

Justina: A platform for autonomous land vehicle research and development

Module: Entwicklung Autonomer
Mobiler System

Assignment by: Tanay Kotecha.

TABLE OF CONTENT

(1)	AUTONOMOUS VEHICLE: HISTORY AND INTRODUCTION.	3
(2)	PLATFORM CONSTRUCTION & SYSTEM DEVELOPMENT	4
	2.1) Modification made to the vehicle	5
	2.2) Computer System	5
(3)	Image processing system	6
(4)	LINE-FOLLOWING APPLICATION	11
	4.1) Line detection	11
	4.2) Line following	12
	4.3) Distance warning	13
(5)	HARDWARE IN LOOP SIMULATOR	14

1) AUTONOMOUS VEHICLE: HISTORY AND INTRODUCTION.

History:

To travel and see different places and to do it as easily as possible, is one of the biggest development and achievement in the history of the mankind. First people used animals like horses but the limitation of power was always a concern. Invention of wheels was a first step towards development, which opened many possibilities. Mr. Nicolas Joseph Gugnot in France in 1769 built a steam powered vehicle. In 1861, when Mr Nicolas August Otto developed his engine and combustion engine started to used in vehicles. And from then onwards there are lot research and development taking place in the field of automobile. The biggest development is in the form of automation. Research on the development of autonomous Vehicle started in late sixties. Today, there are many different projects trying to develop autonomous vehicles either for inside or outside use.

The concept of Autonomous vehicle was invented long back in 1920s, but the first self-sufficient (and therefore, truly autonomous) cars appeared in the 1980s, with NAVLAB. From Navlab to the most recent Google Autonomous Driving car, we have come a lot ahead. But the Research for the improvement will always be a carried.



NAVLAB (1980).



Google Self Driving Car.

What is JUSTINA?

It is a platform for research and development of autonomous land vehicle, which was developed to make it possible to conduct tests in natural surroundings (outdoor). The platform is based on a small car, which was modified and equipped for computer control.

The Research and development of this autonomous vehicle was done by MATTI JUHALA , in year 1992, in Helsinki University of Technology, Finland. From the year 1980 -1992 there were already around 15-20 different autonomous vehicle developed and tested. But the difference between JUSTINA and these autonomous vehicles was that, the other projects before Justina were based on combinations of expensive or special purpose computing systems, prototype image processors and special purpose vehicles and therefore most of the vehicle were too costly, but in this platform a special attention was given to low-cost

requirement and therefore, most of the hardware is composed of low-cost of-the-shelf products.

Criteria of the Platform at the beginning:

There some criteria set, before the start of every project and in case of JUSTINA was no other. The main criteria are as follows:

- A) The platform must be affordable within the low budget and limited resources that can be given to this project.
- B) The platform must allow to be ran in the off-road area and streets near the automobile laboratory.
- C) The platform must be equipped with the image processing and other navigation facilities.
- D) The basic driving system must function reliably under computer control at low speeds.
- E) The computer system must be developed so that new algorithms and sensor systems can be added without the need of wide reprogramming of the installed systems.

2) PLATFORM CONSTRUCTION AND SYSTEM DEVELOPMENT:

2.1) Selection and modification if the vehicle:

For the construction of this platform, a small commercial car, Subaru Just 1.2 EVCT, was selected. The main reasons to select this car are as follows:

- 1) A commercially available small car is less expensive.
- 2) The limited resources of this project can be concentrated on the development of the control, vision and navigations methods instead of the using them on the vehicle itself.
- 3) There is a shielded place for the installation of the computer system in the rear seat and luggage area.
- 4) Subaru Justy was a modern Car at that time, which war more friendly to the enviourment, especially when it is equipped with catalyzer.
- 5) If the car is small it is possible to operate in a limited space.



There were not many modification made in the vehicle, since the main focus was that, the driver should have the complete control over the vehicle in critical situation.

Modification made in vehicle.

- 1) In the interior, the rear seats were removed and the co-driver's seat was reversed. This had provided the necessary space for the computer system and one researcher.
- 2) A new alternator system provided both a 220 v AC and 12 V DC current is added. So that the system has a uninterrupted power supply for the computer.
- 3) The steering and the cruise control is equipped with a stepper motor based actuator and the brakes are converted to be operated by electromagnetic valves.
- 4) A set of micro switches is added to throttle axle for Clutches operation.
- 5) A adaptive PID controller was added to control the throttle valve.
- 6) A joystick was also added to control and steer the vehicle.

For all the above modification there were some important reasons, firstly proper space inside the vehicle was very much important so that the system can be easily installed and a person has space to seat and operate it.

The Steering control has its own processor card with interrupt-controlled software. Stepper motor was used as an actuator, so that I can turn the throttle axle. Steering speed was depended on speed of stepper motor. And the main reason, because of which a stepper motor was used, was that the driver can take control on the vehicle in critical situation. This stepper motor was used in Unipolar mode, because in unipolar mode the torque generated is lower as compared to the bipolar mode. The torque requirement was different in Justina, because they had tested turning the vehicle in two different speeds.

The first test was for more than 3km/h turning speed is 250 steps per/sec, this means turning speed of wheel is $6.67^{\circ}/\text{sec}$. When the vehicle is moving slowly, the turning torque required is high and therefore, turning speed is lowered to 100 steps/sec or $2.67^{\circ}/\text{sec}$ and this was their 2 test. In test the determined torque under good friction was not sufficient, and so further developments were still possible.

