

Robots go automotive - The SPARC approach

Frédéric Holzmann*, Mario Bellino†, Sascha Kolski†, Armin Sulzmann*, Gernot Spiegelberg* and Roland Siegart†

*DaimlerChrysler AG - Truck Product Creation
Stuttgart, Germany

Email: {frederic.holzmann, armin.sulzmann, gernot.spiegelberg}@daimlerchrysler.com

†Ecole Polytechnique Fédérale de Lausanne (EPFL)
Lausanne, Switzerland

Email: {mario.bellino, sascha.kolski, roland.siegwart}@epfl.ch

Abstract— This paper introduces a new concept for advanced driver assistance by means of a redundant architecture including all system components spanning from environment perception to vehicle controllers. The first part of this paper is an overview of the project framework and the research platforms. After that the elements of the architecture themselves will be described. The use of sensors and the fusion of their outputs will be presented. Different controllers will be used depending on the scenarios around the vehicle in order to provide a theoretic solution. This solution will be downsized after that with a dynamic vehicle model to the feasible safe motion vectors. This list of motion vectors will be compared to the driver's command and will lead to the choose of his/her command or an other safe motion vector if the driver does not react convenient to the situation. The final part describes some preliminary results and concludes towards future work and research issues.

I. THE SPARC PROJECT

A. The concept

In Germany each year about 60,000 people are getting injured or lose their lives in accidents on the road. In about 97% of these cases, the accidents are due to a human mistake because of a wrong interpretation of the environment or lack of information. According to [1], about 40% of those accidents could be avoided by some warnings or a preventive and helpful control of the driver's command. The other accidents could be avoided or their consequences drastically decreased by overwriting the driver's command at the last moment.

DaimlerChrysler AG and its partners have started the European project SPARC^{1,2} on January, 1st 2004 with the aim to develop a new concept of active safety system for heavy goods vehicles and personal cars. The challenge is to integrate an intelligent system to achieve a predictive safety technology.

The SPARC technology will first inform the driver if a problem can occur. After that it will warn him/her with a dedicated feedback. If the driver does not react or he/she reacts incorrectly, the system will have to intervene at the

very last moment before the point of no return in order to keep the vehicle in a safe state. Up to that last moment the driver is having the command of the vehicle and the system is only giving a feedback.

B. The architecture

In this project every elements of the short time loop (from the environment sensing to the command control) will be made redundant in order to improve the safety of the vehicle. As defined by Lunenfeld in [2], the tasks to realise are serialised like on figure 1: environment sensing, planning, choose of a scenario and coordination of the aggregates.

Different sensors will be used on the vehicle and their

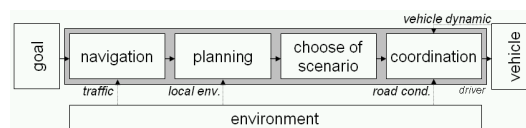


Fig. 1. Model of the data flow into the vehicle

outputs will be fused by a **Data Fusion** to model the environment. The environment model will be provided as a list of objects and a model of the lanes and their corresponding probabilities. The planning function will be realised by an intelligent control (**Virtual Driver**). It analyses the environment and provides the safe driving strategies depending on the geometry and the dynamic of the vehicle, that are described as transfer functions. The driver's command and the safe driving strategies will be compared into the intelligent switch (**Decision Control**). The both commands are given as a motion vector (acceleration, steering) with the driver's fitness and the environment model's quality. The complete architecture is shown on figure 2.

As the system has to balance the driver's command, the physical link between the human interface and the aggregates has to be cut and replaced by the Drive-by-Wire technology. Therefore it will be possible to coordinate the different aggregates with a dedicated ECU, named

¹Secured Propulsion using Advanced Redundant Control

²Additional information is available under the web site <http://www.sparc-eu.net/>

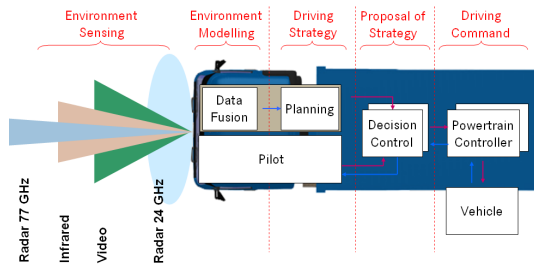


Fig. 2. SPARC Architecture

Powertrain Controller, corresponding to the control task. This Powertrain Controller realizes the given motion vector without asking from the source. After that the human interface could be modified and integrate a feedback.

