The Racing Corgi Robot

1st Nattawat Lertpanitarn
marksarapornvisit@gmail.com

2nd Karnrawee Manochomphu
karnrawee.gaaw@gmail.com

3rd Tharuesida Teerathorndamrongdej
t.tharuesida@gmail.com

4th Patcharapong Thongchim
Answersir@hotmail.com

5th Natthanun Chawpongpang
Natlovelerd@gmail.com

6th Phakawat Chullamonthon
Zeno.phakhawat@hotmail.com

7thNithi Anutarawekin
Nithi2909@hotmail.com

8th Charunwat Jongprasert
Worradanai@gmail.com

9th Benjamas Pranomruttanarug
Benjamas.p@kmutt.ac.th

Department of Control System and Instrumentation Engineering,

King Mongkut's University of Technology Thonburi Bangkok, Thailand

***Abstract*— This article introduces an innovative design of the autonomous robot racing name "Corgi". The Corgi racing robot can autonomously operate at high speed with the complexity of the system developed. The car is built based on RC car which its chassis and suspension are used in the Corgi. The autonomous system composed of two microcontrollers and a microprocessor used to control all component in the car system. The self-driving system incorporates three basic behaviors which are obstacle avoidance, lane tracking, and direction control. This behavior performed using the capability of image processing, LiDAR surveying and rack and pinion servo. Safety function is also included in the system for emergency stop purpose which could be activated by a push button and wireless devices. More details Corgi are going to be explained in next and next chapter.**

Keywords—Andaman, racing car, corgi, robot, odroid, competition (key words)

I. INTRODUCTION

Nowadays, autonomous driving technology has gained attention in automotive industry. Since Electric Vehicles (EVs) are becoming more popular due to their affordable cost and less pollutions to the environment, Intelligent Assistive Transportation System (IATS) has been developed to assist drivers in many ways [1]. For example, a traffic status monitoring system determines the status of traffic of a particular region from real time data accumulated from smartphones [2], an autonomous parallel parking system uses an advanced control scheme to assist a driver in parking the vehicle [3], a technique for ground vehicles localization makes use of stereo vision combined with a laser range finder and a GPS receiver allowing the driver to know his current location [4], etc.

There are many competitions in autonomous vehicles running each year, such as, Autonomous Vehicle Competition at SparkFun Electronics [5], DARPA Grand Challenge [6], Robotics Competition: 25th Intelligent Ground Vehicle Competition [7], and International Autonomous Robot Racing Challenge [8]. The main objective of these competitions is to gain and share a design experience that is at the very cutting edge of engineering education. Moreover, the competitions have proved the possibility of autonomous driving to be true in the near future.

For this autonomous robot racing challenge [8], the paper presents the framework of our autonomous vehicle, the racing corgi robot, including the hardware design and the software systems composed of environment recognition system, path planning system, object avoidance system, and steering control system.

II. HARDWARE DESIGN

The Corgi Racing Robot developed by the authors is presented in figure 1. The chassis is modified from a toy car. The Corgi has two level of a usable area which is upper deck and lower deck from a newly built body. The body is built from a cut aluminum sheet which creates usable areas when make an assemble. The upper deck is used to install three components which are Odroid, Lidar sensor, and emergency stop button pole. The lower deck is used to install Uno Arduino, battery and many electrical devices. The drive motor, power transmission, suspension and steering servo motor are located on the under of lower deck which are placed on the toy car chassis. The ESP8266, webcam and emergency stop button are located on the pole on the rear of the car. Some of the car parts are created with 3D print technology which is pole holder and webcam housing. Those mentioned above work together to be the Corgi Racing Robot. More details are shown in the tables below.