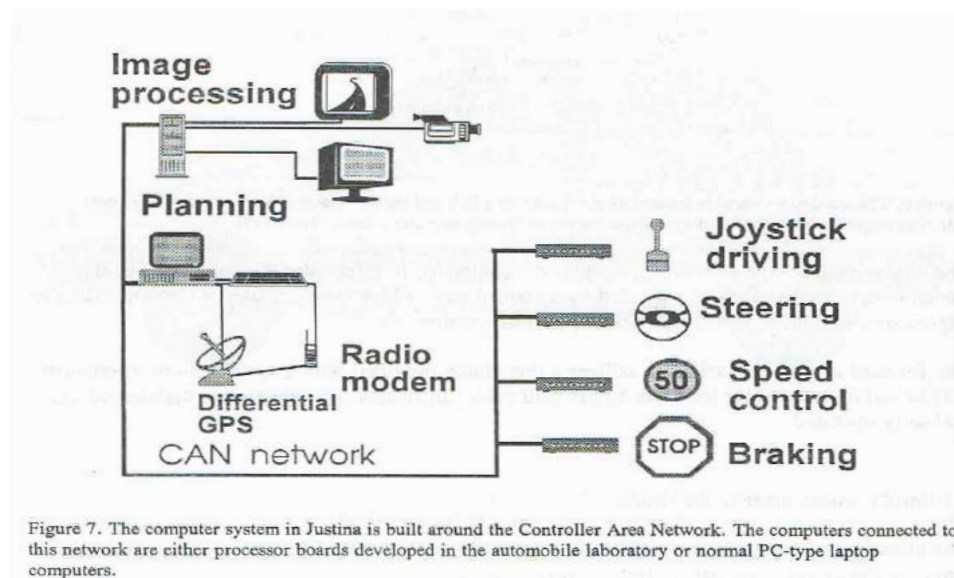
For the control of the magnetic clutch a set of micro switches is added to throttle axle. The switches correspond to those assembled originally in the accelerator paddle. This installation was necessary, because in JUSTINA pedal is not moving under computer control.

The Brake and change over valve modification were also necessary, because in JUSTINA it was necessary to have both controls i.e. Manual and by computer. With this change over valve, the brake line pressure is produced either by the driver via a normal master cylinder or by the computer via an electro hydraulic brake system. With this installation driver can always apply the brakes by foot.

There was also a joystick installed. With this joystick it is possible to steer and control the speed of the vehicle. This joystick consists of two –axle potentiometer and processor card. It is also used for testing of different control algorithms and also during testing of other driving control system to return vehicle back on the road.

2.2) Layout of the computer control

A flexible and modular computer system is built in JUSTINA. The main idea of Justina was to create a platform where different applications can be tested and that was the reason for making the system flexible and modular. It also allows new hardware or software modules to be easily added. Different calculation and control tasks have been distributed to different nodes in this network. There are 4 main layers in this layout, First is the Image processing application, where the image processing is done and different applications like line following can be run. Second is Computer planning. Third is navigation, satellite and GPS and fourth is CAN= Controlled area Network, which controls and transfers the data from all these layers to the output controls. After careful examination of different protocol possibilities, the controller Area Network (CAN) was selected. This protocol is developed by German Bosch GmbH and American Intel co.



3) Image Processing System:

Image processing system consists of :

- i. Black and white video Camera.
- ii. A frame Grabber.
- iii. A frame Processor.
- iv. An IBM- type computer.

Software toolset for the Image processing:

Special workbench developed for testing of algorithms and ideas on a desktop computer using recorded video or saved images.

Workbench is built from:

- a) Standard 80386 level computer
- b) Data translation DT2851 frame grabber and DT2858 frame processor.
- c) VGA monitor and secondary monitor for video images.

Program includes assembler language low level subroutines and C language higher level. The camera was a Sony CCD black & white camera with automatic iris control system and minimum lighting requirement of 3 lux.

The Frame Grabber was a DT2851 (a black and white translation frame grabber) This consists of 512 by 512 pixels in 256 grayscale level and normal European video standard.

The Special image processor was a DT2858 used for fast operation processing of mathematical operation with pixel value.

Frame grabber and frame processor are used in memory mapped mode, because it is possible to handle image pixel elements without need of DMA movements. Memory mapping is done by jumpers for the selection of the base memory. Controlling of the card is done by 16 bit registers. Base address of this register can be selected by jumpers on the card.

The Frame Grabber:

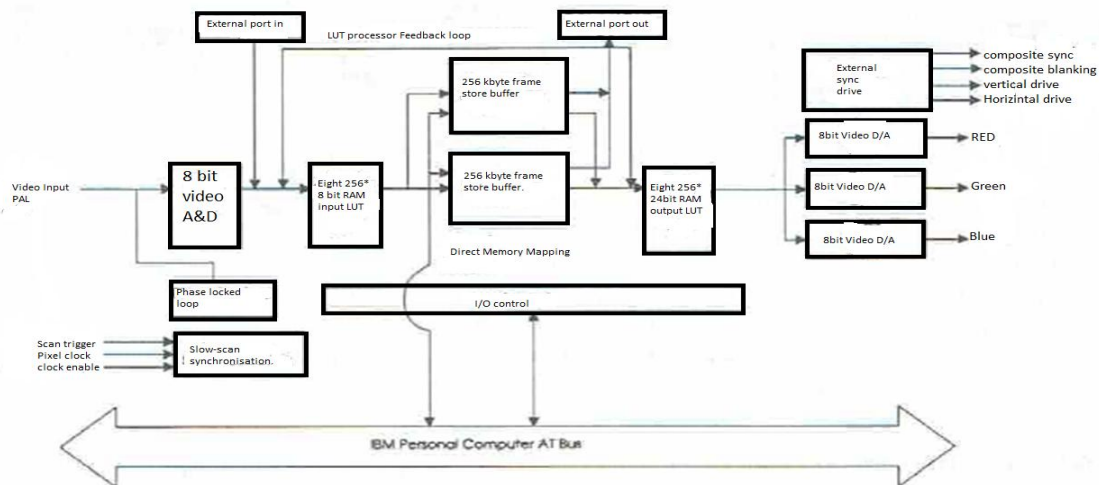


Figure G1. The DT2851 frame grabber can handle video signals in PAL format. The construction includes eight look-up tables for input and output handling of digitized signal. It also includes two frame store buffers. The grabber is controlled via an IBM personal computer AT bus /84/.

Video signal is converted into digital form in DT2851 high resolution frame grabber. The frame grabber consists of eight 256 by 8-bit RAM input look up tables, two 256 by 256 by 8-bit frame store Memory buffers, eight 256 by 24 bit RAM output look-up tables and Three 8-bit video RGB outputs (red green blue). It digitizes an entire video frame in 1/25 of a second. The video signal is converted into 10 MHz rate into 512 lines with 512 pixels per line with 8 bit per pixel. The 8- bit information for each pixel represents one of the 256 possible gray level.

The 8-bit input values are sent through one of eight input LUT, the two 256-256 by 8 frame memory buffer allows digitize input signal to be stored in the memory. These are jumper selectable to occupy any of the thirty two 512kbyte blocks within 16mbyte memory space. The Operations are accomplished through look up table processor. It also consists of eight 16-bit I/O registers for control and status purposes, which are Jumper selectable.

The Frame Processor:

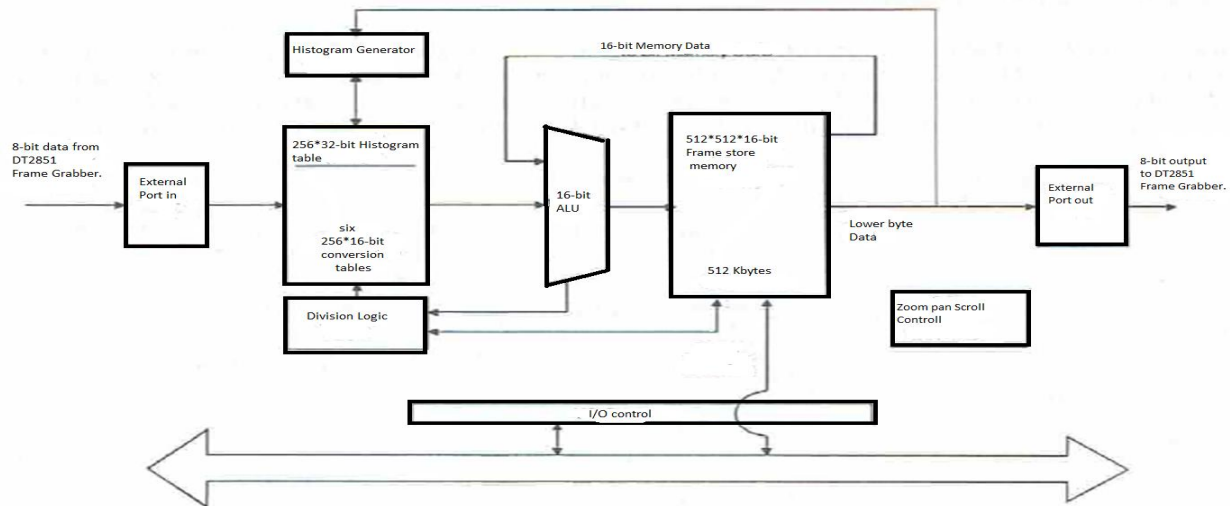


Figure G2. The DT2858 frame processor can be used to speed the image processing. The construction includes six conversion tables, histogram generator, and frame store memory /85/.

The Frame processor consist of a histogram generator, two 512 by 512*16-bit frame store memory buffers, two external input output ports, which are connected to DT2851 frame grabber and a zoom pan scroll control.

The main functions of DT2858 frame processor are as follows,

- DT2858 is designed for high speed 16- bit image processing
- Addition and subtraction of constant
- Multiplication by a constant
- Arbitrary nonlinear transformation of pixel value¹¹¹
- Addition and subtraction of two frame
- Frame averaging
- Division and normalization
- Histogram generation
- Zoom pan and scroll

All the operations are Stored with 16-bit of precision to prevent loss of accuracy and eliminate round of errors in intermediate calculations.

