

Vision Based Control in Driving Assistance of Agricultural Vehicles

D. Khadraoui*, C. Debain**, R. Rouveure***,
P. Martinet*, P. Bonton* and J. Gallice*

* *LASMEA, Université Blaise Pascal UMR 6602 du CNRS,
63177 Aubière cedex, France*

** *CEMAGREF, Division Techniques du Machinisme Agricole,
03150 Montoldre, France*

*** *CEMAGREF, Division Electronique et Intelligence Artificielle,
BP.121, 92185 Antony, France*

Abstract

This article presents a real-time control system for an agricultural mobile machine (vehicle) based on an on-board vision system using a single camera. This system has been designed in order to help human beings in repetitive and difficult tasks in the agricultural domain. The aim of the robotics application concerns the control of the vehicle with regard to the reap limit detected in image space. The perception aspect in relation with the application has been described in previous work, and here we deal with the control aspect. We integrate image features issued from the modelling of the scene in the control loop, in order to perform image-based servoing technique. The vehicle behaviour described here concerns bicycle and neural models, and three control laws are then synthesized. The first and the second are modelling approaches and use an interaction between the scene and the image space. They are based on regulation of a task function. The third is a “black-box modelling” technique and is based on a neural network. Finally, experimental results obtained with these different control laws in different conditions are presented and discussed.

Keywords : Agricultural mobile vehicle, Vision-based linear control, Interaction matrix, Neural network, Camera.

Contents

1	Introduction	4
2	Task Function Approach	6
2.1	Modelling of the Scene	6
2.2	Vehicle Modelling	7
2.3	First Law : State Space Design	9
2.4	Second Law : Regulation of two Task Functions	11
3	Neural Approach	14
3.1	Neural Model	14
3.2	Choice of Neural Network	15
3.3	Training the Network	16
4	Experimental Results	17
4.1	Implementation Considerations	17
4.2	Results using the first Law (state space design)	19
4.3	Results using the second Law (regulation of two task functions)	20
4.4	Results using the third Law (neural approach)	22
5	Conclusion	25
6	References	27

List of Figures

1	Parts of the servoing chain	5
2	3D and 2D camera frames	7
3	Vehicle kinematic modelling	8
4	Principle of neural network training	15
5	Data storage (step 1) and learning set creation (step 2)	15
6	Choice of neural network architecture	16
7	Creation of training data	16
8	3D data training	17
9	Test procedure	18
10	Parallel architecture	19
11	First law: test at 4 km/h with adapted parameters ξ and ω_0	20
12	First law : test at 4 km/h with 16 pixels of noise in ρ	21
13	First law : test at different velocities	21
14	Second law : test at 4 km/h with adapted parameters λ and β	22
15	Second law : test at 4 km/h with 16 pixels of noise in ρ	23
16	Second law : test at different velocities	23
17	Third law : test at 4 km/h	24
18	Third law : test at 4 km/h with 16 pixels of noise in ρ	24
19	Third law : test at different velocities	25
20	Results with image processing algorithms	27

1 Introduction

The domain of agricultural work is in general extremely complex. Guiding an agricultural machine has more in common with driving a cross-country vehicle than with steering a high speed vehicle along a motorway, notably because of the nature of agricultural terrain. If we wish to cover all agricultural situations which are today managed by human pilots, the guidance system must be particularly robust in order to adapt to the variety of natural situations encountered. Indeed, the extreme variety of the disturbances which could modify the behaviour of the vehicle is unknown in advance, and makes difficult and delicate the development and synthesis of servoing systems. Among these disturbances, we may note the problems of interaction between the wheels and the soil (sliding with wheels locked, wheel-spin and skidding), the slopes, and the intrinsic features of the vehicle. The appearance and disappearance of these disturbances are not foreseeable and they could evolve in time and depend on modifications of the climatic conditions and the vehicle mass.