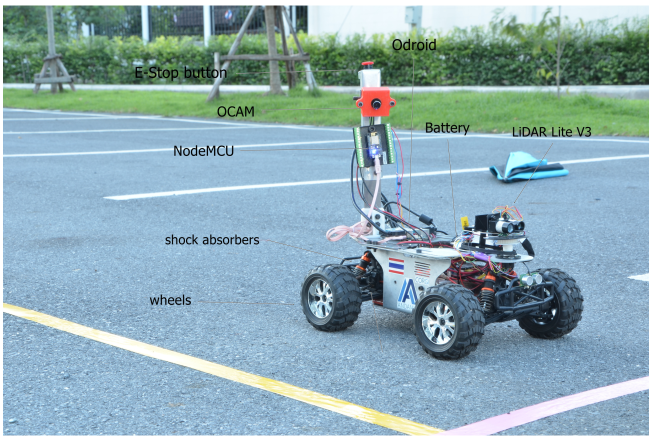


Figure 1. The Corgi Racing Robot

Table 1. The Corgi Racing Robot specification

Parameter	**Value**
Length	45cm
Width	33cm
Height	53cm
Weight	4.54kg
Range of wireless E-stop	35feet
Wireless E-stop delay	2sec

III. OVERALL ARCHITECTURE

Figure 2 shows the overall architecture diagram of the racing corgi robot. The main principle is to use the image-based information obtained from the webcam to start and navigate the vehicle using the computer controller. Firstly it detects a green light from stream images to start the vehicle by sending an initiate command to an Arduino controller, or Arduino I in figure 2. The Arduino controller is used for collecting all data from the emergency stop, ultrasonic sensor, and initial/interrupt command from the computer controller. It then sends a speed command to another Arduino controller, or Arduino II to process the data and control the speed motor and steering motor, consequently. The lane detection system with visual servoing works together with the object avoidance system in order to navigate the vehicle so that the vehicle stays in the boundary and is able to avoid either dynamic or static obstacles in the front while racing. The desired steering angle sent from the computer controller is directly sent to Arduino II to adjust the vehicle speed by assuming that if the steering rate is low, the vehicle should race fast. However, the vehicle speed depends on the distance detected from the ultrasonic sensor as well. If the distance is far, the vehicle should race fast. More information about speed control will be discussed later.

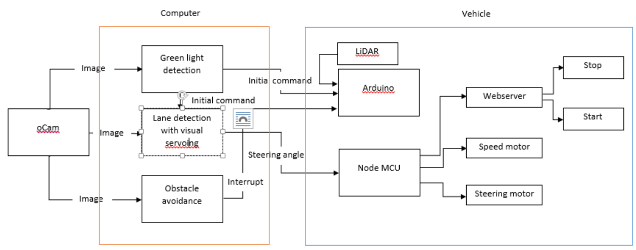


Figure 2. Overall architecture diagram

IV. SOFTWARE ALGORITHMS IN CONTROLLER DEVICES

A. Green light detection

Greenlight detection the most important algorithm that has an easy concept. With capable of image processing that allows the robot to sense the green light which is the sign that competition is started. An image is captured with OCAM 1-GCN-U in RGB color format. And then increased a robustness to lighting the changes by converting picture format into HSV color format. Once HSV ranges of the green light are defined. The condition of a vehicle start is also defined. Greenlight detection accuracy is enhanced by limit the detection zone to the left or right edge to reduce a chance to detect the wrong object. The basic principle of the algorithm is shown in figure 3.