Functions of Image Processing:

- Continuous Image acquisition and display on the screen.
- Image acquisition to memory.
- Gray level on the selected row and zooming of image shown on computer.
- A histogram of image on the screen.
- Threshold selection for binary conversion.
- High pass and low pass filtering.
- Edge detection with sobel or laplacian filter.
- Edge detection with horizontal, vertical and 45 degree line filtering.
- Roberts's operation.
- Hough transformation.
- Saving image in a pcx picture format

After the image acquisition from the video camera, it's displayed on the screen. This image is stored then in the memory. Then selected row from image is transferred into graylevel and

zooming of the image is done on the computer. Then the histogram of the image is generated and showed on the screen.

The filters are in form of matrices and are called kernels in image processing. The kernels used in edge detection have a sum of element 0. Thresholding helps to find objects in an image if these objects are significantly brighter or darker than the background. High pass filter is used to make image appear sharper. and low pass is just opposite of it. Low pass smooth out noise, high pass filter amplifies the noise. High pass filter can cause small details to be exaggerated. High pass can sharpen the image but overdoing it can degrade the quality of the image. Sobel and all the other filters except High pass and low pass filter are edge detection filters.

$$\begin{array}{l}
 \text{Highpass} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad \text{Laplacian} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \\
 \text{Lowpass} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \text{Vertical line} = \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix} \\
 \text{Sobel} = \begin{bmatrix} 1 & 2 & 1 \\ -1 & -2 & -1 \end{bmatrix} \quad \& \quad \begin{bmatrix} -1 & 1 \\ -2 & 2 \\ -1 & 1 \end{bmatrix} \\
 \text{Horizontal line} = \begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} \\
 \text{45 degree line to the right} \quad \text{45 degree line to the Left} \\
 \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix}
 \end{array}$$

In these functions of the image processing, one of the main function is Hough transformation. After the edge detection is done from the above filters, this Hough transformation is used to do line detection from the edge.

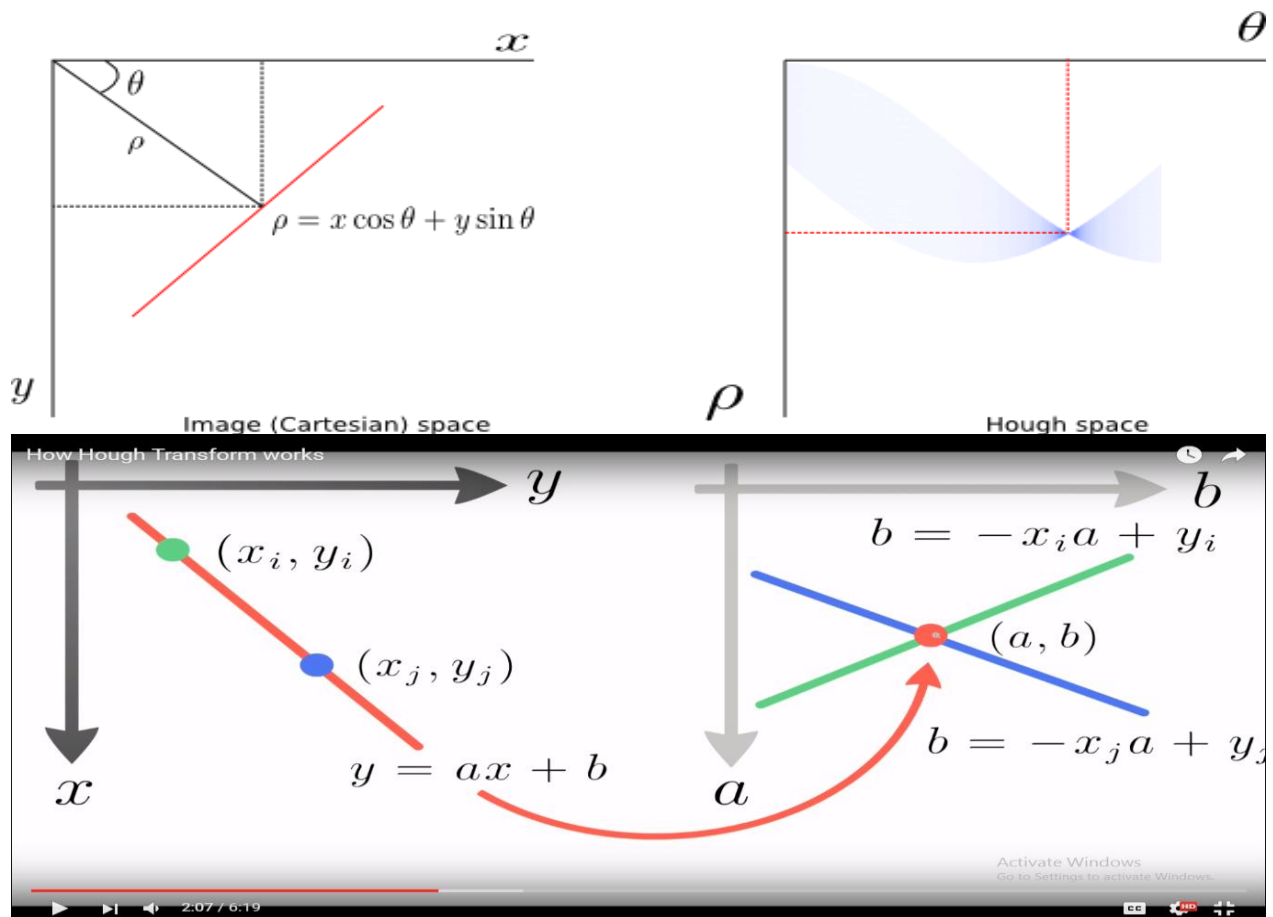
Hough transformation:

Hough transformation is used to find aligned points in images that can create lines.

