

An Autonomous Vehicle Driving Control System*

KHALID BIN ISA

Department of Computer Engineering, Faculty of Electric and Electronic Engineering, University College of Technology Tun Hussein Onn, 86400 Parit Raja, Batu Pahat, Johor, Malaysia

E-mail: khalid_hj_isa@yahoo.com

ADZANAN BIN JANTAN

Department of Communication and Computer Systems Engineering, Faculty of Engineering, University Putra Malaysia, 43400 UPM, Serdang, Selangor, Malaysia

An autonomous vehicle driving control system carries a large number of benefits, whether for the engineering industry or engineering education. The system discussed here provides the measurements obtained from vision—namely, offset from the centerline at some look-ahead distance and the angle between the road tangent and the orientation of the vehicle at some look-ahead distance—and these are directly used for control. This paper also presents simulation results regarding autonomous vehicle dynamics and control, along with methods, techniques and approaches for developing this system which can be used by engineering educators for automotive, robotics or image processing courses.

INTRODUCTION

AUTOMATION OF the driving task has been the subject of much research recently, but to design and simulate an autonomous vehicle driving control system is not easy, because it is such a complex system. Over the last decades, many engineers and educators had problems in estimating the results of an autonomous vehicle driving control system, but, in recent years, computers have been used to great effect in simulating such systems. Therefore, the main objective of the system discussed in this paper is to demonstrate an autonomous vehicle driving control system to engineers, students and researchers by means of a simulated system.

Another objective of this system is to prove that the steering command for the vehicle lateral control can be determined by processing and analyzing images taken whilst driving a vehicle. Then, by combining the steering command and other vehicle dynamics parameters, the vehicle's dynamics performance can be determined too.

SYSTEM DESIGN

The system design for an autonomous vehicle is dependent on a number of tasks that can be performed by the vehicle. Since the experiment only uses one video camera as a sensor, this paper can only present the lane detection task, along with the dynamics and control of the vehicle.

Fig. 1 presents the general flow of the system. The system consists of four subsystems: the sensor (video camera), image processing, the controller and the vehicle.

THE SENSOR

The sensor is the key element of an autonomous vehicle system, because it provides the information about a driving scenario. The system discussed in this paper uses a single video camera as a sensor. To get the input data from the image, the video image sequences must be captured. The input data of this system is provided by colour image sequences taken from a moving vehicle. A single colour video camera is mounted inside the vehicle behind the windshield along the central line. This records the images of the environment in front of the vehicle, including the road, the vehicles on the road, traffic signs on the roadside and, sometimes, incidents on the road. The video camera saves the

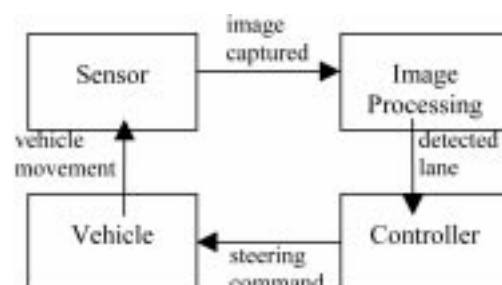


Fig. 1. The four subsystems of a vision-based autonomous vehicle driving control system.

* Accepted 30 May 2005.

video images in AVI file format, then the video file is transferred to the computer. The images are captured at a rate of up to 30 frames/second, but, since this system uses MATLAB as the system development, so the rate of image capture is only 15 frames/second. The image processing subsystem takes an image from the memory and starts processing it in order to detect the desired lane.

IMAGE PROCESSING AND ANALYSIS FOR PREDICTING AND DETECTING THE VEHICLE LANE

The goal of the image processing is to extract information about the position of the vehicle with respect to the road from the video image. Two major processes are implemented: the pre-processing process and then the lane detection process. The goal of pre-processing is to remove image noise and make the images sharper. The goal of lane detection is to detect the desired lane of the vehicle in order to obtain the look-ahead distance and the lane angle. This process is based on the real-time data of video sequences taken from a vehicle driving on the road.

Many lane detection algorithms have been developed over the years, and several detection systems have been reviewed. The LOIS [1] system used a deformable template approach to find the best fit of road model, whether straight or curved. The research groups at the University Der Bundeswehr [2] and Daimler-Benz [3] base their road detection functionality on a specific road model: lane markings are modelled as clothoids. This model has the advantage that knowledge of only two parameters allows full localization of lane markings and computation of other parameters, like the lateral offset within the lane, the lateral speed with respect to the lane and the steering angle. The approach in [4] is an evolutionary approach to lane markings detection, using collaborative autonomous agents to identify the lane markings in road images.

The lane detection process used by this system handles road area segmentation and shadow removal using colour cues. It then finds the best linear fit to the left and right lane markers over a certain look-ahead range using a variant of Hough Transform. From these measurements, the lateral position and orientation of the vehicle with respect to the roadway at a particular look-ahead distance can be computed and estimated. This detection can be applied for both painted and unpainted roads. Fig. 2 presents the four processing steps of the lane detection algorithm: image segmentation, edge detection, Hough Transform, and lane tracking.

Regions of interest measurement and features extraction using colour cue segmentation

Segmentation of the images is crucial for the analysis of the driving environment. Many traditional systems have been proposed for recovering

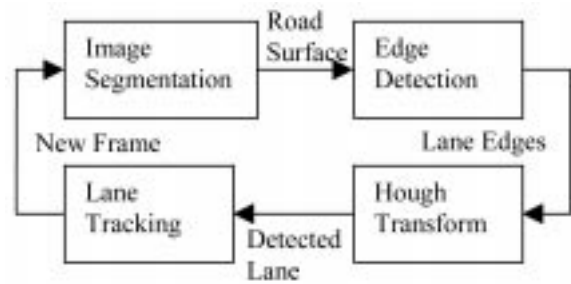


Fig. 2. Image processing steps of the lane detection algorithm.

parts of the traffic scenes by extracting geometric features in order to infer and verify the existence of certain categories of objects. Another relevant method of segmentation is based on region growing and clustering. The effectiveness of such a method crucially depends upon the capability of measuring similarity, such as in texture and pixel colour information. The method described in this paper used a colour cue to measure segmentation. It is shown as a colour-based visual module providing relevant information for localization of the visible road area, independently of the presence of lane boundary markings and in different lighting conditions.

In the road image, the road area had characteristics such as the following: 1) most of the lower part of the image was considered as the road area, and 2) road areas have a quasi-uniform colour, resulting from the fact that the road area is generally a grey surface in a more coloured environment. Although the absolute surface colour can provide useful cues for this task, the response of the colour imaging device is mediated by the colour of the surfaces observed, by the colour of light illuminating them, and by the setting of the image acquisition system. To have better control over variations in pixel values for the same colour, and to remove the shadows, the RGB colour space must be converted to the HSV (hue, saturation and value) colour space.



Fig. 3. Original image of frame one in the RGB colour space.



Fig. 4. Intensity (V) image in the HSV colour space.



Fig. 5a. Road surface as the object/region of interest. Other objects/background are converted to black (0).