Today, the driving of mobile agricultural vehicles is in general entirely done by man. Steering control requires precision and dexterity and can require the driver's full attention in complex situations. It is tiring for the driver to keep focused on this single task, causing losses in cost and work quality. Automatic control of speed and steering allows the human, present on the vehicle, to optimize use of the machine and the current agricultural task. His attention for driving is only necessary in exceptional situations. The development of a guidance system in the agricultural domain can also be conditioned by other criteria such as problems of cost and ease of implementation and utilization. The technological choices must therefore remain realistic in order to interest a manufacturer of agricultural machinery. Furthermore, the harsh conditions of agricultural work render this domain particularly demanding with regard to the equipment.

A significant amount of research work has been done in the domain of agricultural robotics. This subject covers a wide range of potential applications, from the inspection and/or picking of fruit or vegetables to the concept of an autonomous robot working in the agricultural environment. Computer vision can therefore be of significant value in order to automatically (or semi-automatically) drive a robotic apparatus which aims to replace human beings in repetitive or hard tasks in a natural environment (Jarvis 1990; Sandini et al. 1990; Derras et al. 1991; Amat et al. 1993; Casals et al. 1993). Today, techniques of visual servoing are used to control robot manipulators (Chaumette 1990; Espiau et al. 1992; Khadraoui et al. 1996) but there are still few applications in mobile robotics (Pissard-Gibollet and Rives 1991; Jurie et al. 1994; Khadraoui et al. 1995). For a mobile robot, the main problem in using these techniques is due to the presence of non-holonomic mechanical connections which limit robot movements. In

this context, traditional visual control laws are in general synthesized by separating the vision aspect from the control module.

Classical methods, which are implemented for agricultural vehicle guidance are based on vision in 3D space (Derras et al. 1993; Debain et al. 1994; Klassen et al. 1994). But with the vision-based control technique, the philosophy consists in reaching a particular configuration in the 2D image plane and not a situation between camera and object. So this approach has the advantage of avoiding the intermediary step of 3D estimation of the environment with regard to the robot. Problems related to the reconstruction of the 3D world are then eliminated and no explicit calibration of the camera is needed.

The studies presented in this paper were carried out on a mobile agricultural vehicle which has the steering wheels placed at the rear.

We present three methods of vision-based control integrating visual features directly into the control loop like represented in figure 1 :

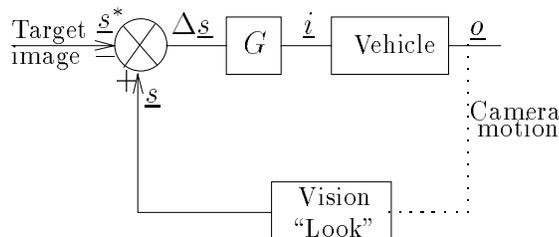


Figure 1: Parts of the servoing chain

- \underline{s}^* is considered as a reference target image to be reached in the image frame,
- \underline{s} is the value of the visual information currently observed by the camera and computed by image processing,
- G is the vector control gain to be synthesized by the task function approach or the neural approach,
- \underline{i} is the control variable of the vehicle representing the steering angle of the machine,
- \underline{o} is the set of outputs characterizing the machine's position and orientation.

Two of the control laws are modelling approaches and based on a task function approach. The third is a “black-box modelling” approach and is based on neural networks. The main point of this paper is the presentation of experimental results using three different techniques of control.

2 Task Function Approach

We adopt the task function approach developed by *Espiau et al.* (Samson et al. 1991; Espiau et al. 1992). This function can be thought of as representing a virtual kinematic constraint between the camera and the target. In this case, we explicitly use modellings of the scene and the vehicle in order to synthesize the control laws.

2.1 Modelling of the Scene

The modelling of the scene consists of the development of an interaction matrix which enables the system to relate the visual information to the different control inputs (Chaumette 1990). We develop the problem considering that the camera used is modelled by the classical *pinhole* approximation.

Knowing the camera velocity screw T_c defined by three translational and three rotational velocities $\underline{V} = (V_x, V_y, V_z)$ and $\underline{\Omega} = (\Omega_x, \Omega_y, \Omega_z)$, we can relate it by means of :

$$\dot{\underline{s}} = L_{\underline{s}}^T T_c \quad (1)$$

where $L_{\underline{s}}^T$ is the interaction matrix related to situation \underline{s} (Chaumette 1990; Hager et al. 1996). The expression of $L_{\underline{s}}^T$ depends on the nature of the visual information contained in \underline{s} and is obtained by using the well known equation of optical flow measurement to 3D structure and motion in the scene (Paul 1982; Faugeras 1993) .

