Implementation of direction control on self-driving car prototype

N E Fauziah1,2, I P D Wibawa1, and C Ekaputri1
1Electrical Engineering, School of Electrical Engineering, Telkom University, Bandung, Indonesia

E-mail: nisaevifauziah@gmail.com

Abstract. A Self-driving car needs sturdy steering system. Therefore designing direction control must be precise to the track. This direction control will help the car know the current position and direct it to the right angle. Directional control results in a turning angle. Turning angle is obtained from the tangent value between two Region of Interest (ROI) in the frame captured. The turning angle obtained is conditioned to match the angle of the servo motor as the front actuator of the car prototype. The results of this research are the turning angles that are read in the Python program on a straight lane have an average of 90.4198 with an average error of 1.086, the right turning lane has an average turning angle of 99.5502, 112.96973 and 117.0711 with an average error 3.03727, 3.62493 and 3.0636296, the left turn 58.7540333, 71.218 and 80.1277667 with an average error of 1.61674, 1.88093 and 1.48696 so that they can direct the car prototype in accordance with the sharpness of the bend in the lane.

1. Introduction
A self-driving car prototype is a prototype vehicle that can move unmanned in its environment. To reach the goal, the road-based image processing is needed. Several research groups have demonstrated methods for knowing the turning angle of the car prototype and have obtained impressive results on this task [1-6] and kinematic modelling for car mathematical approaches [1-3].

Based on the explanation presented in the paper and journal above, this research discusses more the direction control design that is designed more simply. Direction control is designed to determine the direction of motion of a car prototype. The results of the direction control is the turning angle of a car prototype towards the trajectory by the image processing method.

2. Direction Control
When the self-driving car prototype runs as far as x cm it will appear the lane of the car prototype dynamically. The car prototype will run along the lane and turn on the bend at the turn angle obtained from the two Region of Interest (ROI) on the frame captured. Determination of the behavior of the car prototype when meeting the right-turn lane is to find out the horizontal and vertical position of the actual car (x, y) and the position of the midpoint of the road leading to the cornering lane.

2.1. Turn angle
Based on the kinematic model Ackerman-like vehicle concept in controlling the direction of a car prototype motion requires three coordinate frames as a parameter determining the turning angle,
inertial coordinates, lane coordinates, and car prototype coordinates can be seen in Figure 1. Motion direction control is done by comparing the position of origin \((x, y)\) and the actual position of the car prototype \((x, y)\). The projection of the car coordinates with the lane coordinates produces an orientation of \(\theta\) as depicted in Figure 2. The formula to get the value of theta can be seen in equation 1. The range of \(\theta\) values obtained is between -90 to 90 degrees which is then reduced by the elbow from the inertia coordinates produce a turning angle \(\theta_d\) which are further elaborated in equation.

\[
\theta = \arctan\left(\frac{x_2 - x_1}{y_2 - y_1}\right)
\]

\[
\theta_d = \theta - 90
\]

Figure 1. Schematic coordinate frame

Figure 2. Determination of turning angle based on three coordinate frames

2.2. Image processing

The 640x240 pixel frame is divided into two ROIs as rectangular ROI samples located at the bottom and centre of the frame. This sample will be used as a limitation of the black area on the trajectory with restrictions on the area that can be known as the edge of the trajectory being passed.

After getting the ROI from the frame then the segmentation method is applied to the frame to divide the image into several parts based on colour. Segmentation in this research is used to find out the edge of the lane that is passed. The first step in this segmentation is to determine the HSV value that is right for the black section of the lane. The HSV value is set to make parts other than black in the unread image. This segmentation divides the captured camera sensor into two parts, the black section of the lane, road marking and the surrounding environment of the lane. Setting the HSV value is done by the Trial-Error method. Then, the black part of the lane that enters the ROI part of the frame
is drawn in a closed contour to get the coordinates of the midpoint of the lane and the midpoint of the camera caught on each contour. Determining the contours of the frame can be seen in figure 3. The centre point obtained serves as a helping component in finding the turning angle of a car prototype.

![Figure 3. Determining the Turning Angle of a Car prototype](image)

The turn angle is obtained by drawing a straight line from the midpoint of the first contour at first ROI to the midpoint of the second contour at second ROI. Then, drawing a straight line from the midpoint of the camera on the first contour to the midpoint capture the camera on second contour. The tangent between the two projection lines produces an angle of $\theta$ can be seen in figure 4. The range of turning angles that are read is -30 to 30 degrees adjusting the maximum turning angle of the steering wheel by 30 degrees to prevent slip on the wheels. The front actuator as steering wheel used is a 180 degree servo motor so that the turning angle value of the prototype car is around 60 to 120 degrees. Flowchart of the system can be seen in figure 7.

![Figure 4. Determining the turning angle on the lane](image)

3. **Implementation and Result**

The control law for the self-driving car described in this paper was tested on the 1:10 scaled vehicle, as depicted in figure 5. The vehicle was modified to allow for an automatic turn at a maximum angle of 120 degrees and a minimum angle 60 degrees. The car prototype is equipped with a web camera to provide three coordinate frames of the vehicle with regard to the centre of the lane, a determined lane as a provider of vision-based lane tracking task to obtained turning angle. The image processing runs using a personal computer on python programs.