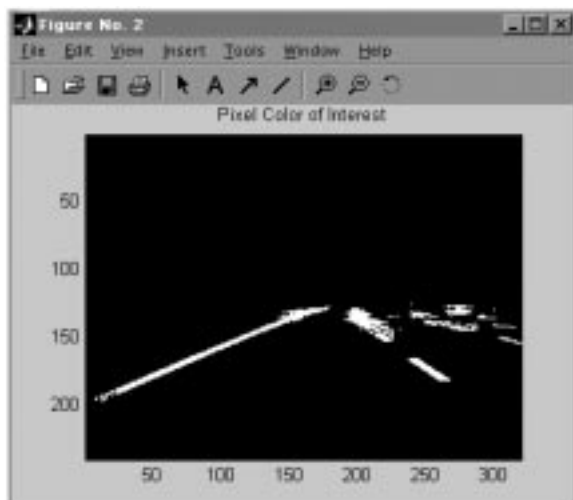


Fig. 5b. Lane marking extraction based on the colour of pixels of interest.

After converting the RGB image to an HSV image, we specified the coordinates of regions of interest in the image. Fig. 3 presents the original image and Fig. 4 shows the image after the colour conversion process. Fig. 5 shows the image by image segmentation process by computing the coordinates of regions of interest and determining the pixel colour of interest.

Extraction and detection of vehicle lane edges using the Canny edge detector

The purpose of the edge detection process is to extract the image edges using an edge detection operator or an edge detector. The operator will locate the position of pixels where significant pixels exist. The edges are represented as white and non-edges will be black. Fig. 6 shows the lane edges of the image. Edges in images are areas with strong intensity contrasts. Edge detecting of an image significantly reduces the amount of data and filters out useless information while preserving the important structural properties of the image. Edge detection is the most common approach for detecting meaningful discontinuities in the grey level. Intuitively, an edge is a set of connected pixels that lie on the boundary between two regions.

This system used a Canny edge detector to locate the position of pixels where significant edges exist. By applying the Canny edge detector to a road image, two images that denote the edge pixels and the orientation of gradient can be obtained. The Canny edge algorithm is known as the optimal edge detector. This detector is an enhancement of the many existing edge detectors. Several constituents of this detector have improved current methods of edge detection. The first and most obvious is the low error rate. It is important that edges occurring in an image should not be missed and that there is no response to non-edges. The second constituent is that the edge points are well localized. In other words, the distance between the edge pixels as detected by the detector and the actual edge is to be minimal. A third is that there is only one response to a single edge. This was implemented because the first two constituents were not substantial enough to eliminate the possibility of multiple responses to an edge.

The Canny's basis for good detection is its low probability of not marking real edge points and of falsely marking non-edge points. This is achieved by using the following equation:

$$SNR = \frac{|\int_{-w}^w G(-x)f(x)dx|}{n_0 \sqrt{\int_{-w}^w f^2(x)dx}} \quad (1)$$

f is the filter, G is the edge signal, and the denominator is the root-mean-squared (RMS) response to noise $n(x)$ only. Canny's good localization criterion is also close to the centre of the true edge. Below is the equation used to measure the localization, using the RMS distance of the marked edge from the centre of the true edge.

$$\begin{aligned}
 \text{Localization} &= \frac{1}{\sqrt{E[x_0^2]}} \\
 &= \frac{|\int G(-x)f(x)dx|}{n_0 \sqrt{\int_{-w}^w f^2(x)dx}} \quad (2)
 \end{aligned}$$

Features isolation and approximation of the vehicle lane using Hough Transform

Hough Transform was used to combine edges into lines, where a sequence of edge pixels in a line indicates that a real edge exists. By using the edge data of the road image, Hough Transform will detect the lane boundary on the image. This is because this technique detects shapes from image edges, and assumes that primitive edge detection has already been performed on an image.

This technique is most useful when detecting boundaries whose shape can be described in an analytical or tabular form. The key function of this system is to map a line detection problem into a simple peak detection problem in the space of the parameters of the line. Although there may be curves in the road geometry, a straight line will still be a good approximation of a lane, especially within the constraints of a reasonable range for vehicle safety, because the curve is normally long and smooth.

Hough Transform is a technique that can be used to isolate features of a particular shape within an image. It can be divided into two types: classical and generalized. Because it requires the desired

features to be specified in some parametric form, the classical Hough Transform is most commonly used for detection of regular curves, such as lines, circles, ellipses, etc. A generalized Hough Transform can be employed in applications where a simple analytic description of a feature is not possible.

Hough Transform works by letting each feature point (x, y) vote in (m, b) space for each possible line passing through it. These votes are totalled in an accumulator. If, for instance, a particular (m, b) has one vote, this means that there is a feature point through which this line passes. If it has two votes, it means that two feature points lie on that line. If a position (m, b) in the accumulator has n votes, this means that n feature points lie on that line. The algorithm for the Hough Transform can be expressed as follows:

1. Find all of the desired feature points in the image.
2. For each feature point:
 - For each possibility i in the accumulator that passes through the feature point, increment that position in the accumulator.
3. Find local maximum in the accumulator.
4. If desired, map each maximum in the accumulator back to the image space.

After the coordinates of the lane line in the accumulator space have been identified, we remapped the line coordinates of the lane to the image space, so that the lane can be highlighted. Fig. 7 shows the results of Hough Transform and Table 1 shows the coordinate of the lines.

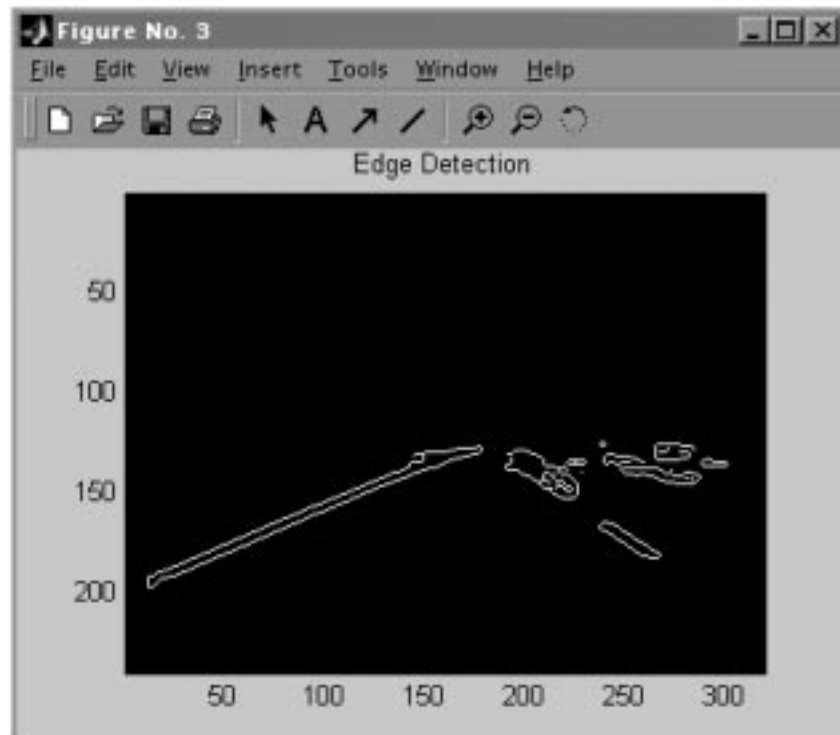


Fig. 6. Edges of lane marking and some unwanted edges.