The main functions of Hough transformation are as follows,

- Clean the accumulator
- Acquire the image
- Convert the image into binary form
- Test every pixel in selected image window
- $Y=a.x+b$, where,

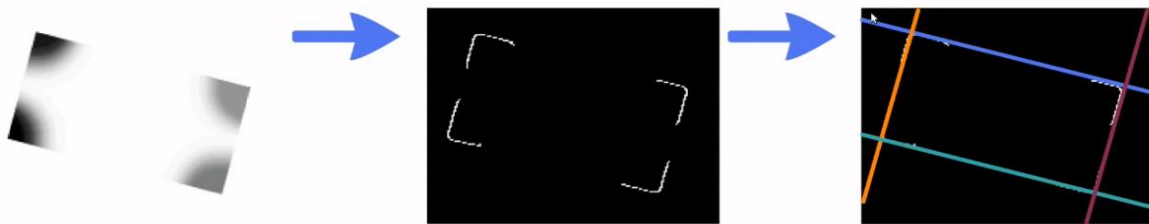
- a= gradient of straight line
b= intercept with the y-axis
- If the binary value of pixel is one, go through every gradient value a and calculate intercept value of b
 - $B=y-a.x$
 - Add one to the accumulator having indices a and b
 - Search for the accumulator having the largest value
 - Display found line on the monitor



input image

edge image

detected lines



Here in the above photo, we can see there are two different equations shown, one is in the first photo where $p = x \cos \theta + y \sin \theta$ and in the second photo $y = ax + b$, both these equations are used as Hough transformation, but because of the fact that sin and cosine calculation takes longer time in the computer, the second equation was used.

Every point has infinity number of lines going through it. The computer time can be shortened by using look up values from a table in the memory. If the angle of the possible line is not near 90° than the gradient approaches infinity. This means that the range for a is very large and accumulators grows beyond the memory capacity. If the gradient is not too wide and is near to 90° , then the above equation can be used. And for 180° sine and cosine must be used.

In the second image the equation used to calculate a line is explained, it shows in the first picture a X-Y plane where this line is located having two points (X_i, Y_i) and (X_j, Y_j) . In the other picture it shows a a and b parameter area, where green line is number of possible green points and blue line is number of possible blue points. When we intersect these two lines we get a common point (a, b) , at this point we can calculate the value by putting this a and b value in the equation. All the points from the red line must be passing through this common point (a, b) . once we get the point we add one to accumulators having the value a and b and search for the accumulator having largest value. This way a line is detected from the edge. This Hough transformation was used for line detection initially, but now it is used for circle and ellipse detection.

4) LINE FOLLOWING APPLICATION:

The line following application is a procedure to detect a line and follow the same line. This application consists of three things,

- 4.1) Line Detection
- 4.2) Line Following
- 4.3) Distance warning

This application was developed using video signal. It was tested on natural sandy, worn and dirty center line marking. Wooden boards were used to simulate the road marking. During the test it was confirmed that it functioned adequately at 15 km/hr. A video camera is often used in autonomous vehicle so that a local map of the surrounding is built while the vehicle is moving so that local map can later be used while navigating in the same area. But, video camera in Justina was used differently, which gives direct feedback, without any map building. The main advantage of this was, calibration of camera was not required. But the disadvantage was that, there was no local map already stored. For this line following application right installation was necessary, therefore details of it were also provided.

Construction of camera;

- 1) Situated inside the car in place of rear view mirror.
- 2) Facing on center line forward and slightly downwards.
- 3) Inclination angle 0.15 radians.
- 4) 3 to 78 m front and 2 to 35 m wide.

4.1) Line detection

The step by step process of the line detection is as follows,

Image capturing

Gray level

Threshold selection

Converting to binary

Selected threshold given to driving program

Line detection (Hough transformation: a and b value sent to CAN with count of white Pixel)

Initialization of CAN and Image processing

For the edge detection, the captured image is converted in to binary form. Here the threshold limit is difficult to determine, because of the weather condition, especially shadow and reflection from water are the problem. Variation of the line color intensity is also considered. The Thresholding limit is set manually before the test runs, the Thresholding limit is selected so that separation of the lines is achieved, The procedure of an automatic thresholding selection was also developed and tested, Automatic Threshold selection was based on gray level histogram. The histogram is calculated by a step of eight gray levels. Threshold is found summing the histogram value of different gray level group down from the totally white. Threshold is the histogram value of the group when the sum reaches predetermined limits. The procedure tries to keep the number of white pixels in binary image almost constant. Threshold limit is tested at the beginning and when white pixel is outside the selected limit value.

