e-Yantra Summer Internship Programme 2015

IIT Bombay

PID based Path Planning

Interns:

Dhirendra Sagar
Email:sagar.dhirendra@gmail.com
Mob: 8858643336

Uttam Kumar Gupta
Email:23guptageek@gmail.com
Mob: 9473735800

Mentor:

Amiraj Dhawan

Under the guidance of
Prof.Kavi Arya

5 June 2015
to
8 July 2015

Contents

Abstract

Title of the project PID based Path Planning suggest moving of Firebird V robot using Proportional-Integral-Derivative (PID) controller. Here we are controlling the robots motion using a PID based closed loop feedback system. We will discuss the effect of each of the PID parameters and demonstrate how to use a PID controller to improve the system performance.

To understand the working of PID we first implemented it on White Line Follower using 7-white-line-sensor-strip and then applied it's implementation in Image Processing to control the firebird using video streaming taken from webcam on a host PC and then command the firbird using xbee serial communication to follow a path to a movable destination point containing various obstacles in between them.

Objective of the work

The aim of the project is to detect Firebird Robot V using Image Processing in an arena containing Obstacle and a given movable end location, Plan the Robots Motion using PID closed loop feedback system..

Sr.No	**Tasks**	**Deadline**
1.)	Learning Fire Bird V programming, PID controller	4 days
2.)	Develop PID controller for white line follower	3 days
3.)	Tune PID controller for white line follower	5 days
4.)	Learning Python programming, OpenCV ,Xbee serial communication	4 days
5.)	Develop Motion Commands and communication between firebird V and laptop	2 days
6.)	Detection of Firebird V using image processing	2 days
7.)	Map the path for moving destination point with avoiding walls	5 days
8.)	Develop the PID controller for Movement of fire bird V on mapped path	2 days
9.)	Tune the PID controller for smoother movements	2 days
10.)	Testing the movements with different walls configuration	1 day
11.)	Tutorial and documentation of code	3 day

Completion

1. Developed and tuned the PID Controller for White Line Follower.

2. Slave bot is successfully detected and controlled using Image Processing.

3. Slave robot followed manually controlled Master robot with avoiding walls of green color.

Requirement

- **Atmel Studio 6.0**: For embedded C programming.
- **Python 2.7**: Python 2.7 IDLE is used for python programming.
- **openCV**: This library provides various modules which are used for processing an image. Color masking, Contours detection etc, are done by it.
- **numpy**: numpy used to handle array in python
- **X-CTU**: For configuration of xbee modules to send motion commands from laptop to fire bird
- **pyserial**: To send command using xbee serial communication through python.

Implementation

Task 1 - Learning Firebird V programming, PID controller

- Learned Fire bird V programming through Tutorial provided with kit
- For study of PID controller followed these links:
  control of mobile robots from http://www.coursera.org/course/conrob
  and
  http://www.robotix.in/tutorials/categ/avr/pid

Task 2 - Develop PID controller for white line follower

- To develop a PID controller for white line follower first we use three white line sensor strip, but it didn't workout.

- Then we used 7 white line sensor strip to follow the line, for that we need to connect a jumper at J4 and intialise port spi pin configuration to 4 White Line sensors are connected at slave microcontroller on ATMEGA 8.

- From Left to Right each sensor is given weight of -3 to 3. Reading of each sensors is multiplied by respective weights.

- Reading is taken from WL sensors and multiplied with weight and sum is fed to PID controller to Process it's mechanism and return the correction value which in the end is added to the motor speed to control the Firebird to be on white line.

Task 3 - Tune PID controller for white line follower

- We Mapped the correction value to the range of 0 to 255 as the Firebird can achieve max speed of 255(reference) only.

- To tune the PID controller created for white line We changed the Kp, Ki, and Kd values.

- For simpicity xbee is interfaced with the fire bird for changing value of P, I, and D by 0.1 to see the effect of these values for PID controllor tuning.

- To get a proper method for tuning tried ZieglerNichols to tune pid.
  - For proper tuning we started with Proportional P (I, D are 0) and followed a white line.
  - Then introduced Derivative D (I is 0) to improve the response time.
  - And finally added Integral I to get a better settling time.

- Successfully ran fire bird V on white line having curves and right angle path in it.

Task 4 - Learning Python programming, OpenCV, Xbee serial Communication

- To understand Python programming followed
  https://docs.python.org

- For understanding OpenCV We followed
  http://docs.opencv.org

- For serial communication we used xbee modules. There are two xbee modules one is interfaced with firebird and other to laptop. To configure xbee's we used X-CTU.

Task - 5: Develop Motion Commands and communication between firebird V and Laptop

- In this task we installed pyserial to communicate with fire bird V and python serial library is imported.
- We sent hex code of 8,6,4,2 and 5 from host device via serial xbee communication using python and recieved by Firebird V programmed in embedded C.
- Using hex code of 8 Bot moves forward, for 6 right, for 4 left, 2 for back and 5 for stop.

Task - 6: Detection of Firebird V using Image Processing

- To detect bot we placed markers on it using Pink and sky blue color.
- Video straming feedback taken from cam in RGB format is converted into HSV format.
- Mask the color on bot and store the Centroid of masked Contour.

Task - 7: Map path for moving destination point while avoiding walls

- To map the Path from source to destination point, image(video frames) is converted into 30*30 grid.
- Map Fire bird V Markers and walls in grid.
- Path from Source to Destination mapped using Heapq algorithm for maze solving.
- Path is drawn after connecting all two successive grid coordinates returned by Heapq Algorithm.

Task - 8: Develop the PID controller for movement of fire bird V on Mapped path

- Here We Created a PID controller for following path.
- The input of the PID controller is the angle for slave Bot's rear markers and path mapped.

- If slave bot's front marker is in left of path, error is negative and PID controller process on this error and controls the bot motor speed to be in path and vice versa for positive errors.

Task - 9: Tune the pid controller for smoother movement

- The experience we gained in tuning PID Controller for White Line Follower is applied here.
- Slave bot successfully smoothly followed the moving master bot while avoiding walls.

Task - 10: Testing the movements with different walls configuration

- To test the movement of bot we tried a lot of configuration of walls and bot is following the mapped path smoothly and accurately while avoiding walls.

Task - 11: Tutorials and documentation of code

- Tutorials are created on PID based White Line Follower.
- Documentation is done for Image Processing based path planning .

Result and Discussion

- The white line follower is working fine but still can be tuned better for more smoother movements.
- We can use auto PID tuner to tune it to perfection.
- PID controller output can be mapped to the system to see the control graph.
- Using Image Processing and serial communication we controlled slave bot to follow a moving master bot while avoiding obstacles.
- In image processing masking a color is tough task as the contrast changes for image and video taken on different lightening condition, so we only worked with video streaming.
- To overcome with the problem of masking we tried variable masking but it didn't workout well.
- Path mapping based on grid map is creating problem in avoding walls, we need to dilate walls size to overcome the problem of bumping the bot with walls which is not a good practice.
- It can be improvised using local path and then global path planning.

Problems Faced

- The main problem faced are following:
  1. Installation of Ubuntu.
  2. Finding a Error value for PID controller.
  3. Mapping Error to the motor speed.
  4. Calibration of 7 White Line sensor with master microcontroller.
  5. Masking Color with varying lighting condition.
  6. Check for the orientation of Slave bot in arena.

References

- http://www.coursera.org/course/conrob
- http://www.robotix.in/tutorials/categ/avr/pid
- https://docs.python.org
- http://docs.opencv.org
- https://www.pololu.com
- https://en.wikipedia.org/wiki/ZieglerNichols method
- http://stackoverflow.com/