Lane tracking

A distinction can be made between the problems of lane detection and tracking. Lane detection involves determining the location of the lane boundaries in a single image without strong prior knowledge regarding the lane position. On the other hand, lane tracking involves determining the location of the lane boundaries in a sequence of consecutive images, using information about the lane location from previous images in the sequence to constrain the probable lane detection in the current image. Some systems use different algorithms for lane detection and tracking, but this system uses the same algorithms for lane detection and tracking, which will save on processing time.

Since we can obtain up to 30 frames per second, the difference between images in the sequence will be very small. Therefore, we do not need to process each entire image in terms of lane tracking. From the previous lane position, we can have a pretty good idea of lane position in the current frame. Therefore, in the Hough transformation, the angle (θ) can be restricted by the estimated range from previous frames. This will improve the computational speed and the accuracy of detection. In each second, the first several image frames will be

processed by the lane detection algorithm, and this will provide a good estimate of lane tracking for the next frames. The lane tracking scheme described above will process the rest of the images in a few seconds.

VEHICLE MODELLING

Designing a vehicle controller such as a lateral controller requires a model of the vehicle's behavior. There have been two approaches to this modeling: dynamics and kinematics. In this system we used a dynamic model approach, because it gives a highly accurate portrayal of the vehicle's behavior and because the controllers designed with this approach are sufficiently robust with those dynamics. The dynamics of the vehicle for system simulation is a three-dimensional vehicle (car) with six degrees of freedom (6 DOF). The degrees of freedom is the number of movements of the vehicle. The vehicle has 6 DOF because it can have 3 translational (x, y, z) and 3 rotational (pitch, yaw, roll) DOF. Besides the degrees of freedom, several other factors—such as the vehicle's weight, center of gravity, cornering

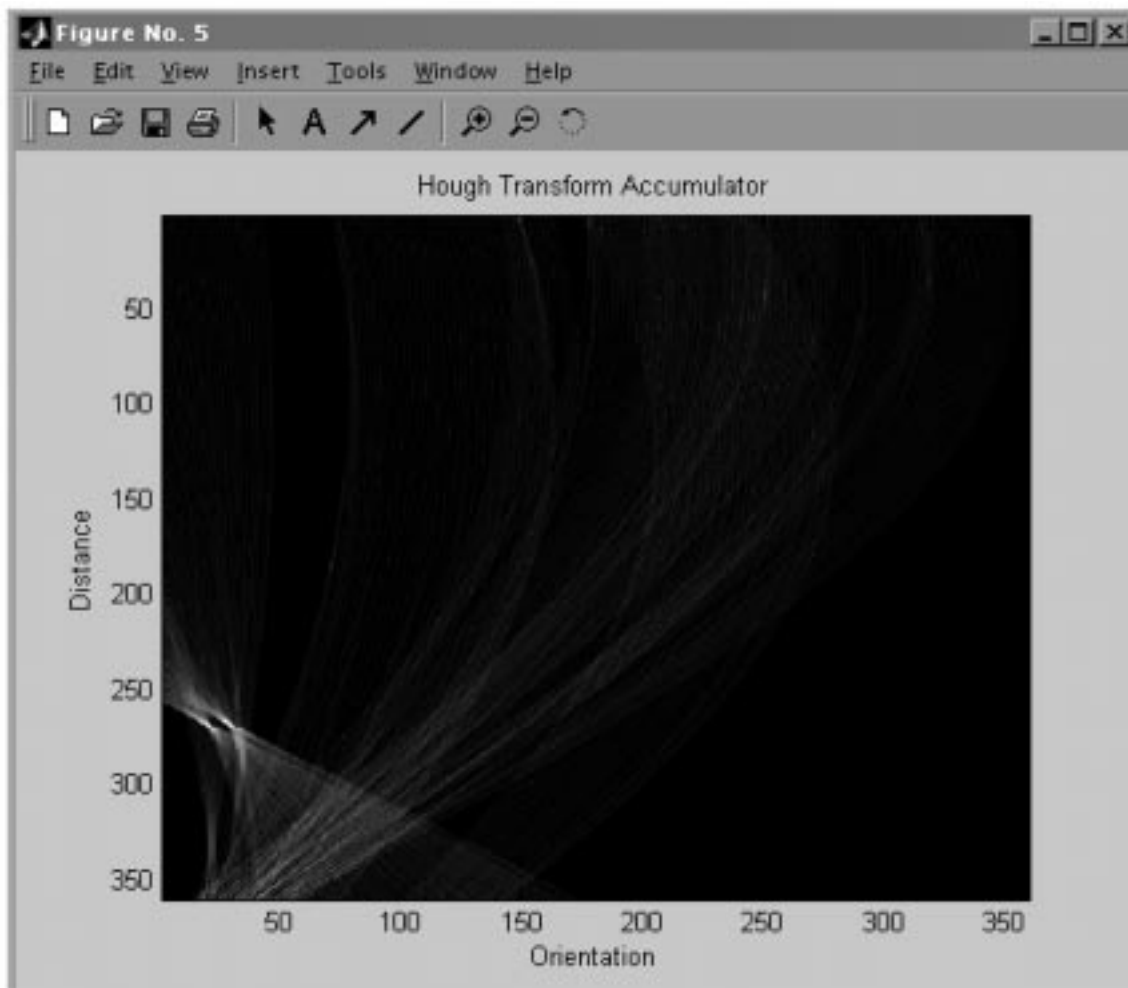


Fig. 7a. The Hough Transform accumulator estimates the line coordinates of the lane.