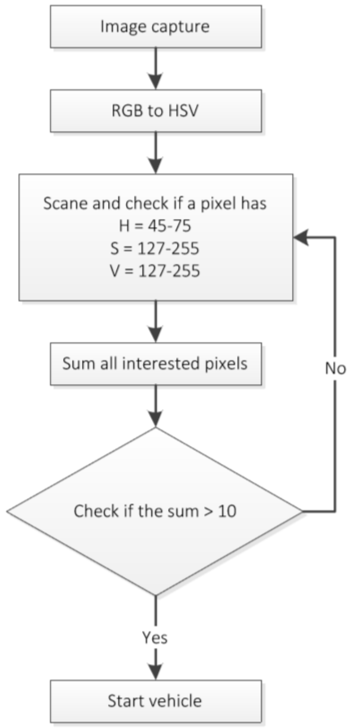


Figure 3. Green light detection algorithm flow chart

B. Lane detecion

Since the robot has the capability of image processing, so we take advantage to make the robot see things like a human eye. And then design the algorithm that makes the robot to detect the lane. Our course consisted of one lane with a solid white line on the right. Only the white line was detected and used as a tracking reference. Since the reference was detected on the right of the vehicle, the vehicle needed to keep a distance far from the reference on the left. An offset, equivalent to 120 pixels, was added to move the reference to the left and keep the vehicle in the middle of the boundary. Another situation was considered in which the two types of the lanes were switched. Now the solid white line was detected on the left of the vehicle and the orange cones were obstacles. The offset of 120 pixels was added to move the reference to the right and keep the vehicle in the middle of the boundary. The tracking performance in the two similar courses was almost perfect except that vibrations of a camera due to the rough surface. The vibrations caused the vehicle to move a little side to side when driving and it was still under control. However, this can be solved by slowing down the vehicle. The basic principle of the algorithm is shown in figure 4.

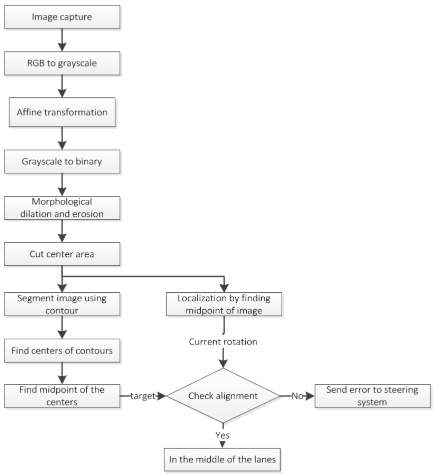


Figure 4. Lane detection algorithm flow chart

C. Obsacle avoidance

According to brand new LiDAR technology, this innovation allow robot to measure the distant between object to itself with 100% accuracy in 360 degrees direction. Corgi Robot has LiDAR lite v3 sensor module to detect the obstacle. The distance in 360 degrees is obtained from a LiDAR-Lite connected to Arduino microcontroller. A small dc motor is used to drive the LiDAR sensor module holder to rotate 360 degrees. A group of data is obtained and collect in a sequential from of data called array. And then the array is transform into a graph shown in Fig. The data in the array will be refreshed and ready to receive new group of data. The LiDAR sensor module is installed on the front of the robot to let the sensor scan for object in front of it in cone direction. The diagram of sensor module position and direction of detection are shown in figure 5 and figure 6 respectively.

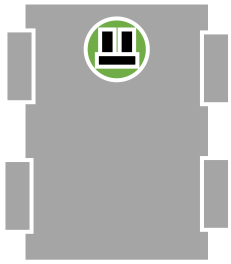


Figure 5. LiDAR Lite V3 position diagram

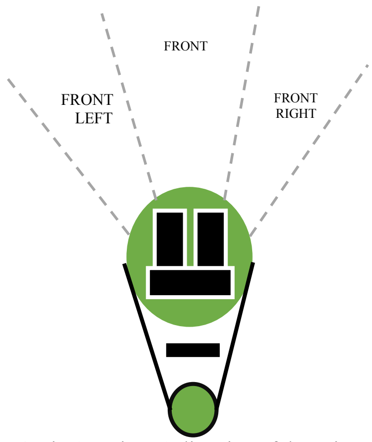


Figure 6. LiDAR Lite V3 direction of detection diagram

V. DRIVING SYSTEM IN VEHICLE

A. Speed control

Corgi's speed depends on the angle of the servo motor which obtained from PID controller explained in next below topic which has range of center error between 0 and 50 degrees. 0 degrees error is represented that the car is going straight this case will use the maximum speed to drive but in case of turning left or right immediately will consider the minimum speed to drive the car for avoiding the understeering situation. For the deep detail, the technique of speed control of Corgi car uses the linear equation to control, by using range of max and min angle (90-130 degree) relate with max and min speed (190-255 m/s).

B. Steering control

In a typical control system (Figure 7), the process variable or output is the system parameter that needs to be controlled. The process variable is the servo motor angle of the car which need to be controlled to reach the right position (go along the yellow lane). The sensor is used to measure the process variable and provide feedback to the control system. The oCam-1CGN-U is the sensor that detect car's position. Set point is the desired or command value for the process variable and it also is the angle position which lead to be along the yellow lane, the value is come from the python's command. Without the PID controller, the car can run with good stability in only low speed so when we increase its speed, it is instability, so the car cannot reach its goal directly which cause of error. So we need to improve this system by using PID controller to get rid of that error.