The function *scenario choosing* could not be integrated twice in the vehicle. The driver's choice will always be accepted by the decision control as long as it is not dangerous. Otherwise the virtual driver will choose itself the new scenario without taking the driver's command into account.

II. DEMONSTRATOR AND RESEARCH PLATFORMS

The EPFL works on new concepts on data fusion with its own vehicle and delivers step after step its results to DaimlerChrysler AG, which integrates its own technology with the results from the EPFL and the other partners into the final vehicle.

A. Hardware framework at EPFL

The EPFL uses a development framework introduced by Fleury et al. in [3]. This framework allows to execute the algorithms as modules on a laptop. The tasks, coded in GenoM follow the same approach that is used in embedded systems like typical automotive platforms.

The hardware basis for these GenoM modules is formed by a Pentium driven notebook running Linux. It is connected to the proprioceptive data perception as well as the CANbus in a SMART ForTwo. Although there is a single system controlling all the different tasks from data perception to environment modelling, a distributed system is simulated as it will be used in the SPARC demonstrators by the use of different GenoM modules for the different task. Thus all processes run independently in their own time slots.

In order to prevent the computational capacity from the complete use by the vision algorithms, a separate processor will be used only for these algorithms. In the first revision of the hardware platform, a computer with a Pentium 4 at 2.8GHz and Windows XP professional is dedicated to vision tasks. The frame grabber that will perform the acquisition is the Matrox Meteor II digital, which is connected to an automotive camera from Siemens VDO. This camera grabs gray images of 750 pixels width and 400 pixels height, with

a frame rate of about 20fps. Moreover, it is equipped with an internal technology that is able to normalize the lighting condition, which allow the vision algorithms to suppose that images are taken with more or less the same contrast. The final revision of the vision platform will completely suppress the initial computer and use an embedded solution. This device will directly be connected to the camera and acquire images of the camera with specific drivers without the Matrox frame grabber. This simplified architecture will allow embedding all the necessary devices to acquire and compute image tasks listed above.

B. Workframe used at DaimlerChrysler AG

The development of the whole system is mainly made in four parts. First the software delivered by the EPFL and other partners are integrated into a model under Matlab/Simulink with specific controllers. As soon as the system reaches a milestone, the generated code is automatically integrated into the multi agent system named ANTS [4]. If the algorithms are validated, they are integrated into the dedicated ECUs and integrated into the vehicle. Most of the scenarios will be checked on a tests bench. After that the vehicle will be tested with its new technology on closed roads.

III. BRINGING THINGS TOGETHER - ENVIRONMENT MODELLING

The actual modelling process is performed in three steps, namely prediction, perception and fusion. The prediction is based on a navigation system delivering both the global vehicle position as well as a map of the actual vehicle environment. This prediction corresponds to the localisation of the vehicle on a street map. Next generation navigation systems will deliver more precise information on road parameters like curvature, etc. Exploiting this information the recognition of the upcoming curves or intersections can be predicted for a long distance ahead the vehicle. This prediction is *in situ* fused with the perception results to obtain a probabilistic environment model. This model contains the position and yaw rate of the vehicle relative to each lane, the width of each lane and the curvatures by means of a curve radius. The obstacles are represented mainly by their positions relative to the vehicle, their sizes and a motion vector modelling the speed relative to the vehicle. The link between the representations of lanes and other traffic participants is done by assigning the single objects to lanes.

A. Collecting data

The data collection forms the first step in the perception process. Here the raw sensor data is read and processed to extract the features like lanes and other traffic participants. The sensors are taken as logical sensors containing both the actual perception and the processing leading to that higher level information.

