Autonomous Land Vehicle Navigation using Artificial Neural Networks

Muthu Ganesh P.
R.M.K. Engineering College
Department of Information Technology
Chennai, Tamil Nadu, India

Sai Nithin Singh C.
R.M.K. Engineering College
Department of Information Technology
Chennai, Tamil Nadu, India

Abstract

Autonomous land vehicle navigation is used to assist the driver partially or completely with the help of modern technology in a non-intrusive manner. This work concerns with the addition of an additional neural network which works separately and takes care of the road signs and other such entities. It detects obstacles and tracks them till it is safely passed. The information about the obstacle is passed on to the driving network which deals with the obstacle by steering the vehicle away, or deciding some other means of avoiding the obstacle. Thus, the safety of the system is enhanced.

1. Introduction

1.1 Artificial Neural Networks (ANN):

Artificial Neural Networks, also called parallel distributed processing systems (PDPs) and connectionist systems, are intended for modeling the organizational principles of central nervous system, with the hope that the biologically inspired computing capabilities of the ANN will allow the cognitive and sensory tasks to be performed more easily and more satisfactorily than with conventional serial processors. In simple terms ANN is an attempt to mimic the functionality of the human neural network system (brain).

1.2 Autonomous Land Vehicle Navigation:

Autonomous land vehicle navigation is used to assist the driver partially or completely with the help of modern technology in a non-intrusive manner. Currently, the most discussed technology with respect to autonomous land vehicle navigation is the use of impregnated magnets along a path to guide the vehicle. But the drawback of this system is the cost and the practicality of implementation. Autonomous land vehicle navigation using ANN addresses all these drawbacks along with the added functionality of learning the different driving styles under different circumstances using on-the-fly training [1]. This system uses active sensors to interact with the environment and concurrently learn from the user using a non-intrusive gaze tracking system through ANN. The issues addressed by this system are:

- Rapidly Adapting the Lateral Position of the Vehicle.
- Kinematical Control of Vehicle using Simulated Highways for Intelligent Vehicle Algorithms.
- Inter-Vehicle Interaction using Visibility Estimation Techniques.
- Simultaneous Localization and Mapping with Detection and Tracking of Moving Objects.
- Path Intersection Detection and Traversal.
- Predicting Lane Position for Roadway Departure Prevention.
- Driving in traffic: Short-Range Sensing for Urban Collision Avoidance

1.3 Active Sensor Control for Autonomous Driving System:

The Autonomous Land Vehicle (ALV) is a neural network based system which has been successful in driving robot vehicles in a variety of situations. However, since ALV maintains no state information about the world, but processes each sensor frame individually, it can become confused on sharp curves when the field of view no longer displays the important features in the scene. A steerable sensor allows the perception system to select the desired field of view to maximize the information content of a sensor frame. For a vision system that builds a map of the road, it is straightforward to point the camera in the desired direction, but ALV directly outputs a steering command, without generating an intermediate road representation. The system interprets this steering command as a point on the road and pans the camera in the desired direction. However since ALV is trained with a fixed sensor orientation, the position of the sensor during training is implicitly encoded in the weights and moving the camera results in the outputs of the network being invalid for the given configuration. The system solves this problem by post-processing the steering response of the neural network as a function of the current sensor configuration. A significant advantage of this approach is that existing networks can run under this new system without any modification or retraining.

The system's basic architecture is a three layered artificial neural network shown in figure. A reduced resolution camera image is fed into a 30x32 array of input units, which are fully connected to a hidden layer of 4 units. The hidden units are fully connected to a vector of 30 output units, and the steering response is given as a Gaussian activation level centered on the correct steering curvature. ALV's neural net is trained "on-the-fly", and the human driver's steering responses are used as the teaching signal. ALV is able to learn from this limited data by artificially expanding its training set. Each original image is shifted and rotated in software to create 14 additional images in which the vehicle appears to be situated differently in relation to the road. The training signal for each of these new images is calculated by assuming a pure pursuit model of driving and transforming the original steering response accordingly [1].

1.4 Inter-Vehicle Interaction using Visibility Estimation Techniques:

Reduced visibility is one of the key factors in many traffic accidents. It is very difficult to consistently find high contrast targets at various known ranges from a moving vehicle. This system overcomes this difficulty when detecting the position and curvature of the road ahead in camera images by utilizing whatever features are visible on the roadway, including lane markings, road/shoulder boundaries, tracks left by other vehicles, and even subtle pavement discolorations like the oil stripe down the lane center when necessary.

In order to estimate visibility the road feature should be detected. In this process an aerial image of the road is taken and a cross-section of the aerial image perpendicular to the road, called the road template is created.