![Figure 5. Car prototype design](image)

Practical experiments were conducted on a private circuit as depicted in figure 6. The circuit is composed of several cornering sections with obstacles added. As can be observed, the control objective is to achieve the right direction of the car with the turning angle as the input of the actuator.
Various practical trials were conducted so as to test the validity of the control law for different initial conditions in real circumstances. During the tests, the reference camera sensor of the car prototype is set to suitably with the lane as depicted in figure 8.
In the experiments, a control system was used to autonomously guide the vehicle based on visual measurements. This test is done by calculating the angles that are formed from the midpoints of ROI1 and ROI2 on the straight, right-turn, and left-turn read in the Python program, then compared with the lane angle of 60, 70, and 80 degrees for left-turn lane, 90 degrees for straight lane, 100, 110, and 120 degrees for right lane that have been made before. Car prototypes are placed on the test track and then lifted and then put back onwards 30 times.

![Figure 8. The placement of a camera sensor on a car prototype](image)

**Figure 8.** The placement of a camera sensor on a car prototype

![Figure 9. Test result on a 90-degree trajectory](image)

**Figure 9.** Test result on a 90-degree trajectory

(a) Testing on a 60 degree trajectory
(b) Testing on a 70 degree trajectory
(c) Testing on a 80 degree trajectory

![Figure 10. Test result on a left-turn trajectory](image)

**Figure 10.** Test result on a left-turn trajectory (a) 60-degree, (b) 70-degree, (c) 80-degree
Fig. 9-11 show measurements result of the turning angle of the lane, the average angle read on the 90 degrees lane is 90.4198 degrees and the average error is 1.086 degrees, the average angles that are read on 100, 110, and 120 degrees lane are 99.5502, 112.96973 and 117.0711 degrees respectively with an average error of 3.03727, 3.62493 and 3.0636296 degrees, and the average angles that are read on lanes 60, 70, and 80 degrees are 58.7540333, 71.218 and 80.1277667 degrees respectively - the average errors of 1.61674, 1.88093 and 1.48696 degrees.

Based on the graph in Fig. 9-11, the largest average error value on the left turn lane and the right turn lane is 70 degrees and 110 degrees. This is caused by the exact position of the contour at the confluence of the two lanes so that the right and left sides of the contour at ROI 1 are not in the same lane that is the 60 degree and 70 degree paths so the midpoint of the contour that can be completely not from the 70 degree path also occurs at the 110 degrees lane. Based on the three test tables on each left turn trajectory, the value of the turn angle that is read is unstable but remains close to the actual value, this is caused by the factor of putting the robot back after being tested for 30 times. In this test range, the possibility of a car prototype is not in the middle of the position of the car is not perpendicular to the track.

![Testing on a 100 degree trajectory](image1)
![Testing on a 110 degree trajectory](image2)
![Testing on a 120 degree trajectory](image3)

(a) (b) (c)

Figure 11. Test result on a right-turn trajectory (a) 100-degree, (b) 110-degree, (c) 120-degree

4. Conclusion

The direction control design allows the measured turning angle range to be 60 degrees to 120 degrees. Based on the turning angle test to the straight lane, right turn lane, and the left-turn lane the average angle read on the 90 degrees lane is 90.4198 degrees and the average error is 1.086 degrees. The average angles that are read on 100, 110, and 120 degrees lane are 99.5502, 112.96973 and 117.0711 degrees respectively with an average error of 3.03727, 3.62493 and 3.0636296 degrees. The average angles that are read on lanes 60, 70 and 80 degrees are 58.7540333, 71.218 and 80.1277667 degrees respectively - the average errors of 1.61674, 1.88093 and 1.48696 degrees. Future work on this research is the design of directional control can be added algorithms to get a car bow that when car prototypes meet the bend lines which are 9 cm in front of him did not immediately turn so that the mobile robot has a gap in turn.
Acknowledgements
This research was funded by the Telkom University for research collaboration with Hanbat National University and Research Institute for Tea and Cinchona.

References
[1] I. Kecskés, Z. Balogh, and P. Odry, “Modeling and fuzzy control of a four-wheeled mobile robot,” 2012 IEEE 10th Jubil. Int. Symp. Intell. Syst. Informatics, SISY 2012, pp. 205–210, 2012.
[2] M. A. Sotelo, “Lateral control strategy for autonomous steering of Ackerman-like vehicles,” Rob. Auton. Syst., vol. 45, no. 3–4, pp. 223–233, 2003.
[3] M. Werling, L. Gröll, and G. Bretthauer, “Invariant trajectory tracking with a full-size autonomous road vehicle,” IEEE Trans. Robot., vol. 26, no. 4, pp. 758–765, 2010.
[4] D. Kuan, G. Phipps, and C. Hsueh, “Autonomous Robotic Vehicle Road Following,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 10, no. 5, pp. 648–658, 1988.
[5] A. A. M. Assidiq, O. O. Khalifa, M. R. Islam, and S. Khan, “Real time lane detection for autonomous vehicles,” Proc. Int. Conf. Comput. Commun. Eng. 2008, ICCCE08 Glob. Links Hum. Dev., pp. 82–88, 2008.
[6] R. Reyes and R. Murrieta-Cid, “An approach integrating planning and image-based visual servo control for road following and moving obstacles avoidance,” Int. J. Control, vol. 0, no. 0, pp. 1–15, 2019.