The scene is represented by a straight line for which we find an equation in the image frame of the camera. We express the position of the machine and its orientation according to the (θ, ρ) parameters of the line measured and we define the set of visual features :

$$\underline{s} = (\theta, \rho)^T \quad (2)$$

The problem then consists in defining the interaction matrix related to the desired position $\underline{s} = \underline{s}^*$. This represents the relation between the variation of the visual features and the velocity screw which defines the set of machine displacements at the equilibrium position. From (1), we have :

$$\dot{\underline{s}} = L_{[\underline{s}=\underline{s}^*]}^T T_c \quad (3)$$

The equation of the plane containing the desired line and according to the camera frame is expressed as :

$$y \cos \alpha + z \sin \alpha + h = 0 \quad (4)$$

where α represents the angle of inclination of the camera (see Figure 2(a)).

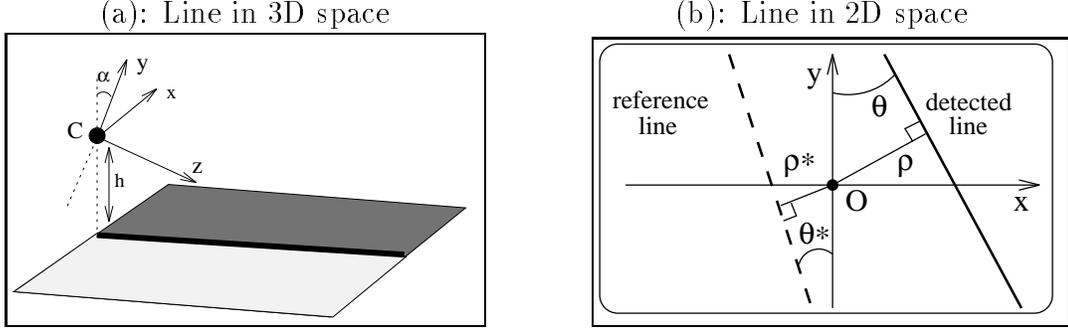


Figure 2: 3D and 2D camera frames

Considering expression (1) and taking the desired features, such as $\theta = \theta^*$ and $\rho = \rho^*$, then the interaction matrix at the equilibrium situation is given by :

$$L_{\underline{s}=\underline{s}^*}^T = \begin{pmatrix} \lambda_\theta^* \cos \theta^* & \lambda_\theta^* \sin \theta^* & -\lambda_\theta^* \rho^* & -\rho^* \cos \theta^* & -\rho^* \sin \theta^* & -1 \\ \lambda_\rho^* \cos \theta^* & \lambda_\rho^* \sin \theta^* & -\lambda_\rho^* \rho^* & (1 + \rho^{*2}) \sin \theta^* & -(1 + \rho^{*2}) \cos \theta^* & 0 \end{pmatrix} \quad (5)$$

with :

$$\begin{cases} \lambda_\theta^* &= -(\cos \alpha \cos \theta^*)/h \\ \lambda_\rho^* &= (\rho^* \cos \alpha \sin \theta^* + \sin \alpha)/h \end{cases}$$

In the next section we present the vehicle modelling.

2.2 Vehicle Modelling

We use a machine which has two steerable wheels and two driven wheels (like a car). At a constant steering angle, this machine describes a circle. In order to model the displacement of the machine, we consider a bicycle model which assumes the following :

- there are no flexible parts,
- the vehicle moves on plane surface,
- there is no translational slip between the wheels and the surface,
- there is sufficient rotational friction between the wheels and the surface.

We establish the general equations relative to its behaviour taking into account its kinematic characteristics. It is useful to approximate the kinematics of the steering mechanism by assuming that the two rear wheels turn slightly differentially. Then, the instantaneous centre of rotation can be determined purely by kinematic means. This amounts to assuming that the

steering mechanism is the same as that of a bicycle. Let the angular velocity vector directed along the y axis be $\dot{\psi}$ and let the linear velocity directed along x axis be \dot{x} .