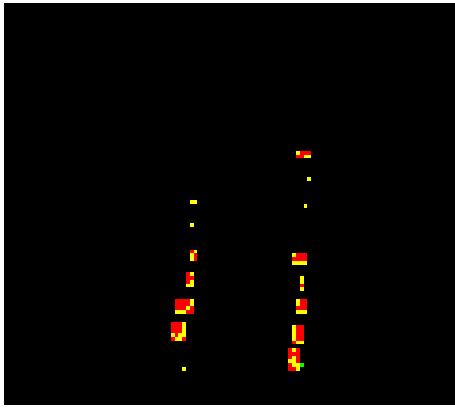


Fig. 3. Structured laser data taken on EPFL parking lot
The elements seen are the boundaries of the vehicles parked ahead

1) *Lidar*: To make those sensor data usable, an understanding of the scene has to take place. How this understanding can be achieved is here demonstrated by means of lidar data. The lidar is implemented as a time-of-flight based scanner emitting laser beams and measuring the time to the receipt of the reflection. Thereafter the collected data delivers a distance for every angle the system performed a measurement. The first step performed on this data is the segmentation separating the data into areas of strong connectivity, which are considered objects. Figure 3 visualises these data taken on the EPFL parking lot. For each of those objects the position and the size are calculated in a coordinate system moving with the ego vehicle. Those objects are tracked over the time to gather information about their motion relative to the ego vehicle. Thus the logical lidar sensor outputs the position, the size and the relative motion of each detected object together with probabilistic variables describing the certainty of each of these parameters.

This method is particularly of great interest for automotive solutions because it rely on several sensors and is not affected by the breakdown of one of the sensors. Moreover, the measurements that are done by the different devices are combined and the resulting position or shape is much more precise if it would rely on a single sensor. This effect is especially observed in case of complementary sensors.

2) *Lane detection*: In recent years, the lane detection algorithms were largely studied, but a lot of research is currently being done to find an optimal solution. The GOLD [5], ARGO [6] and RALPH [7] projects were all based on extraction of vision features, and use one or two cameras depending on the chosen approach. It is clear that such systems are mainly affected by shadows and bad weather as rain or fog. Indeed, the results are strongly related to the contrast of the objects contained in the scene, and changes in illumination can induce severe influence in the algorithm results. In order to build a more robust solution of the lane of the truck, several developments

have been done by fusing or combining the results taken from different sensors. The idea is to use different physical principles to measure the same data in order to get a more robust solution whatever external conditions are present, as rain or fog. Thus, camera-radar [8] or camera-laser [9] [10] combination has been used. The SPARC project will use a similar approach to have a more robust representation of vehicle's lane, which will be described in following section.

One of the major issue in lane detection algorithm is to deliver reliable data in a short processing time. Indeed, the time elapsed between the acquisition and the results of image processing have to be as small as possible to have consistent data. To minimize the computations, the image is not considered globally but several regions of interest³ are defined. Thus, the algorithm does not need to proceed the whole image but concentrate on a smaller region that contains the desired information. A multi-point and multi-hypothesis [11] model is used to define and find the current lane were the vehicle is evolving. As detailed in the article above, the algorithm will define several models to find the points in the different ROIs that are likely to belong to the roadside (multi-points). After giving several mathematical definitions of a lane, the article [11] gives different suggestions to choose in every ROI the point that fulfil a maximum of these hypothesis (multi-hypothesis). Thus, every ROI is composed by several points that try to describe the position of the lane. Finally, a model of lane can be fitted to the most probable points. This model can be linear, polynomial or can use splines or another description. The figure 4 plots the results for 6 ROIs in a left curve. This figure shows that the algorithm does not need any lane marker as white line, and choose the most probable lane depending on vehicle's trajectory. However, shadows can influence precision of algorithm, but multi-points strategy tries to minimize such bias. Time performance of such algorithm is about 3ms on the initial revision of the vision platform.