First thing first, the characteristic of driving is in the mathematic form. It can collect the value of input and output which we keep from the car running and calculate it in MATLAB.

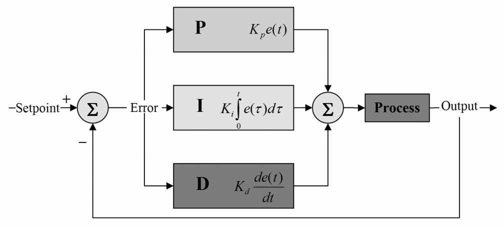


Figure 7. The system which include PID controller

The Corgi car's model is

$$G(s) = \frac{0.9947}{0.016s^2 + 1.078s + 1}$$

From the Ziegler-Nichols Tuning (second method)

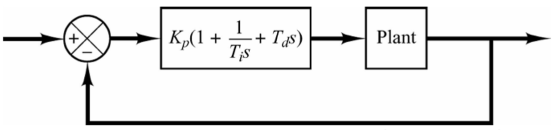
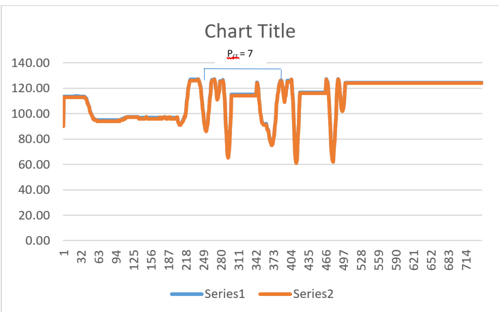


Figure 8. Ziegler-Nichols PID tuning model

To find each parameter, parameters Kp is Kp, Ki is Kp/Ti and Kd is Kp*Td. First, the Kp is less number and then increase it until the car had steady-state oscillation note this gain as Kcr (critical gain), then plot graph of output (Figure 9) from the car to find Pcr (critical period).
At Kcr = 1.17 the steady-state oscillation occurs.



P_{cr} = 7 sec.

Figure 9. The graph show the servo output of the car

Calculate all number by using the Ziegler-Nichols gain estimator chart (Figure 10).

- Gain estimator chart

Type of Controller	K_p	T_i	T_d
P	$0.5K_{cr}$	∞	0
PI	$0.45K_{cr}$	$\frac{1}{1.2}P_{cr}$	0
PID	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

Figure 10. The Ziegler-Nichols gain estimator chart

Results Kp = 0.7, Ki = 0.2 and Kd = 0.6125

Finally, set all values into code to run the car. But there is some little error which need to be tuned by manual method again, so fixing it by adjust the gain until the car has no error.

C. Wireless E-stop control

To stop the vehicle remotely, we use an EMERGENCY STOP button on web server (192.168.4.1) shown in figure 11 and receiver to shut down the vehicle. By use Node MCU to make web server. The webpage created by \HTML and PHP language in Arduino and communicate by TCP/IP. And another function is WIFI disconnect from web server the car will stop which the safety is ON.

Figure 11. Web server hosted by NodeMCU

VI. Experimental Result

Some experiments have been carried out at King Monkut's University of Technology Thonburi (KMUTT) to demonstrate the vehicle performance.

A. Indoor race

To demonstrate the performance of the racing Corgi robot, we first started our experiment on a winding indoor course shown in Fig. 6. In this experiment, the vehicle was required to track the reference line long the course.

Figure 12, winding indoor course. The course surface is polished stone where the light reflection has high influence on image processing. The course contains slopes up to 5° approximately. A solid yellow line represented as the tracking reference that the vehicle had to track on. We used the lane detection technique described above to detect the reference. Due to the influence from reflections, the tracking performance was acceptable with a speed of 20% PWM but it got poorer when increasing the speed.

Figure 13 shows image processing detecting yellow line from captured image.

Figure 12. Corgi running on indoor circuit

Figure 13. image processing tracking the yellow line

B. Outdoor race

We have performed on the outdoor courses as follows:

1) Green light detection

To start the racing Corgi robot, we used the traffic light as shown in figure 14, the same model as in the competition, to turn on the speed. Fig. Red light and green light are captured by the camera shown in figure 15 mounted with a pole on the rear of the vehicle. Even though it was a cloudy day, the experiment was still successfully performed since the vehicle start running when detecting the green light. Figure 16 shows image processing detecting the green light from captured image.