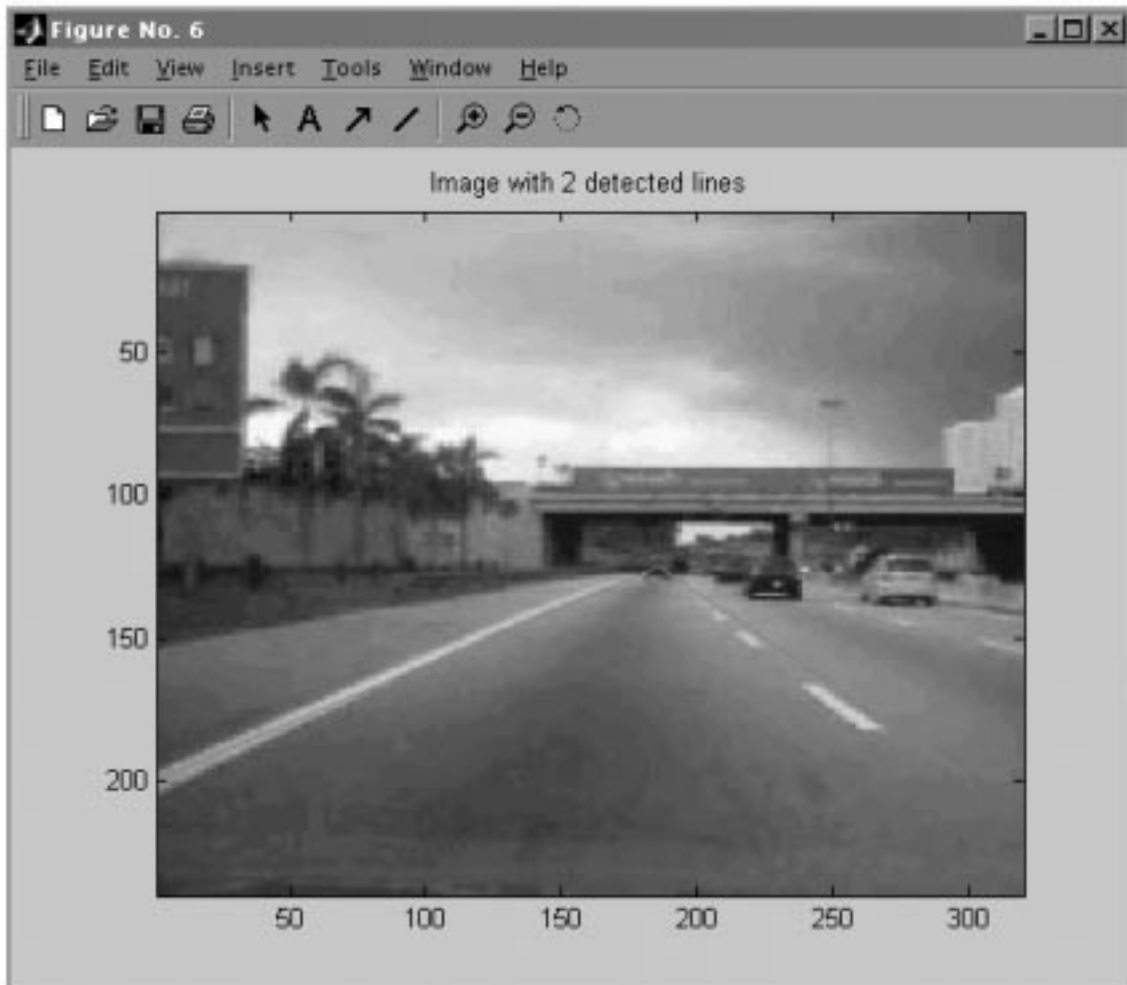


Fig. 7b. Original image with detected lane.

Table 1.

Line coordinates of the lane			
Line	x	y	z
Line1	300.2403	185.9707	0.5711
Line2	199.0000	125.0087	199.4778

stiffness, wheel slippage, and others—were taken into account in dynamic modeling.

Since this is a simulation-based system, the vehicle dynamics are strictly governed by the theoretical differential equations and the parameters used in those equations. The resulting controllers are, in general, solved in terms of these parameters. Fig. 8 shows the notations or parameters and physical system of a four-wheel vehicle. We consider the car to be a rigid mass, with a coordinate system centered on the center of mass of the car.

The meaning of each term is as follows:

- a Distance from centre of gravity (CG) to front axle

- b Distance from centre of gravity (CG) to rear axle
- d Distance from car centerline to each wheel
- X, Y Earth-fixed coordinates
- U Vehicle's longitudinal velocity
- V Vehicle's lateral velocity

Some parameters for the four-wheel vehicle used in our experiments were:

- G = 32.2 m/sec² Gravity
- w = 1590.0 kg Weight of car in kilograms
- m = w/G Mass of car
- a = 1.22 m Front axle to CG
- b = 1.63 m Rear axle to CG
- C_{af} = 3646.1 kg/rad Cornering stiffness for front axle
- C_{ar} = 4557.6 kg/rad Cornering stiffness for rear axle

The vehicle parameters and the outputs from the image processing process were used for the vehicle controller. Then we computed the vehicle dynamics performance. Therefore, the system simulation shows the vehicle's dynamics in the

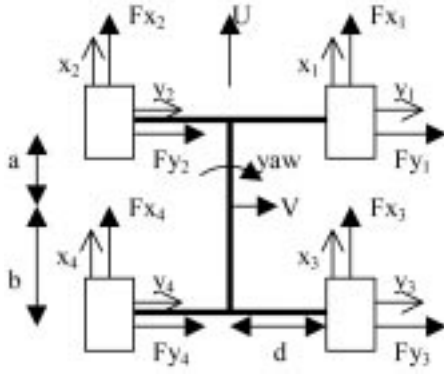
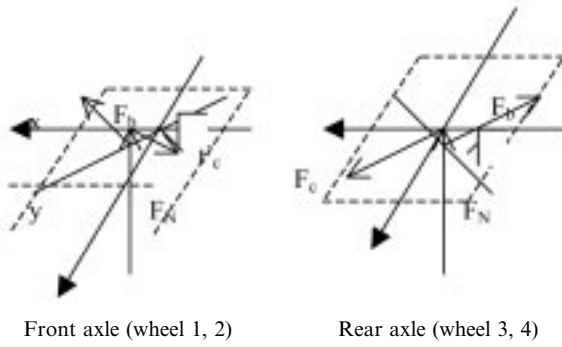


Fig. 8. The definition of the basic vehicle dimensions.


 Fig. 9. The forces acting on the tires are normal, F_N , lateral forces, F_c and brake forces, F_b .

longitudinal (acceleration and braking modes) and handling (lateral, yaw and roll modes) elements. ‘Handling’ evaluates the dynamics of the vehicle (car) in response to the driver’s steering input. ‘Longitudinal’ evaluates the acceleration and braking responses of the vehicle. Other important factors in determining vehicle dynamics are tire model and vehicle forces.

Tire model

In this system, the tire model is based on the mathematical equations of a magic tire formula.

$$F_y = F_{y0}(\alpha, \gamma, F_z) \quad (3)$$

$$\alpha_y = \alpha + S_{Hy} \text{ Where } S_{Hy} \text{ is horizontal shift} \quad (4)$$

The coefficients in the formula are calculated from parameters measured on a physical tire, which makes the model realistic. The magic tire model is an empirical model developed by Pacejka and colleagues at Delft University that has been verified experimentally to be fairly accurate. It makes it possible to determine lateral as well as longitudinal forces (brakes) and the aligning moment. Lateral force is a function of the variables slip angle (α), the normal force acting on the tire, F_z , and the camber angle (γ). On the other hand, the brake force, or longitudinal force, is the opposite of lateral force.

A common way of modeling tires is to neglect the influence of horizontal and vertical shift.

Consequently, all the parameters used for calculating the shift are set to zero. A tire model that includes vertical and or horizontal shift experiences a static lateral force. Ignoring the shifts will thus make the static force disappear. Since it is not possible to vary the camber angle in the simulation model, it is also natural to ignore the influence of that variable. Removal of the camber parameter simplifies the formula even more. The lateral force ends up being a function of α . The slip angle was calculated by means of the steering angle δ_i and the local velocity and rotation vector.

$$\alpha_i = \delta_i \arctan\left(\frac{v_{yi}}{v_{xi}}\right) \text{ Where } i \text{ for } 1 \text{ and } 2 \quad (5)$$

$$\alpha_i = -\delta_i \arctan\left(\frac{v_{yi}}{v_{xi}}\right) \text{ Where } i \text{ for } 3 \text{ and } 4 \quad (6)$$

Vehicle forces

The force equations are derived in the local coordinate system and the force acting on the model is thus projected onto this coordinate system. Fig. 9 shows the forces on the front and rear tires.