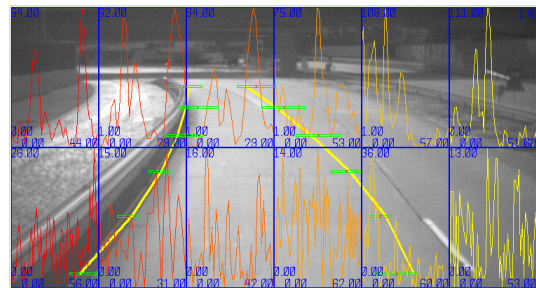


Fig. 4. Results of lane detection of a left curve with unpainted white line, computation is done on six small rectangular ROIs (in green) for the left line, and 6 others for the right line. The profile of the gradients of the ROI are superimposed on the picture, where the first line (6 big rectangles) deliver the profile for the left lane beginning with the bottom ROI. Printed in the top right, the processing time of the image is 1.6ms.

³The **Region Of Interest** will be denoted by the ROI acronym.

IV. THE VIRTUAL DRIVER

The virtual driver has to take 3 elements into account in order to provide an exhaustive list of safe motion vectors : the driving rules, the physical limits due to the road curvature and the dynamic of the vehicle. First of all the theoretical possibilities depending on the driving rules and the environment will be computed. After that the physical limit due to the road is applied in order to keep the vehicle into a safe state. Finally the possibilities will be fused with the vehicle capacities in order to be sure that the vehicle could realize what the system has computed.

A. Definition of the scenarios

As long as no crossing roads are detected, a pattern (shown on figure IV-A) for the straight lane is used. In this case, they are 8 standard regions of interest around the vehicle that lead to the use of some controllers : vehicle following, speed control, overtaking, lane insertion, emergency brake etc. The pattern is matched on the environment and enables all the physically allowed scenarios.

If the presence of an object like a pedestrian could not

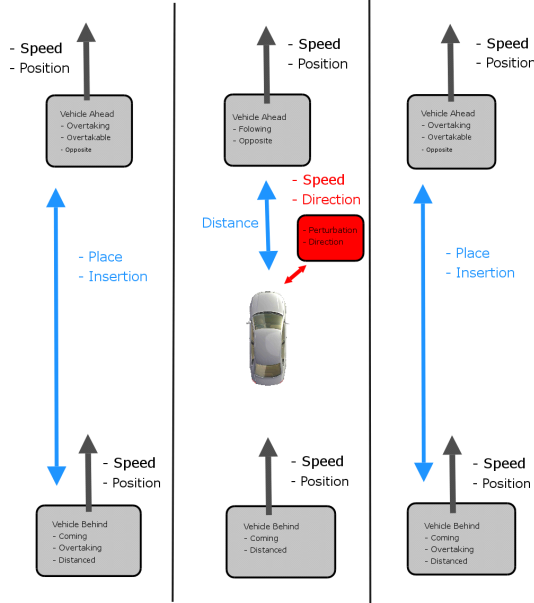


Fig. 5. Standard pattern for the straight lanes
The vehicle is in the centre of the map, in grey are defined the possible position of objects, which have influence of the choice of scenarios. In red is defined an unforeseen element on the road that must be avoided.

be set to one of the region of interest, it will be defined as unforeseen object (in red on the figure IV-A) and the vehicle will try to avoid this object by going in the opposite direction.

B. Modelling a scenario

Each scenario is defined as an agent with some properties :

- A validity range: (E^{**}) , sub-part of the model of the environment (E^*) . This range is the union of obligatory elements and the exclusion of additional parts.
- A longitudinal and a lateral controller: $M_{long.}$, $M_{lat.}$. Their outputs are a range for the acceleration (γ) and for the steering (θ) . An optimum of the scenario will be set with its quality or dangerousness (Q) . The quality is a number between 0 and 255. If the number is bigger than 100, the scenario will not provoke any kind of accident.

$$M : \begin{cases} M_{long.} : E^{**} \longrightarrow \begin{cases} \gamma_{min}, \gamma_{max} \in \mathbb{R} \\ \gamma_{opt} \in \mathbb{R} \end{cases} \\ M_{lat.} : E^{**} \longrightarrow \begin{cases} \theta_{min}, \theta_{max} \in \mathbb{R} \\ \theta_{opt} \in \mathbb{R} \end{cases} \end{cases} \quad (1)$$

For both acceleration and steering, the system collects the three given points and fits a Gaussian curve (like on figure IV-B) with them in order to extract a quality curve. At this moment a quality for each acceleration or steering is defined. The best quality is given for the optimum and out of the range the quality is defined as null by default. The longitudinal and lateral curves fusion together in order to deliver a 2D scenario description like on figure IV-B.