Using the bicycle model approximation (see Figure 3(a)), the steering angle δ and the radius of curvature r are related to the wheelbase L , as in (Kelly 1994) by:

$$\tan \delta = -\frac{L}{r} \quad (6)$$

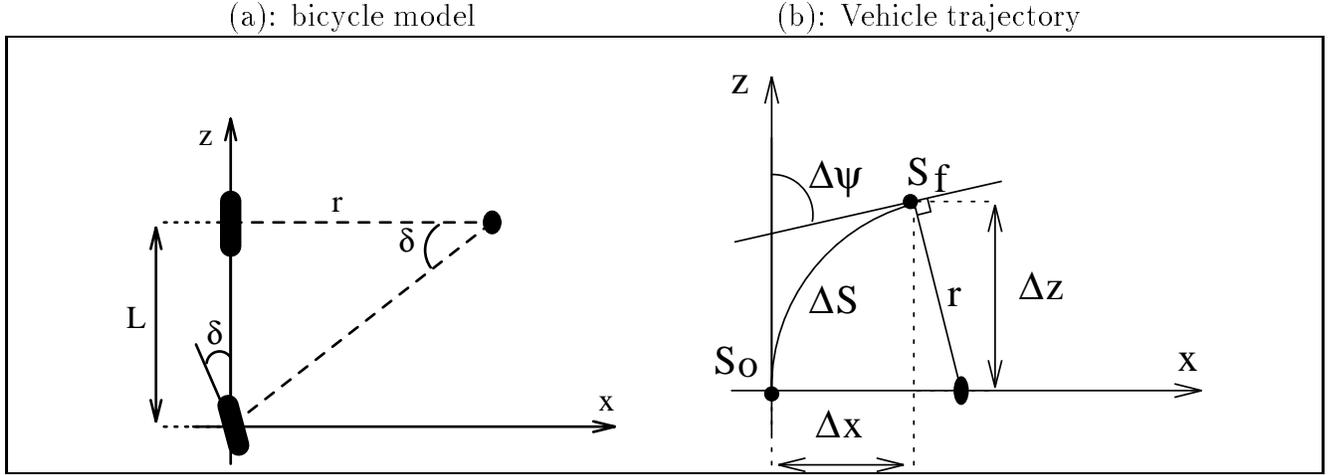


Figure 3: Vehicle kinematic modelling

In Figure 3(b) we show a small portion of a circle ΔS representing the trajectory to be followed by the machine. We assume that it moves with small displacements between an initial curvilinear abscissa S_0 and a final one named S_f such that:

$$\begin{cases} \frac{1}{r} = \lim_{\Delta s \rightarrow 0} \frac{\Delta \psi}{\Delta S} = \frac{d\psi}{dS} = \frac{\dot{\psi}}{\dot{S}} \\ \dot{S}^2 = \dot{x}^2 + \dot{z}^2 \end{cases} \quad (7)$$

In fact, the rotational velocity is obtained as:

$$\dot{\psi} = -\frac{\tan \delta}{L} \sqrt{\dot{x}^2 + \dot{z}^2} \quad (8)$$

The lateral position denoted by x can be computed by assuming that the machine moves with small displacements. In the case of longitudinal motion along the z axis during a time interval Δt , the machine moves through distance Δz (see Figure 3).

We express:

$$\begin{cases} \Delta z = r \sin \Delta \psi \\ \Delta x = r(1 - \cos \Delta \psi) \end{cases} \quad (9)$$

Eliminating r from equations (9), we obtain:

$$\Delta x = \Delta z \frac{1 - \cos \Delta\psi}{\sin \Delta\psi} \quad (10)$$

Without loss of generality we can consider that the initial conditions are null since the frame is linked at the position S_0 and then $\Delta x = x$, $\Delta z = z$ and $\Delta\psi = \psi$. We compute the derivative over time of the lateral coordinate x of the machine given by (10) which depends on z and ψ , as follows:

$$\dot{x} = \frac{1 - \cos \psi}{\sin \psi} \left[\dot{z} + \frac{z\dot{\psi}}{\sin \psi} \right] \quad (11)$$