When window shown in the figure below is transformed to ground level, the limits are

Distance left limit right limit

6.3m -1.5m 0

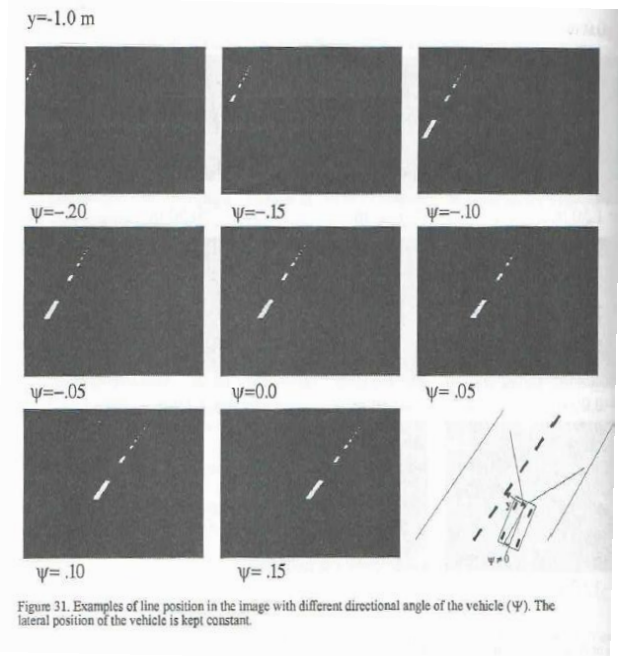
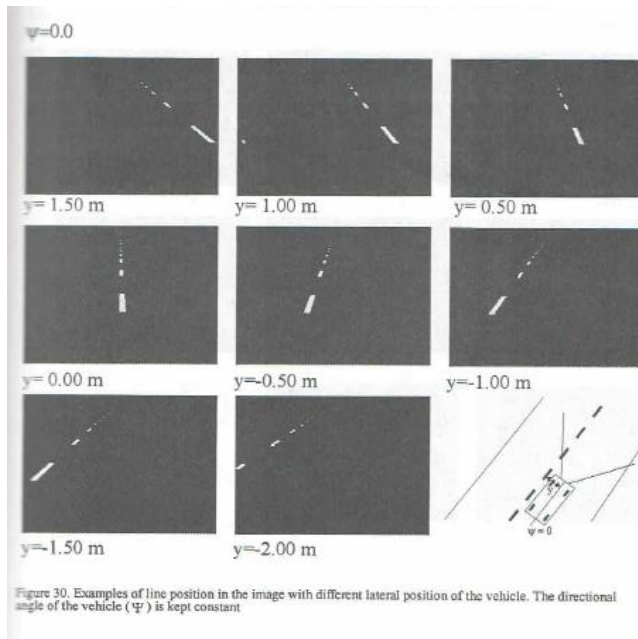
19.2m -4.0m -1

With this limits, values of a and b are calculated using this selectable limits and step for a



4.2) Line Following

The line following is done using another computer. It allows easy combination of different Navigation sensor. One of these sensors is IMAGE PROCESSING COMPUTER. Navigation based on the set of target points (Defined by Distance, Speed, Angle). The angle definition can be either constant or follow the center line. The vehicle is traveling at a constant speed. This speed sent through throttle control computer. Required angle is sent through steering control computer via CAN timed by computer interval timer. This happens 10 times every second.



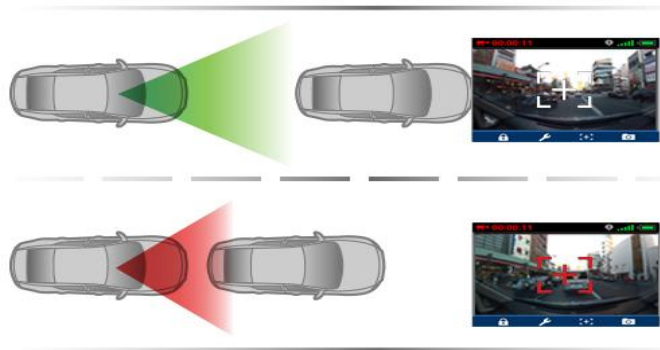
In this two image, we can see that in first image the angle is kept constant throughout $\Psi=0$ and the distance keeps in changing. In the second image the distance $y=0$ and the angle Ψ keeps on changing.

Line position is measured at certain distance from vehicle or certain video line. The limit for line following in JUSTINA was based on distance from the origin to centerline. This was 10.7 meters from origin. Or line 200 in the image. The purpose off steering control is to maintain the relation of the line and the vehicle in the lateral position. The steering angle correction is calculated from error between this point and the set value using adjustable amplification and null correction. The control tries to steer the vehicle, so that center line is kept at 10.7 m in front and -1.5 m in lateral direction.