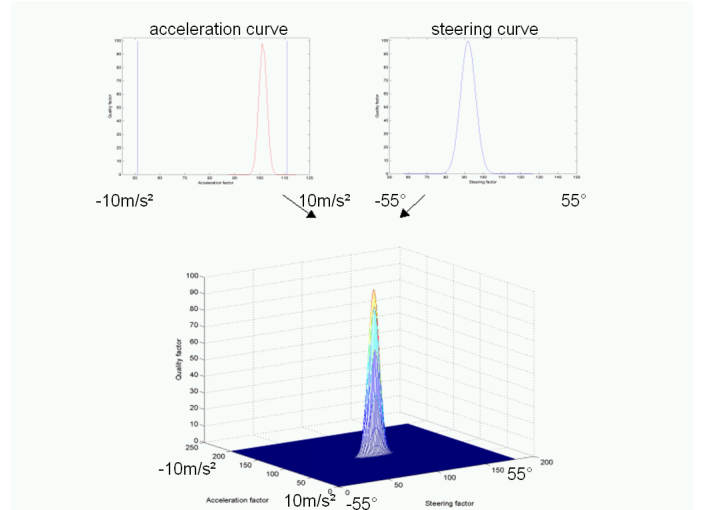


Fig. 6. Example of the fusion of two curves to generate a scenario
Longitudinal control (acceleration) : Adaptive Cruise Control
Lateral control (steering) : Lane following

C. Effect of the road's curvature on the speed

The maximal speed depends not only on shields but also on the curvature of the road. The speed of the vehicle V and the curve radius R are linked with the centrifugal force $F_y = \frac{v^2}{R}$. If the vehicle is too fast in a curve, the driver will have to brake preventively in order to reduce the centrifugal force. This force is only counterbalanced with the tires' friction capacity : the friction coefficient of the road μ -friction, the slip

and the quality of the tires. The preventive braking assumes that the slip will not change abruptly on the next meters, and can compute the maximal speed into the curve ahead the vehicle. An algorithm for the degradation of the speed can also compute the maximal speed on the current position in order to be slower than this maximal speed at each moment.

D. Dynamic of the vehicle

The virtual driver wants to create some theoretical scenarios, but it has to take the vehicle capacities into account. The braking unit, the engine and the steering unit send back to the virtual driver a second order converging transfer function like the equation (2) to explain there current capacities.

$$H(p) = C + \frac{K}{1 + 2\frac{\kappa}{T}p + \frac{1}{T^2}p^2} e^{-j \cdot p / \tau} \quad (2)$$

In this function there are defined the current state (C), the maximal capacity (K), the dead time (τ) and the maximal rate. These values permit to know the maximal steering, braking and acceleration capacities and the delays. Therefore they could downsize the motion vectors map and may disable some extreme scenarios that could not be realised practically. On the figure 7 the theoretic safe rang (red) is downsized to a realistic comfort range, which could be realised in practice.

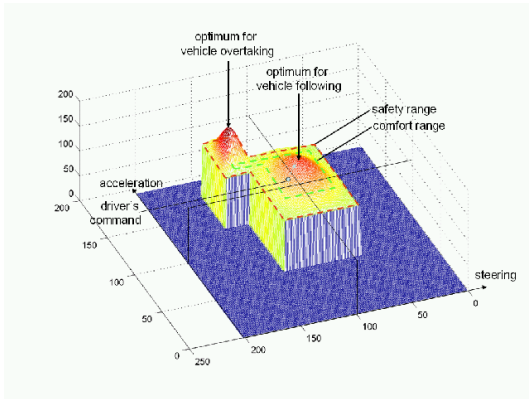


Fig. 7. Definition of the possible dynamic of the vehicle

V. DECISION CONTROL

The decision control has to understand what the driver wants to do and to help him/her to improve his/her driving. If the driver's command tends to be dangerous, it will be overwritten by the decision control. It uses the motion vectors map coming from the virtual driver and matches the driver's command on the map to know the quality of the driver's command and the topology of the scenario. The best driving optimum corresponding to the scenario chosen by the driver could be extracted and used to send a feedback to the driver.