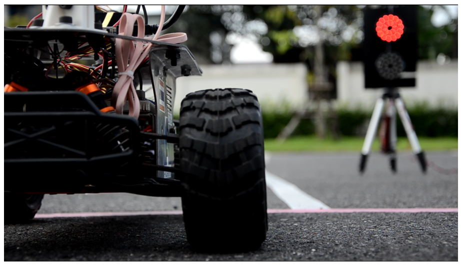


Figure 14. Traffic light placed on the right of the lane

Figure 15. Camera for image processing mounted on the pole

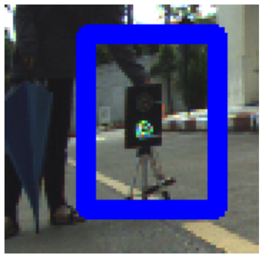


Figure 16. image processing detecting the green light

2) Single lane detection

According to figure 17, our course consisted of one lane represented by a solid white line on the right. Only the white line was detected and used as a tracking reference. Since the reference was detected on the right of the vehicle, the vehicle needed to keep a distance far from the reference on the left. An offset, equivalent to 40 pixels, was added to move the reference to the left and keep the vehicle in the middle of the boundary. Another situation was considered in which the two types of the lanes were switched. Now the solid white line was detected on the left of the vehicle and the orange cones were obstacles. The offset of 40 pixels was added to move the reference to the right and keep the vehicle in the middle of the boundary.

The tracking performance in the two similar courses were almost perfect except that vibrations of camera due to the rough surface. The vibrations caused the vehicle to move a little side to side when driving and it was still under control. However, this can be solved by slowing down the vehicle.

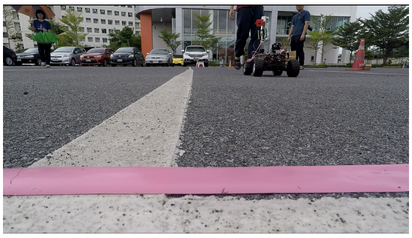


Figure 17. Solid white line tracking

VII. CONCLUSION

The design of autonomous racing car including intelligent object detection and object avoidance techniques is presented in this work. The novelty of this work emphasizes on utilizing only a camera and a distance sensor to control the vehicle. Image processing techniques are presented to smartly steer the vehicle. Only an ultrasonic sensor is used in the design to control speed of the vehicle. The experiments have shown that the vehicle can successfully track the lanes and autonomously avoid obstacles in different situations. Different strategies have been proposed to drive the vehicle in different race courses.

ACKNOWLEDGMENT

We would like to express our deep gratitude for the financial support from King Mongkut's University of Technology Thonburi, and Dynamic Spring Co.,Ltd for supplying some parts to build the robot. We are most grateful indeed for this wonderful opportunity.

REFERENCES

[1] D. Fernandez-Llorca, R. Quintero Minguez, I. Parra Alonso, C. Fernandez Lopez, I. Garcia Daza, M. A. Sotelo, and C. A. Cordero, "Assistive Intelligent Transportation Systems: The Need for User Localization and Anonymous Disability Identification," IEEE Intelligent Transportation Systems Magazine, vol. 9, no. 2, pp. 25–40, 2017.

[2] K. Dolui, S. Mukherjee, and S. K. Datta, "Traffic status monitoring using smart devices," 2013 International Conference on Intelligent Interactive Systems and Assistive Technologies, Aug. 2013.

[3] B. Panomruttanarug, "Application of Iterative Learning Control in Tracking a Dubins Path in Parallel Parking," International Journal of Automotive Technology, to appear

[4] L. Wei, C. Cappelle, and Y. Ruichek, "Unscented information filter based multi-sensor data fusion using stereo camera, laser range finder and GPS receiver for vehicle localization," 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), Oct. 2011.

[5] https://avc.sparkfun.com/

[6] http://archive.darpa.mil/grandchallenge/

[7] http://www.igvc.org/

[8] https://robotracing.wordpress.com/