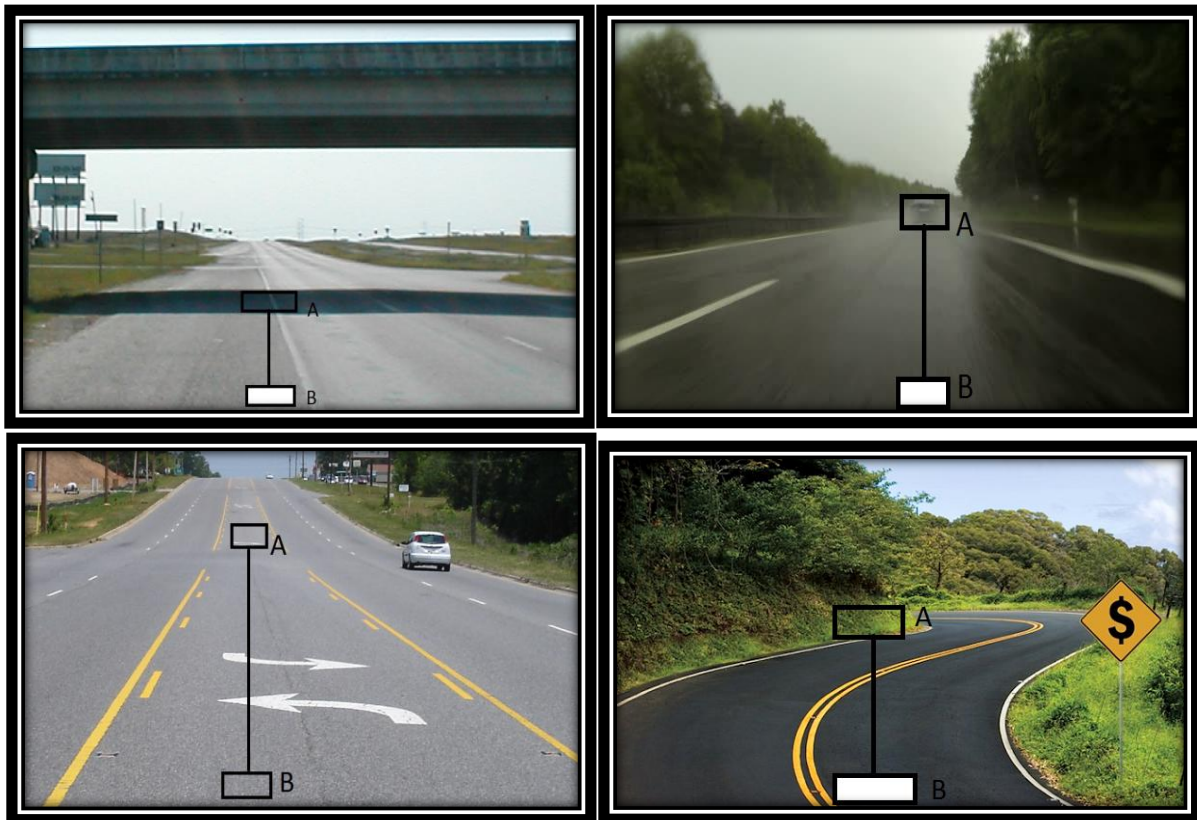
4.3) Distance warning

A new application was developed for distance warning. The main purpose was, to warn the driver, when another vehicle is near. Basically it is based on gray level change. The application was so build, that there were two windows, which were used for the distance warning, the gray level was calculated in these two windows. If the difference between the gray level of the window was more than the limit value, then there something unusual in front of the vehicle. And the signal was sent to the vehicle. During test, the accuracy using this

simple program was around 90%. The inaccuracies were due to incorrect warnings due to shadow or extra white spots.



Distance warning.



Inaccuracies in Distance warning application

In the above images we can see every image has two windows A and B, in the bottom left image there is no change in the graylevel of both the window and that's shows that there is nothing in front of the vehicle. On the other hand in the other three image due to shadow or vehicle or by some other possibility there is a change in the graylevel in window B. once this greylevel change has been detected, the warning is immediately sent to the vehicle driver.

5) Hardware in loop simulator

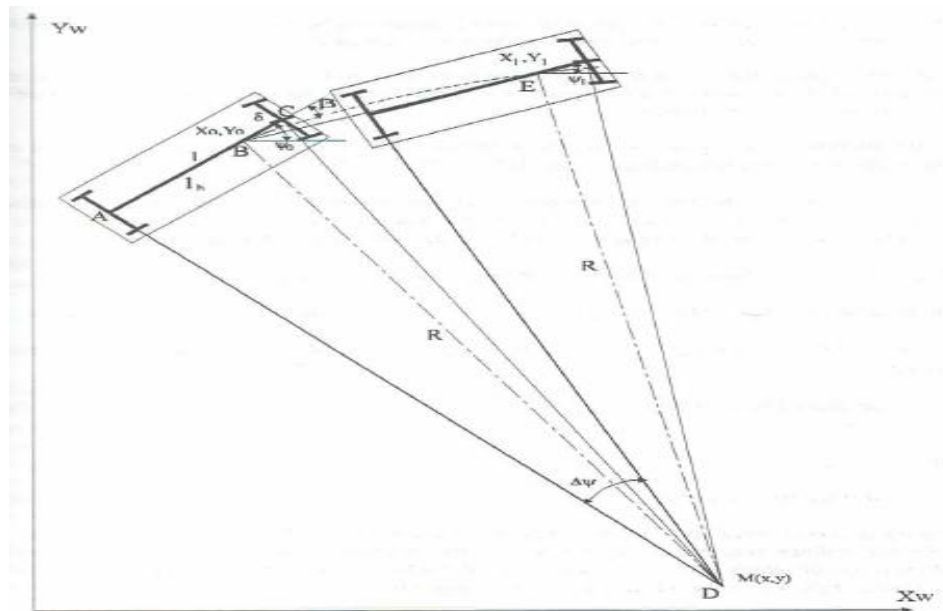
Hardware in loop type simulator was developed for Testing of Autonomous Vehicle vision control program and Selection of different program parameters.

Main elements of this Simulator are,

- Calculation of vehicle current position. ($t_1 = t_0 + \Delta t$)
- Construction of an artificial image.
- Calculation of vehicle control parameters.
- Post- processing the results.