A. Finding and tracking the driving optimum

The optimums are founded by the attraction of several particle filters. The position of these particles are defined on the centre of the different regions of interest. A region of interest is an area within important variations of the qualities of the motion vectors.

The system makes a spatial derivation of the map with the Sobel's masks [12]. A threshold is dynamically computed by using the histogram of the derived map. At this moment there is only one optimum per area.

During the second step, the system looks at the motion vectors into those boundaries in order to find the optimums. A Newton's searching method is used to find the optimum of these areas defined by the founded contour. The starting point of the particle filters is the geometric centre of each area, where the a-prior probability is the highest. After that the area is cleared and the algorithm is checking the next contour.

It is not necessary to extract the optimums at each computation time steps. As long as the virtual driver does not change the number of enabled scenarios, the number of optimums stays the same. The decision control just tracks the optimums because it is more than 8 times faster. The positions of the optimums are searched with the same Newton's method. The position of the starting point is the hypothetical new position of the optimum based on extrapolated displacement after the former position.

B. Setting the current scenario

On normal situation, there is always more than one possible scenario. Therefore the system has to find which scenario is chosen by the driver. The optimum of the choosen scenario is supposed to be near from the driver's command. The different positions of the driver's command give an extrapolated lane over the time, which is supposed to tends to the optimum. The closest optimum from the extrapolated lane is chosen by default for the first estimation. If the chosen scenario is not the right one, the driver will not take the feedback into account. Therefore the decision control has to monitor the reaction and try another optimum until it finds the right one. The monitoring analyses the direction of the variations of the driver's command and if they are attracted or not by the new choosen optimum.

There are some standard cases where this problem could occur. If the driver wants to overtake without blinking. In that case the system cannot know, that the driver wants to overtake. Therefore the system supposes, that the driver wants to stay on the current lane. In this case the given feedback to the driver is supposed to help the driver to stay on his/her lane. As the driver wants to change the lane, he/she will counteract and does not change his/her mind. At this moment, the system will understand its mistake and choose the next scenario: the lane changing. As the feedback will be accepted by the driver, the system will be sure about its choice.

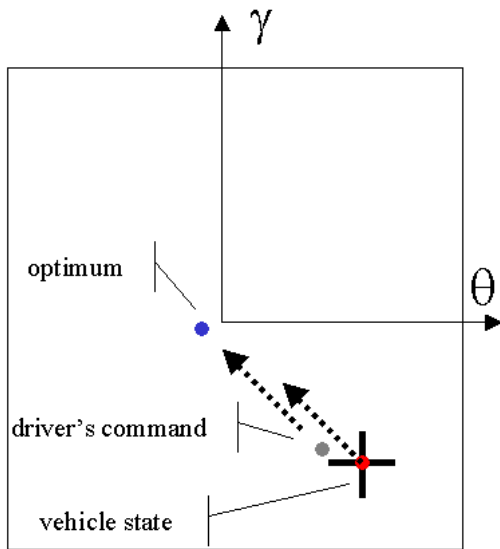


Fig. 8. Feedback dispatching

If there is an emergency, there will only be one scenario validated. In that case, if the driver does not react quickly, it will still try to keep a former scenario that will no more be validated. However the decision control will already deal with the single safe scenario and not accept the mistake of the driver.

C. Generating an adequate feedback

The feedback to the driver mixes two physical information :

- The current motion vector of the vehicle. This information comes from the powertrain controller and explain the real acceleration and the steering angle. In normal state, the current motion vector is quasi the driver's command.
- The difference with the optimum. This is practically the distance (acceleration - steering angle) between the driver's command and the optimum. This difference is represented as stimulation going in the direction of the optimum.