In this system, the vehicle experiences normal forces, lateral forces and brake forces. The sum of the projected forces, F_{xi} , F_{yi} , F_{zi} (i for number of tires), in the local coordinate system leads to the following equations:

$$F_{xi} = -F_{ci} \sin(\delta_i) \cos(\theta) - F_{Ni} \sin(\theta) \quad (7)$$

$$F_{yi} = F_{ci} \cos(\delta_i) \cos(\phi) + F_{Ni} \sin(\phi) \quad (8)$$

$$F_{zi} = F_{N1} \cos(\theta) \cos(\theta) - (-F_{c1} \sin(\delta_i) \sin(\theta) - F_{ci} \cos(\delta_i) \sin(\theta)) \quad (9)$$

Since the vehicle is moving, the use of Newton’s second law of motion gives the following equations:

$$\sum F_x = F_{x1} + F_{x2} + F_{x3} + F_{x4} = m\dot{v}_x \quad (10)$$

$$\sum F_y = F_{y1} + F_{y2} + F_{y3} + F_{y4} = m\dot{v}_y \quad (11)$$

$$\sum F_z = F_{z1} + F_{z2} + F_{z3} + F_{z4} = m\dot{v}_z \quad (12)$$

VEHICLE CONTROLLER

The controller methodology can be divided topically in many different ways: by method control input (two-wheel steering (2WS), 4WS, direct yaw control (DYC)), by controller design method (optimal, neural network, input scaling), or by a controller implementation structure (feed-forward, feedback, or combinations). Several aspects of designing a control system for a vehicle have been examined extensively in the past, both in the physics literature [5], as well as in control theory studies. The problem of control in a dynamic setting, using measurement ahead of the vehicle, has been explored by Ozguner et al. [6], who proposed a constant control law proportional to the offset

from the centerline at a look-ahead distance. Ackerman et al. [7] proposed a linear and non-linear controller design for robust steering. Taylor et al. [8] considered the problem of controlling a motor vehicle based on the information obtained from conventional cameras mounted on board the vehicle. Ma et al. [9] looked at the problem of guiding a nonholonomic robot along a path based on visual input.

In the system discussed here, we used a feedback 2WS controller. As mentioned before, the vehicle controller requires a model of the vehicle's behavior, either a dynamics or kinematics model. Therefore, in this system the controller is based on the mathematical model of four-wheel vehicle dynamics. There are several controllers in an autonomous vehicle. This system focused on a longitudinal controller and a lateral controller.

Feedback 2WS

A feedback controller has several advantages over purely feed-forward control. Perhaps the most important advantage is that a feedback controller maintains stable vehicle characteristics during changes in driving conditions. There are several types of feedback controller, such as proportional feedback and state feedback. In this system, we used proportional feedback.

With proportional feedback, it is possible to force the vehicle, via feedback, to have neutral steering characteristics. Since the model parameters are known, the model predicts neutral steering characteristics with feedback:

$$\delta_r(t) = C \cdot U \cdot \psi \tag{13}$$

with

$$C = m/2.L.(a/C_{\alpha f} - b/C_{\alpha r}) \tag{14}$$

where U is the velocity of the vehicle, ψ is the yaw rate of the vehicle, m is the mass of the vehicle, C is the cornering stiffness of the front and rear tires, and a and b are the distance from the centre of gravity to the front and rear axles, respectively.

Proportional feedback makes it easy to implement a controller and analyze the characteristics of the system. The rate of change of the yaw angle (ψ) is measured in many Volvo cars and this makes it possible to analyze the control system. The proportional feedback controller is designed by taking the output signal ψ from the control subject, multiplying it with a suitable constant and using it as an input signal for the steering wheel angle, δ . Different values of the constant give the system different characteristics.

Lateral control

A lateral controller or steering controller is the main controller in this system. This is because the input of this controller is based on the output of the lane detection process. The lateral controller is designed to follow the desired path, but it does not determine what the desired path is. A higher-level planner is responsible for that task. The planner may

take into account data from other sensors, to avoid collisions or arrive at its ultimate destination. The lateral controller does not know or need to know such high-level information. It only needs to know the car's location with respect to the desired path.

The lateral controller determines the steering angle based on the desired lane of the road. This steering angle maintains the vehicle in the desired position on the road. With this type of vehicle control, the driver would be able to remove his hands from the steering wheel and let the vehicle steer itself. Here, the idea is that the vehicle is set to follow some desired path. Sensors on the vehicle detect the location of the desired path. The error between the desired path and the vehicle is calculated and a microcomputer acting as the controller determines how to control the steering wheel to follow the desired path.

A model that allows the lateral forces and the yawing of the vehicle to be described must be used for theoretical studies on cornering, transient steering maneuvers as well as directional stability when driving straight ahead. Body motion in interaction with the wheel suspension must be included. The non-linear behavior of the tires, the effects of the

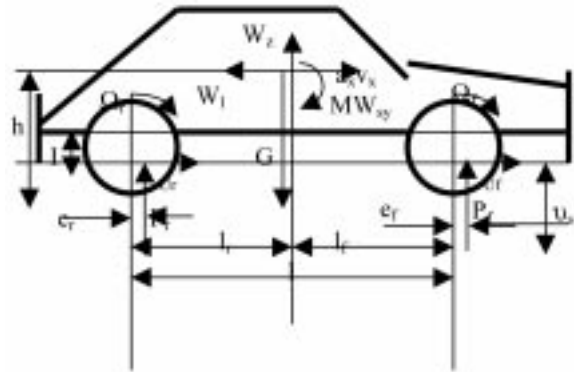


Fig. 10. Vehicle longitudinal dynamics.

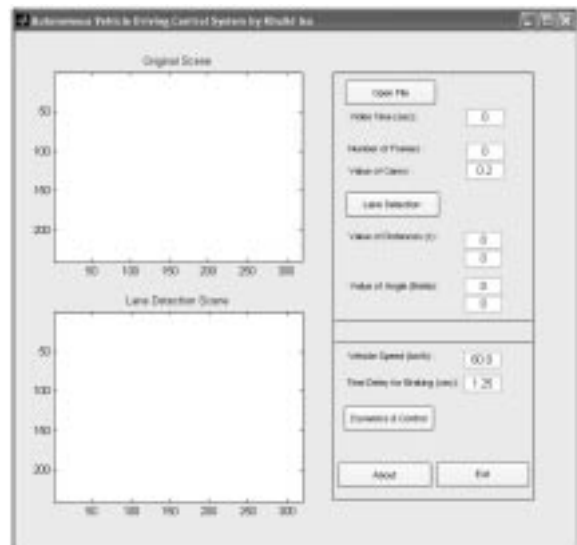


Fig. 11. Graphical user interface of the system.

drive and the steering system lead to very complex models, with the help of which an attempt can be made to simulate the vehicle's handling over a period. However, a few of the basic considerations can be shown using a linearized model.