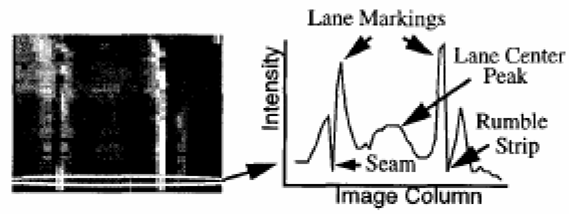

Fig. 1 – Image used to study the road using intensity as a parameter [4].

All the particulars necessary are taken from the road template and finds out the road ahead. The system adjusts the template left or to the right until it matches the particular row's cross-section. The amount of shift gives the lateral displacement.

1.5 Driving in traffic: Short-Range Sensing for Urban Collision Avoidance

This system addresses the issues involved in traffic driving. The requirements for an effective collision avoidance and warning system for urban environments, include the following as a minimum standard:

- Sensing
  - State of own vehicle
  - State of nearby objects
  - Environment
- Knowledge Base
  - Model of the own vehicle and driver
  - Model of other objects
  - Model of environment
  - Model of interaction between all of the above
- Processing and Algorithms
  - What situation we are in?
  - How likely is a collision?
  - How dangerous is the situation?
  - Is an action needed?
- System Response
  - Aware : Baseline Situational Awareness
  - Alert : Potential Obstacles
  - Warn : High Likelihood of Collision
  - Evade : Imminent Collision
  - Notify : Collision has occurred

One method of sensing the nearby objects in an urban environment is using a laser line striper shown in the figure below [2].

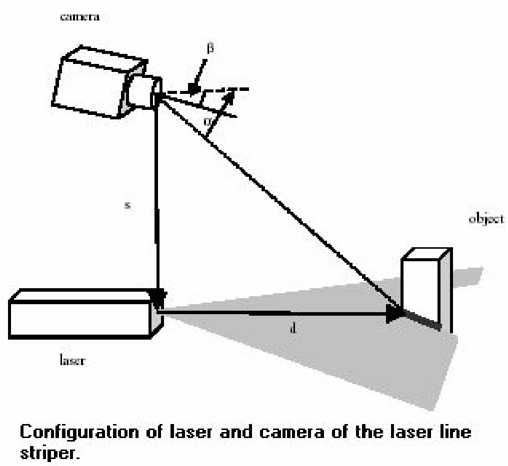

Fig. 2 - Laser and Camera configuration used for short range detection of objects.

1.6 Overtaking Vehicle Detection

To detect vehicles, we do the following: first, we sample the image, perform edge detection, and use our planar parallax model to predict what that edge image will look like after traveling a certain distance. Next, we capture an image after traveling our assumed distance, and compare it to the prediction. For each edge point in the predicted image, we verify that there is a corresponding edge point in the actual image. If there is a match, then our prediction (based on a flat earth assumption) is verified. Otherwise, we know that the cause of the horizontal line in the predicted image was an obstacle (i.e., above the ground plane). There are 4 components to the system:

- Sampling and Preprocessing,
- Dynamic image Stabilization,
- Model-Based Prediction, and
- Obstacle Detection.

The figure 3 on the right is the difference image obtained by taking the difference of the images actually sampled and the image predicted by the system. When the noise is analyzed, the vehicle on the right easily stands out, since its predicted path of motion is varying greatly from its actual path of motion. Thus it is concluded that it is overtaking.

Fig. 3 - Rear View Road Image [3]

Fig. 4 - Same image after 120 ms [3]

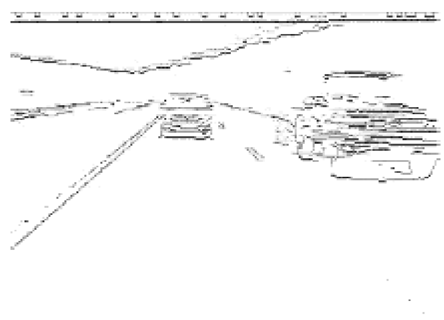

Fig. 5 - Difference Image [3]

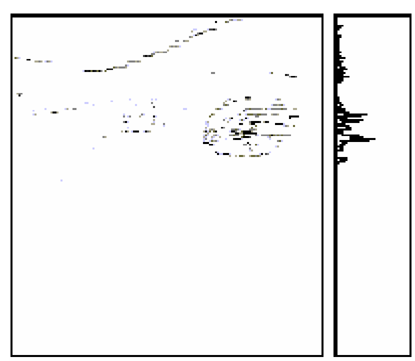

Fig. 6 - Obstacle Image [3]

2. Obstacle Tracking

Our work is extending the above working model developed by NAVLAB, Carnegie Mellon University by adding an obstacle tracking system. We have taken up a two dimensional case, for greater flexibility in case of contour changes on the road. With the initial positions of the obstacle and the autonomous vehicle, the bearing information is simulated using sensor simulator, the output of which is fed to the Least Square Estimator (LSE) filter which gives the estimated obstacle parameters. The errors between the estimated and the simulated obstacle parameters are compared. To reduce estimation error, the backpropogation neural network is incorporated with the LSE filter. The network is trained for a set of inputs and after testing, the network estimates the obstacle parameters. The errors between the simulated and the estimated values are compared with the errors obtained without the aid of the network.