For Vehicle current positions, when time interval Δt is kept small then following Assumptions made are,

- The speed of the vehicle is constant during calculation interval
- The mean steering angle does not change.
- Vehicle is turning over a constant momentary pole, M.
- Moments occur without slip.
- Following Ackerman's Steering theory.
- Role of the vehicle is neglected.
- Road is assumed to be flat



Ackerman's steering theory.

To make the image more realistic, there is a possibility to add a selected amount of white or black points randomly over the image. This disturbs straight area boundaries and makes them more real. This artificial image does not correspond to image produced from the real world by thresholding at a certain gray level, but for the parameter selection purpose it builds a adequate repeatable surroundings. Then this image is transferred to image processing part of the vehicle control program. And calculation time is measured and used as delta t for vehicle position calculation.

The main purpose of the simple simulator is to help search for the different parameters that can have an effect on the accuracy and speed of the control in a real vehicle situation. As an example some parameters are listed;

- Observation window size and position inside the image.
- Resolution inside the window.
- Resolution and limits of the line parameters a and b in Hough transformation.
- Steering speed.
- Camera positioning and focal length.
- Reference point selection for steering error calculation.
- Control method and parameters for steering correction calculation.
- Image quality (amount of extra white pixel).

There were many simulations carried, but an example of the simulation results the image below is shown. Here the effects of driving speed have been examined. The speeds are 2.0, 3.0, 4.0 m/s. Starting position of the vehicle is 1.5 m offset from the center line and direction of angle of the vehicle is 0.1 radians. Due to it has a high steering angle at the start. On straight road there is a small variation of the steering angle which is caused by broken centre line and the limited resolution. At curved part of the street the variation of the steering angle comes higher. This depends strongly on driving speed. At higher speed variation is high.

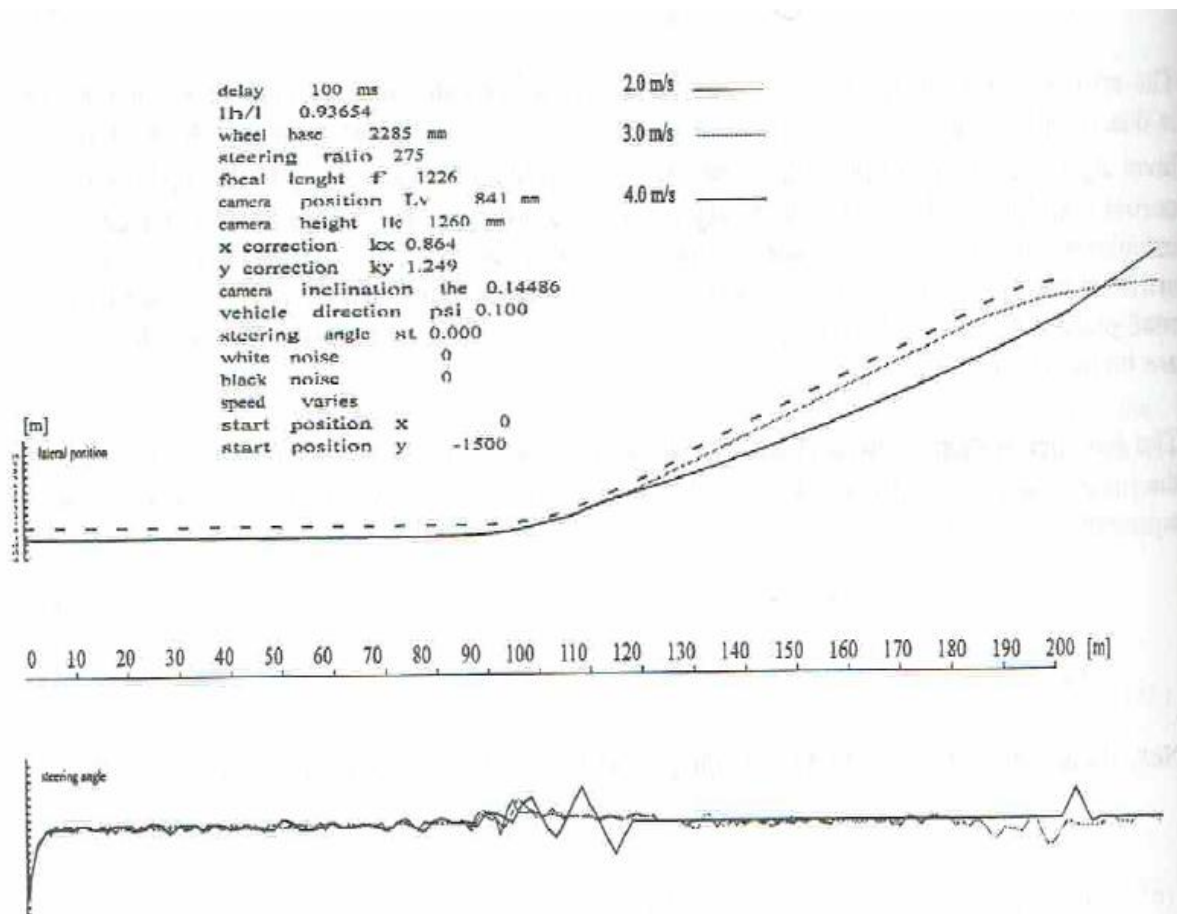


Figure 44. A simulation test run plot. In the plot the effect of driving speed has been tested.