The first step to understanding cornering is to analyze turning behavior at a low speed. At low speed, the tires need not develop lateral forces. They roll with no slip angle, and the vehicle must be able to negotiate a turn. However, at high speed the tires develop lateral forces, so the lateral acceleration is presented. In a lateral controller, the lateral force, denoted by F_c , is called the cornering force when the camber angle is zero. At a given tire load, the cornering force grows with the slip angle. At a low slip angle (5 degrees and less), the relationship is close to linear.

$$F_c = C_{\alpha}\alpha \quad (15)$$

Steady state cornering equations are derived from the application of Newton's second law. For a vehicle traveling forward with a speed of V , the sum of the forces in the lateral direction is:

$$\sum F_c = F_{cf} + F_{cr} = MV^2/2 \quad (16)$$

Also, for the vehicle to be in moment equilibrium about the centre of gravity, the sum of the moments from the front and rear lateral forces must be zero.

$$F_{cf}b - F_{cr}c = 0 \quad (17)$$

$$F_{cf} = F_{cr}c/b \quad (18)$$

substituting back

$$\begin{aligned} MV^2/R &= F_{cr}(c/b + 1) = F_{cr}(b + c)/b \\ &= F_{cr}L/b = F_{cf}L/c \end{aligned} \quad (19)$$

With the required lateral forces known, the slip angles at the front and rear wheels are established.

$$F_{cr} = Mb/L(V^2/R) \quad (20)$$

$$W_f = W.c/L \quad (21)$$

$$W_r = Mgb/L \quad (22)$$

$$\alpha_f = W_v V^2 / (C_{\alpha f} gR) \quad (23)$$

and

$$\alpha_r = W_r V^2 / (C_{\alpha r} gR) \quad (24)$$

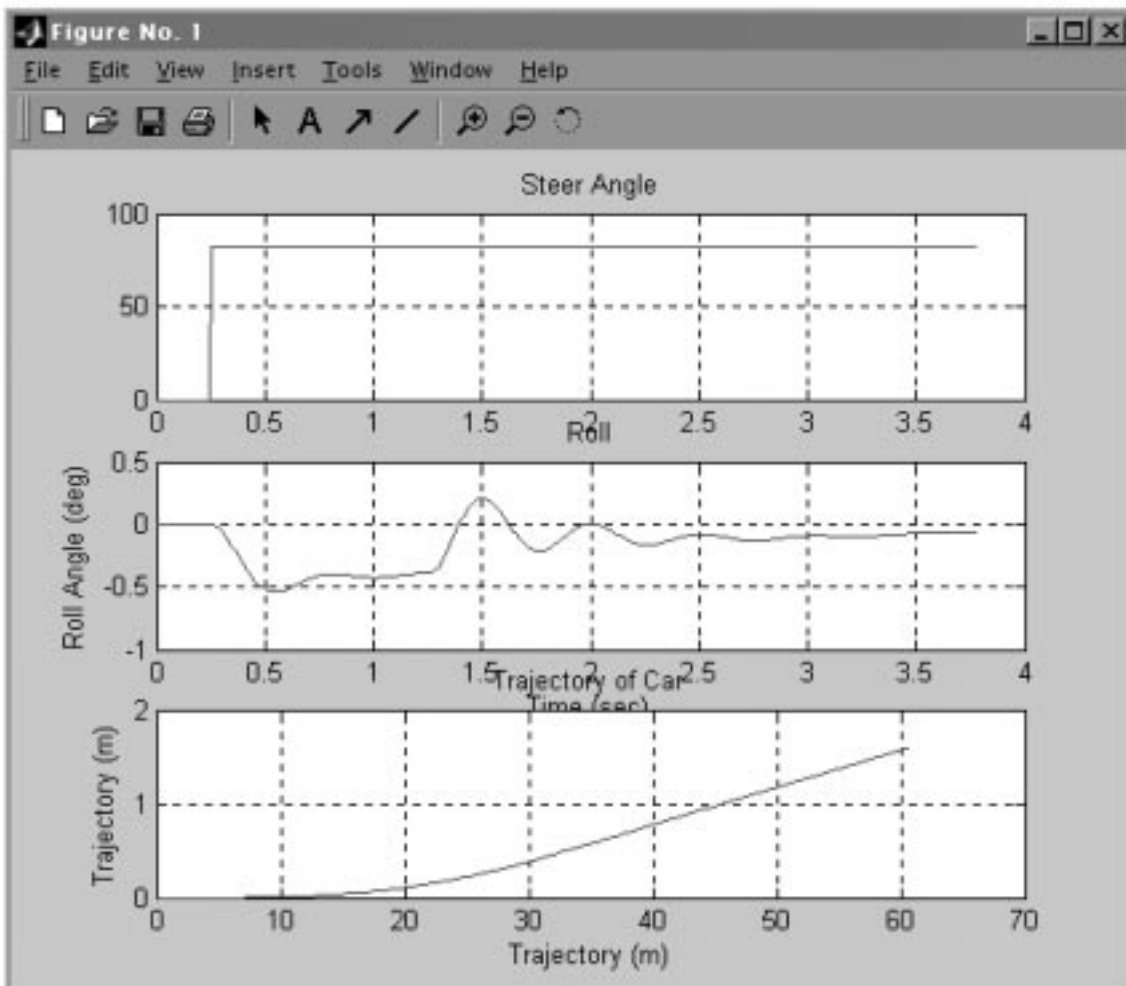


Fig. 12. Steering angle, roll angle and trajectory.

Longitudinal control

A longitudinal controller does not depend on the look-ahead distance and the angle between the road tangent and the orientation of the vehicle at some look-ahead distance, which are obtained from the lane detection process. This controller just depends on the longitudinal dynamics of the vehicle. The behavior of the vehicle when driving straight ahead or at very small lateral acceleration values is defined as the longitudinal dynamics. From the longitudinal dynamics, an evaluation of acceleration, braking and top speed can be calculated. Fig. 10 shows vehicle longitudinal dynamics. From Fig. 10, we can formulate Newton's law by using the following equations:

$$ma_x = U_r + U_f - W_l - G \cdot \sin v_s \tag{25}$$

$$0 = P_r + P_f + W_z - G \cdot \cos v_s \tag{26}$$

$$J_f \dot{\Omega}_f + J_r \dot{\Omega}_r = P_r \cdot (l_r - e_r) - P_f (l_f - e_f) - (U_f + U_r) \cdot h + MW_{ys} \tag{27}$$

From these equations, with equal rolling resistance coefficients for all wheels, an equation for the longitudinal motion can be calculated.

SIMULATION RESULTS

This system was implemented using MATLAB 6.5. The initial experiments used real-time data from an image sequence taken on the Selangor highway. For the experiment, we used six different video scenes recorded on the highway. Fig. 11 shows the graphical user interface (GUI) of this system.