2.1 Tracking Model Derivations

2.1.1 Mathematical Model:

System model at state k+1:

$$X(k+1) = A.X(k) + B.U(k) + W(k)$$

where

$$X(k) = \begin{bmatrix} r.x(k) \\ r.y(k) \\ v.x(k) \\ v.y(k) \end{bmatrix} = \text{State Vector}$$

$$= \begin{bmatrix} \text{Range in x-direction at time k} \\ \text{Range in y-direction at time k} \\ \text{Velocity in x-direction at time k} \\ \text{Velocity in y-direction at time k} \end{bmatrix}$$

$$U(k) = \begin{bmatrix} v.x(k) \\ v.y(k) \end{bmatrix}$$

$$= \begin{bmatrix} \text{Change in relative velocity in x-direction between time k and k+1.} \\ \text{Change in relative velocity in x-direction between time k and k+1.} \end{bmatrix}$$

W(k) = System Noise

$$A = \begin{bmatrix} 1 & 0 & kT & 0 \\ 0 & 1 & 0 & kT \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \text{State Transition Matrix}$$

$$B = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} = \text{Input Matrix}$$

T = Sampling Period

k = Sample Number

U(k) is concerned with vehicle dynamics with respect to the obstacle. Since the vehicle is assumed to be moving with a uniform velocity within the infinitesimal period between k and k+1, B.U(k) term can be taken as zero for theoretical verification purposes. By assuming the system noise as zero, the system model becomes:

$$X(k+1) = A.X(k)$$

2.1.2 Measurement Model:

$$Y(k) = H.X(k) + \upsilon(k)$$

where,

Y(k) : Measured bearing at time k
H : [cos b -sin b 0 0]
: Measurement Matrix
b : Bearing
$\upsilon(k)$: Measurement noise component of the appropriate order.

2.1.3 System Dynamics Model:

The Cartesian state vector formulation is as follows:

Let 'k' be any arbitrary time instant,

$$X(k) = \begin{bmatrix} rx(k) \\ ry(k) \\ vx(k) \\ vy(k) \end{bmatrix}$$

rx(k) = rtx(k) – rox(k)
ry(k) = rty(k) – roy(k)

where,

rx and ry are relative ranges along x and y directions between the vehicle and the obstacle.
t : refers to the obstacle (target)
o : refers to the vehicle (observer)

The measurement process is described by non-linear elation:

$$b(k) = \arctan(rx/ry)$$

where,

b(k) represents the measured target (obstacle) bearing at the k^{th} instant of time and taking tan on both sides we have,

$$\tan(b(k)) = (rtx(k) - rox(k)) / (rty(k) - roy(k))$$

or,

$$\sin(b(k))\ (rty(k) - rox(k)) = \cos(b(k))\ (rtx(k) - rox(k))$$

but,

$$b(k) = bm(k) + v(k)$$

where,

bm(k) is the actual measured bearing at k^{th} instant and v(k) is the measurement noise at k^{th} instant.

This can be formulated as follows avoiding the subscript 'k':

$$(rtx - rox) \cos(bm) - (rty - roy) \sin(bm) = - rs(k).\sin(v(k))$$

where,

$$rs(k) = (rtx - rox) \sin(b) - (rty - roy) \cos(b)$$

i.e,

$$rox.\cos(bm) - roy.\sin(bm) = rtx.\cos(bm) - rty.\sin(bm) + rs(k).\sin(v(k))$$

In the above equation the left hand side denotes the measurement vector H(k), and is chosen as,

$$H(k) = [\cos(bm) \quad -\sin(bm) \quad 0 \quad 0]$$

Therefore the observation sequence is as follows,

$$z(k) = H(k).Xo(k) = H(k).Xt(k) + n(k)$$

i.e,

z(k) is the measurement at k^{th} instant,

Xo(k) is the observer(vehicle) state at the k^{th} instant,

Xt(k) is the target(obstacle) state at the k^{th} instant and

n(k) is the noise sequence at the k^{th} instant.

Hence the measurement scalar model

$$z(k) = H(k) . X(k) + n(k)$$

2.2 Backpropogation Neural Network Training

2.2.1 Forward Pass. Calculation in multilayer network is done layer by layer. The NET of each neuron in the first hidden layer is calculated as the weighted sum of all its neuron inputs. The activation function 'F' then squashes NET to produce the OUT value for each neuron in that layer. Once the set of outputs for a layer is found, it serves as the input for the next layer. The process is repeated , layer by layer, until the final set of network outputs is produced.

