

3.3.2.1 Proportional Action

The proportional term produces an output value that is proportional to the current error value. The proportional response can be represents as per equation below:

Where

$$P_{out} = K_p . e(t) \tag{3.3}$$

Pout: Proportional term of output

Kp: Proportional gain, a tuning parameter e:

Error = SP − PV

t: Time or instantaneous time (the present)

3.3.2.2 Integral Action

The contribution from the integral term is proportional to both the magnitude of the error and the duration of the error. Summing the instantaneous error over time (integrating the error) gives the accumulated offset that should have been corrected previously. The accumulated error is then multiplied by the integral gain and added to the controller output [10]. The magnitude of the contribution of the integral term to the overall control action is determined by the integral gain, K_i. The integral term is given by:

$$I_{out} = K_i \int e(\tau)\, d \qquad (3.4)$$

Where

Iout: Integral term of output

K_i: Integral gain, a tuning parameter

e: Error = SP − PV

τ: Time in the past contributing to the integral response

3.3.2.3 Derivative Action

The rate of change of the error is calculated with respect to time, multiplied by another constant D, and added to the output. The derivative term is used to determine a controller's response to a change or disturbance of the process. The derivative term is given by:

$$D_{out} = K_d \frac{d}{dt} e(t) \qquad (3.5)$$

where

D_{out}: Derivative term of output

K_d: Derivative gain, a tuning parameter

e: Error = SP − PV

t: Time or instantaneous time (the present)

3.4 Path Tracking Algorithm

To achieve the project goal in well-organized manner, the actual algorithm is implemented as follows.

X_1 = Current point

X = Previous point

$\dot{X}$ = Course rate change

ϕ_1 = Current steering angle

ϕ = Previous steering angle

ϕmax = Maximum steering angle

$\dot{\phi} max$ = Maximum steering rate

$\Delta\phi$ = Difference between the ϕ and ϕ_1

dt = Sampling Time

1. Find ground robot course angle from current point, $X_1$to the next waypoint.
2. Calculate the difference between the previous (X) and current point, X_1
3. Calculate course rate change, $\dot{X}$: $\dot{X}$ = $(X_1 - X)/dt$.
4. Calculate current steering angle, ϕ_1
5. Calculate the difference, $\Delta\phi$ between the previous steering angle, ϕ and ϕ_1.
6. If $\Delta\phi$ >$\dot{\phi} max \times dt$, set $\Delta\phi$ =$\dot{\phi} max \times dt$.
7. If ϕ > ϕmax , limit ϕ to ϕmax.
8. Do correction to the rest

3.5 Path Tracking Using PID Controller

In the PID Controller, the control law is determined based on the following errors:

(i) Heading error, ψ_{err}. (ii) Cross-track error, Y_{err}.

The reference heading angle is calculated from two consecutive waypoints
WP of the initial path:

$$\psi ref_j = \tan^{-1}((WP_{yj} - WP_{yj-1)}/((WP_{xj} - WP_{xj-1})) \tag{3.7}$$

j and j − 1 are the current and previous waypoint indexes within which the path is calculated.

The actual heading angle of the path is derived using the previous and current positions.

$$\psi_k = \tan^{-1}((y_k - y_{k-1})/(x_k - x_{k-1})) \tag{3.8}$$

ψ_{err} is calculated from the difference between the reference heading angle, ψ_{ref} and the current actual heading angle, ψ_k.

$$\psi_{err} = \psi ref_j - \psi_k \tag{3.9}$$

Also, the shortest distance from actual position to the reference path, Y_{err} is needed to design the controller for robot and can be calculated using linear equation. From ψ_{err} and Y_{err}, a control law using PID controller is then formulated to regulate the current path state. P term in PID controller is defined as

$$P = K_{P1}\psi_{err} + K_{P2}Y_{err} \tag{3.10}$$

with K_{P1} and K_{P2} are the proportional gains. The I term is

$$I_1 = K_{I1}(\psi_{err}(j) - \psi_{err}(j-1))/\Delta t \tag{3.11}$$

$$I_2 = K_{I2}(\psi_{err}(j) - \psi_{err}(j-1))/\Delta t \tag{3.12}$$

The D term is

$$D_1 = K_{D1}(\psi_{err}(j) - \psi_{err}(j-1))/\Delta t \tag{3.13}$$

$$D_2 = K_{D2}(\psi_{err}(j) - \psi_{err}(j-1))/\Delta t \tag{3.14}$$

REFERENCES

[1] K. C. Koh and H. S. CHO, A Smooth Path Tracking Algorithm For Wheeled Mobile Robots With Dynamic Constraints, *Journal Of Intelligent And Robotics System* (1999) 24: 367-385.