The vehicle dynamics and controller were implemented in this system. Since the processing of this system was based on a video sensor, we assumed that, in the first frame of the video, the vehicle is driving at a certain velocity and, in the last frame, the vehicle is braking, and so the velocity changes. Therefore, we calculated the brake force, velocity, and acceleration of the vehicle and the system then shows the graphs.

The results in Figs. 12, 13 and 14 were based on the video scene shown in Fig. 3. We processed 30 frames of the scene. By setting the vehicle speed in km/h and the braking time in seconds, we calculated the vehicle dynamics. In this case, we set the vehicle speed at 60 km/h and the braking time as 1.25 seconds. Using the graphs, we identified the time that the vehicle was moving as 4 seconds.

From the simulation, we identified that the

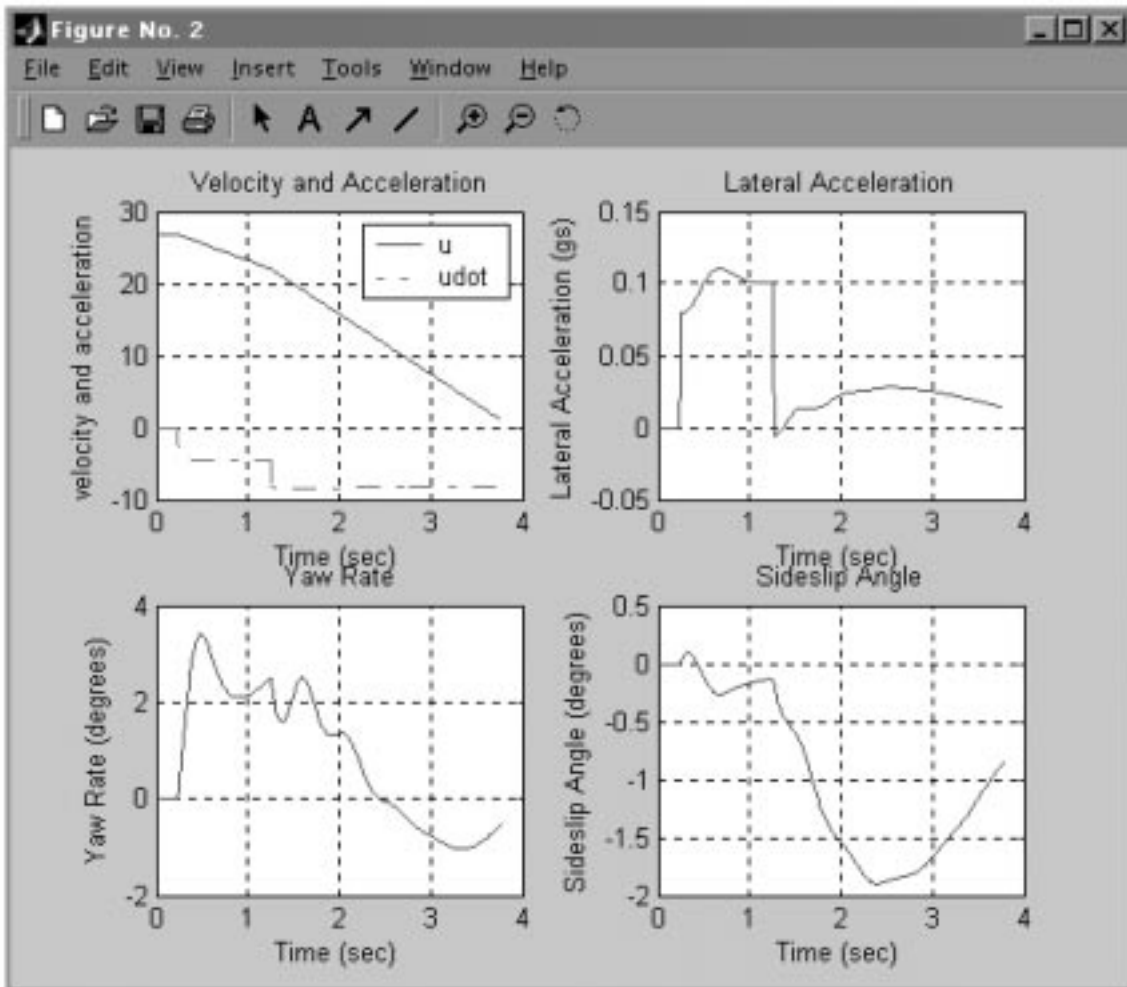


Fig. 13. Velocity and longitudinal acceleration, lateral acceleration, yaw angle, and sideslip angle.