2.2.2 Backward Pass. The networks actual output from the forward pass is compared with the desired output and error estimates are computed for the output units. The weights connected to the output units are adjusted to reduce those errors. The error estimates of the output units are used to derive the error estimates for the units in the hidden layer. Finally, the errors are propagated back to the connections stemming from the input units.

Before starting the training process, all the weights must be initialized to small random numbers. This ensures that the network is not saturated by large values of weights.

2.3 Least Square Estimator Filter

The Least Square Estimator is one the methods providing Target Motion Analysis (TMA). We propose to incorporate this in our ALV model. Here instead of the target moving, the ALV model moves, and the obstacle remains stationary. The basic task is to estimate accurately to the extent possible, the relative position (Rx, Ry) and the relative velocities (Vx, Vy) of the obstacle, from either the Short Range Sensors or sonar noisy measurements of range and bearing. The obstacle can be a stone, a vehicle (parked or in motion), a signboard, etc. The state vector plays a key role in LSE diverging/converging cases.

The statistical characteristics of the noise depend upon the measuring equipment. It is observed that the LSE is optimum only for the case of Gaussian noise. The LSE is an unbiased, stable, and optimal estimator with minimum variance, if the system is stochastically controllable and observable, with some noise assumptions being satisfied.

The recursive LSE is a linear, discrete time, finite-dimensional and sequential recursive system. It assumes the availability of a state model and an observational model. The input to the filter is a sensor or a sonar bearing contaminated with noise and the output is the obstacle parameters.

2.4 Block Diagram of Network Aided LSE

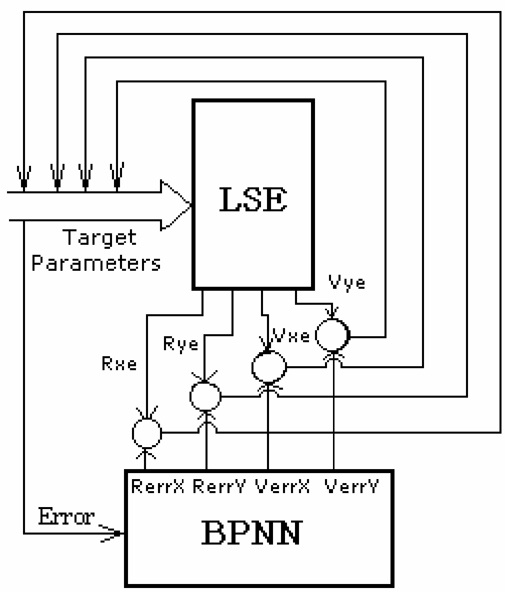

Fig. 7 - Block Diagram of Backpropogation Neural Network (BPNN) aided LSE (Least Square Estimator).

The block diagram shown above illustrates how the LSE functions in combination with the Backpropogation neural network. The compared results of the network and the LSE are fed back and thus the error is deducted.

3. Conclusion

Intelligent vehicles are beginning to appear on the market, but so far their sensing and warning functions only work on the open road. Functions such as run off road warning or adaptive cruise control are designed for the uncluttered environments of open highways. Current sensing/warning/controlling systems generally work only in relatively simple environments. Applications developed for open highways include Adaptive Cruise Control (ACC), which controls the throttle to keep a safe gap behind other vehicles; run-off-road collision warning systems, which alert a driver if the vehicle starts to drift out of its lane; and blind-spot sensors on heavy trucks to warn the driver if they start a lane change without seeing a car in the next lane. Some applications are also on the market for slow speed driving: rear-facing sensors as parking aids, for example. This work of ours gives a spin-off to further studies. Other neural networks such as the Hopfield network can be employed instead of the Backpropogation network.

The Autonomous Land Vehicle Navigation using Artificial Neural Networks puts forward a very promising technology which might change the very way vehicle navigation is perceived as of today. Although still under research, its results are very encouraging and in conjunction with other modern technologies like GPS, ACC, etc. can easily pull down the rate of causality which is very high in today's roadways. The future is very bright for Autonomous Land Vehicles. They have come here to stay and stay they will.

References

[1] Rahul Sukthankar, Dean Pomerleau and Charles Thorpe, Robotics Institute (NAVLAB) - Carnegie Mellon University. Panacea: An Active Sensor Controller for the ALVINN Autonomous Driving System.

[2] Mertz, C., Kozar, J., Miller, J.R., and Thorpe, C., Eye-safe Laser Line Striper for Outside Use. Paper submitted to: IV 2002, IEEE Intelligent Vehicle Symposium, June, 2002.

[3] Parag H. Batavia, Dean A. Pomerleau, Charles E. Thorpe, Robotics Institute, Carnegie Mellon University. Predicting Lane Position for Roadway Departure Prevention.

[4]. Dean Pomerleau, Carnegie Mellon University. Inter-Vehicle Interaction using Visibility Estimation Techniques.