[2] Hemami, A., Mehrabi, M. G., and Cheng, R. M.: Synthesis For An Optimal Control Law For Path Tracking In Mobile Robots, Automatica 28(2) (1992), 383–387.

[3] Kanayama, Y., Kimura, Y., Miyazaki, F., and Noguchi, T.: A Stable Tracking Control Method for an Autonomous Mobile Robots, in: Proc. of the *IEEE Internat. Conf. on Robotics and Automation,* (1990), pp. 384–389.

[4] Hemami, A., Mehrabi, M. G., and Cheng, R. M.: A New Strategy for Tracking in Mobile Robots and AGV's, in: Proc. of the *IEEE Internat. Conf. on Robotics and Automation,* (1990), pp. 1122–1127.

[5] Elnagar, A. and Basu, A.: Piecewise Smooth and Safe Trajectory Planning, Robotica 12 (1994), 299–307.

[6] Tounsi, M. and Le Corre, J. F.: Trajectory Generation for Mobile Robots, Mathematics and Computers in Simulation, 1996, pp. 367–376.

[7] Tsugawa, S. and Murata, S.: Steering Control Algorithm for Autonomous Vehicle, in: Proc. of the Japan-USA Symp. on flexible automation, (1990), pp. 143–146.

[8] Sanhyuk Park, John Deyst and Jonathan P.How.: A new nonlinear Guidance Logic for Trajectory Tracking, Massachusetts Institute of Technology, Cambridge, (2004) MA, 02139, USA

[9] Jeffrey S. Wit: Autonomous Ground Vehicle Path Tracking, University of Florida (2000).

[10] J. Giesbrecht, D. Mackay, J. Collier, S. Verret: Path Tracking for Unmanned Ground Vehicle Navigation, DRDC Suffield Defence Research and Development Canada (2005).

[11] R. Craig Conlter.: Implementation of the Pure Pursuit Path tracking Algorithm, The Robotics Institute Camegie Mellon University Pittsburgh, Pennsylvania (1992).

[12] Tao Dong, X. H. Liao, R. Zhang, Zhao Sun and Y. D. Song.: Path Tracking and Obstacle Avoidance of UAVs - Fuzzy Logic Approach Department of Electrical and Computer Engineering, North Carolina A&T State University, USA (2005).

[13] J.Matthieu Bourgeot, Nathalie Cislo, Bernand.: Path Planning And Tracking in a 3D Complex Environment for an Anthropomorphic Biped Robot, (2002) INRIA Rhone-Alpes, BIP Project, Montbonnot, France.

[14] G. Ambrosino, M. Ariola, U. Ciniglio, F. Corraro, A. Pironti and M. Virgilio.: Algorithms for 3D UAV Path Generation and Tracking, *Proceedings of the 45th IEEE Conference on Decision & Control* (2006) USA.

[15] Jacky Baltes and Yuming Lin.: Path-tracking Control of Non-holonomic Car-like Robot with Reinforcement Learning, Centre for image Technology and Robotics University of Auckland, Auckland New Zealand.

[16] Takeshi Yamasaki, Hirotoshi Sakaida, Keisuke Enomoto, Hiroyuki Takano and Yoriaki Baba.: Robust Trajectory-Tracking Method for UAV Guidance Using Proportional Navigation Department of Aerospace Engineering, National Defense Academy, Kanagawa, Japan (2007).

[17] Guanfeng Yuan.: Tracking Control of a Mobile Robot Using Neural Dynamics Based Approaches The Faculty of Graduate Studies of The University of Guelph (2001).

[18] A. Hemami, M.G. Mehrabi, and R.M.H. Cheng.: A New Control Strategy for Tracking in Mobile Robots and AGV's, Department of Mechanical Engineering Concordia University, Canada (1990).

[19] André Kamga and Ahmed Rachid.: A Simple Path Tracking Controller for Car-Like Mobile Robots, System automatic Laboratory France.

[20] Arturo L.Rankin.: Development of path tracking software for an autonomous steered-wheeled robotic vehicle and its trailer, University of Florida (1997).

[21] K. Astrom and T. Hugglund(1995), Controllers: Theory, Design and Tuning, 2nd edition ISBN 1-55617-516-7.