The figure 8 represents the driving on ice in a left-hand curve. The vehicle starts to slide unstable on the left: there is a small difference between the driver's command and the vehicle's state. Because of lack of practice the driver tries to oversteer more and more on the left and to brake, because he/she supposes that oversteering and braking is the right answer to his/her problem. But the optimum goes in the opposite direction: (no braking - lower steering) in order to have the control of the vehicle back. With this kind of feedback, the driver will understand easier what to do in order to keep the vehicle in a safe state. Otherwise he/she will change too late his/her mind and sometimes causing an accident.

VI. CONCLUSION

First a new concept of active safety system for the vehicles was described. The idea is to make redundant the complete link between the environment sensing to the vehicle control. In that case, a virtual driver, that has some knowledge about the driving, will be used here as redundancy if the driver does not react correctly to an emergency situation. Actually the vehicle could not drive autonomously but it can takes the command at the last moment in order to stay in a safe state.

The virtual driver uses an environment model created by the fusion of the data coming from radars, camera and telematic. The virtual driver creates a motion vectors map (acceleration - steering) within a quality for each possible motion vector depending on the different strategies.

The driver's command will be compared with the virtual driver's output at each time. With this method, the quality of the driving can be computed and a feedback to the driver could be send to improve the driving. In case of emergency, the driver may not react correctly to a problem even if the feedback try to help him, because of the lack of time. In that case, at the last moment, the system will help the driver by modifying the command in order to stay in a safe state.

REFERENCES

- [1] J. Grendel, "Sicherheitsanalyse im strassenverkehr," tech. rep., Bundesanstalt fr Strassenwesen, Mensch und Sicherheit, 1993.
- [2] A. K. R. Nathan Gartner, Carrol J. Messer and R. J. Koppa, *Traffic Flow Theory*. 1975.
- [3] S. Fleury, M. Herrb, and R. Chatila, "Genom: A tool for the specification and the implementation of operating modules in a distributed robot architecture," in *International Conference on Intelligent Robots and Systems*, vol. 2, (Grenoble (France)), pp. 842-848, IEEE, Sept. 1997.
- [4] A. G. S. Goerzig and P. Levi, "Realzeitfaehige multiagentenarchitektur fuer autonome fahrzeuge," pp. 44-55, Nov. 1999.
- [5] M. Bertozzi and A. Broggi, "Gold: a parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE Transactions on Image Processing*, vol. 7, no. 1, pp. 62-81, 1997.
- [6] M. Bertozzi, A. Broggi, G. Conte, and A. Fascioli, "Obstacle and lane detection on the argo autonomous vehicle," *proceedings of IEEE Intelligent Transportation Systems conference*, 1997. Boston, MA.
- [7] D. Pomerleau and T. Jochem, "Rapidly adapting machine vision for autonomated vehicle steering," *IEEE Expert*, vol. 11, pp. 19-27, 1996.
- [8] B. Ma, S. Lakshmanan, and A. O. Hero, "Simultaneous detection of lane and pavement boundaries using model based multi-sensor fusion," *IEEE Transactions on Intelligent Transportations Systems*, vol. 1, no. 3, pp. 135-147, 2000.
- [9] D. Dedieu, V. Cadenat, and P. Soueres, "Mixed camera-laser based control for mobile robot navigation," *IEEE International Conference on Intelligent Robots and Systems*, pp. 1081-1086, 2000.
- [10] J. A. Castellanos, J. Neira, and D. Tardos, "Multisensor fusion for simultaneous localisation and map building," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 908-914, 2001.
- [11] M. Bellino, Y. Lopez de Meneses, P. Ryser, and J. Jacot, "Lane detection algorithm for an onboard camera," *SPIE proceedings of the first Workshop on Photonics in the Automobile*, 2004.
- [12] A. Rosenfeld and A. Kak, *Digital Picture Processing*. Academic Press, Orlando, Fl., 1982.