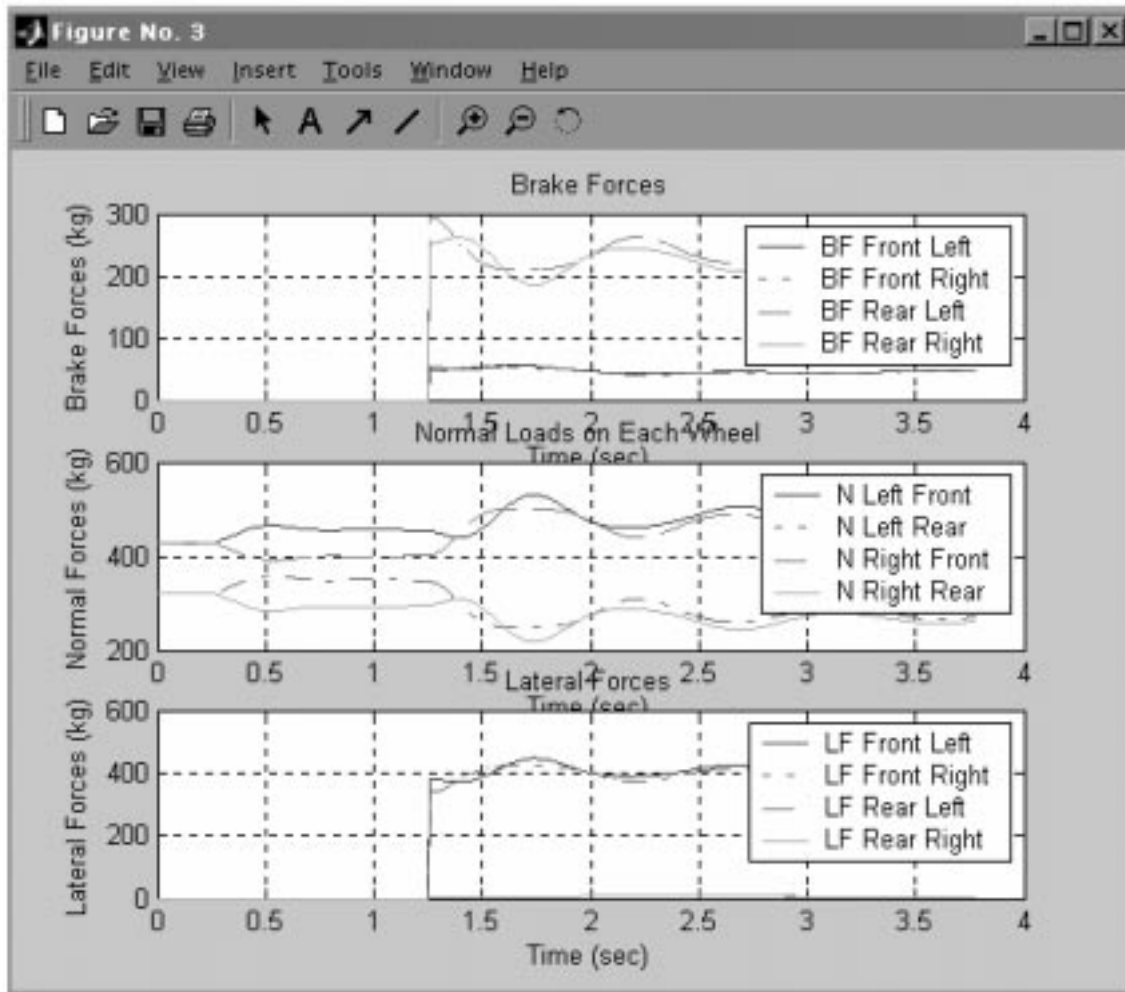


Fig. 14. Brake forces, normal forces and lateral forces.

steering angle for the vehicle was 88 degrees and the roll angle after 1.25 seconds was diminishing. This is because at 1.25 seconds the vehicle has the brake force until it stops moving. The vehicle trajectory in this situation reached 60 meters.

We calculated from the velocity graphs that the velocity when the vehicle is starting is 26 m/sec and that it is getting slower because of the vehicle's brake forces. When the velocity is slower, the longitudinal acceleration decreases. In the lateral acceleration graph, it shows that the acceleration in the braking situation is much lower than in the normal situation.

As mentioned earlier, the vehicle experiences normal forces, lateral forces and brake forces. Since we set the braking time as 1.25 seconds, so the vehicle experienced brake forces starting 1.25 seconds before it stopped. From the brake forces graph we identified that the rear tires experienced heavier brake forces than the front tires. This is because the vehicle modeled in this system used a front-wheel braking system, so most of the brake forces for the front tires came from the braking system component and not totally from the tire-to-road interface forces. On the other hand, the rear

tires experienced brake forces from the tire-to-road interface forces, which provides a tire-to-road coefficient of friction.

From the normal force graph, we identified that the front tires experienced heavier normal forces than the rear tires. This is because the vehicle engine is at the front, making the normal load at the front of the vehicle heavier than at the rear. For the lateral forces, the front tires experienced heavier lateral forces than the rear tires, because we used a front-wheel-drive system for the vehicle. Theoretically, in front-wheel-drive vehicles, the front tires will experience more lateral forces than the rear tires. Besides that, as mentioned above, the brake forces or longitudinal forces are the opposite of lateral forces. Therefore, by comparing the brake force graph and the lateral force graph we identified that these forces are opposite each other.

In educational terms, the results from this experiment are very useful, as this simulation can be used to determine the vehicle's control performance by setting and entering vehicle parameters such as weight, speed, and braking time. In this simulation system, the vehicle weight can only be changed in the MATLAB source code. However, the vehicle speed,

braking time, and number of frames to process can be changed whilst running the simulation. Engineering educators can also use their video files of a vehicle driving to determine the vehicle lateral control. The simulation result can be assessed by comparing the simulation answer with manual mathematical calculations, because this simulation was based on mathematical equations.

CONCLUSION

In this paper, we presented an autonomous vehicle driving system based on video sequences recorded from a vehicle driving along a highway. The focus of this paper was to develop a simulation system of an autonomous vehicle driving control, so that engineering educators or engineers can use it. This system was based on look-ahead distance and the lane angle, which we obtained

through a lane detection process, because these two factors were used with the vehicle dynamics to create a precise control algorithm. In lane detection algorithms, colour cues were used to conduct image segmentation. Then an edge detection method was used to highlight the lane edges. After that, Hough Transform was used to identify the lanes and determine the look-ahead distance and the lane angles. This method has been tested on video data, and the experimental results have demonstrated a fast and robust system. The results of this simulation showed that a dynamic model approach gave a highly accurate portrayal of the vehicle's behavior and that the controllers designed for this approach are robust to those dynamics. So, by using this system, the teaching of autonomous vehicle driving systems becomes faster and more efficient, because it provides information regarding road and vehicle behavior.

REFERENCES

1. C. Kreucher, S. Lakshmanan and K. Kluge, *A Driver Warning System Based on the LOIS Lane Detection Algorithm*, Proceedings of IEEE International Conference on Intelligent Vehicles (1998), pp. 17–22.
2. U. Franke, D. Gavrilla, S. Gorzig, F. Lindner, F. Paetzold and C. Wohler, *Autonomous Driving Goes Downtown*, Proceedings of IEEE Intelligent Vehicles Symposium '98, Stuttgart, Germany (October 1998), pp. 40–48.
3. M. Lutzeler, E. D. Dickmanns, *Road Recognition with MarVEye*, Proceedings of the IEEE Intelligent Vehicle Symposium '98, Stuttgart, Germany (October 1998), pp. 341–346.
4. M. Bertozzi, A. Broggi, A. Fascioli and A. Tibaldi, *An Evolutionary Approach to Lane Markings Detection in Road Environments*, University De Parma, Italy (2002).
5. M. F. Land and D. N. Lee, Where do we look when we steer? *Nature*, **369** (June 1994), p. 30.
6. U. Ozguner, K. A. Unyelioglu and C. Hatipoglu, An analytical study of vehicle steering control, in Proceedings of the 4th IEEE Conference on Control Applications (1995), pp. 125–130.
7. J. Ackerman, J. Guldner, W. Sienel, R. Steinhäuser and V. I. Utkin, Linear and nonlinear controller design for robust automatic steering, *IEEE Transactions on Control Systems Technology*, **3**(1) (1995), pp. 132–142.
8. C. J. Taylor, J. Kosecká, R. Blasi and J. Malik, A comparative study of vision-based lateral control strategies for autonomous highway driving, *The Int. J. Robot. Research*, **18**(5), (1999), pp. 42–453.
9. Y. Ma, J. Kosecká and S. Sastry, Vision guided navigation for nonholonomic mobile robots, *IEEE Trans. Robot. Automat.*, **15**(3), (June 1999), pp. 521–536.

Khalid Bin Isa received B.Sc. in Computer Science from the University of Technology Malaysia (UTM), majoring in Computer Graphics. His thesis was on Noise Cancellation Using Fourier Transform for Speech Recognition. He is currently a Masters student in Computer System Engineering at the University of Putra Malaysia (UPM). His research interests are in the application of digital images and signal processing to real-time systems and to robotic, and satellite, mobile and wireless communication. He is also an Assistant Lecturer at the Faculty of Electric and Electronic Engineering, University College of Technology Tun Hussein Onn (KUiTTHO).

Adznan Bin Jantan is an Associate Professor in the Department of Communication and Computer System Engineering at the University of Putra Malaysia (UPM). His research interests are in digital signal processing and artificial intelligence systems.