The approximation to small angles (ψ and δ are less than 7°) is valid in the case of our application and enables us to simplify equations (8) and (11). This gives the relation between the differential of the lateral coordinate x and the lateral deviation ψ with the steering angle δ by expressing the development of trigonometric equations to the second order. If we consider that the machine moves with constant longitudinal speed $\dot{z} = V$ and that $\dot{x} \ll V$, we can write:

$$\dot{x} = \frac{\psi}{2} \left(\dot{z} + \frac{z}{\psi} \dot{\psi} \right) \quad (12)$$

Taking into account the approximations below, we have $\dot{\psi} \approx \frac{\dot{z}}{r}$ and then $\psi \approx \frac{z}{r}$ since the r is constant and the initial conditions are null (frame fixed at initial position of the robot). We finally find the kinematic model of the machine expressed by the following equations which are similar to those obtained by another method (Khadraoui et al. 1995):

$$\begin{cases} \dot{\psi} = -\frac{V}{L}\delta \\ \dot{x} = V\psi \end{cases} \quad (13)$$

However, we only control the wheel angle δ . We can find an equation linking δ to the vector $(V, \dot{\psi})$. In our application we consider that the velocity V is constant, when we have (Ioos et al. 1988; Murray and Sastry 1993) :

$$\delta = -\frac{L}{V}\dot{\psi} \quad (14)$$

2.3 First Law : State Space Design

We treat here a single input linear system with output parameters (θ, ρ) (see Figure 1). To control such a model, a technique of pole assignment is used.

Here, we elaborate a state model which integrates both the model of the machine and that of the scene (Khadraoui 1996). In the context of our application, we assimilate the machine to a particular mobile robot which moves with limited degrees of freedom. It has non-holonomic

constraints since the number of degrees of freedom of control, δ in our case, is less than the number of degrees of freedom of displacement : translation along x and z axes and rotation around the y axis. In the case where the desired situation is represented by a straight line centered in the image ($\theta^* = \rho^* = 0$) and by considering equation (3), we can express the kinematic screw T_c as :

$$T_c = L_{|\underline{s}=\underline{s}^*}^{T+} \dot{\underline{s}} \quad (15)$$

Determination of the kernel of the interaction matrix (matrix of rank 2), permits us to constrain the lateral translation of the vehicle and its orientation by introducing a rank 2 matrix such as:

$$T_c = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} L_{|\underline{s}=\underline{s}^*}^{T+} \dot{\underline{s}} \quad (16)$$

Consequently, the matrix $L_{|\underline{s}=\underline{s}^*}^{T+}$ is limited to the components that correspond to the lateral and the orientation movements. In fact, the interaction matrix is reduced to:

$$\dot{\underline{s}} = \begin{bmatrix} l_{11} & l_{12} \\ l_{21} & l_{22} \end{bmatrix} \begin{pmatrix} V_x \\ \Omega_y \end{pmatrix} \quad (17)$$

with

$$\begin{cases} l_{11} &= \lambda_\theta^* \cos \theta^* \\ l_{12} &= -\rho^* \sin \theta^* \\ l_{21} &= \lambda_\rho^* \cos \theta^* \\ l_{22} &= -(1 + \rho^{*2}) \cos \theta^* \end{cases}$$

We remark that the velocities V_x and Ω_y correspond to those expressed in (13) for the machine. We write:

$$\begin{cases} V_x = \dot{x} &= V\psi \\ \Omega_y = \dot{\psi} &= -\frac{V}{L}\delta \end{cases} \quad (18)$$

and

$$\begin{pmatrix} \dot{x} \\ \dot{\psi} \end{pmatrix} = \begin{bmatrix} l_{11} & l_{12} \\ l_{21} & l_{22} \end{bmatrix}^{-1} \begin{pmatrix} \dot{\theta} \\ \dot{\rho} \end{pmatrix} \quad (19)$$

By integrating (19) over time we have:

$$\begin{pmatrix} x \\ \psi \end{pmatrix} = \frac{1}{\Delta l} \begin{bmatrix} l_{22} & -l_{12} \\ -l_{21} & l_{11} \end{bmatrix} \begin{pmatrix} \theta \\ \rho \end{pmatrix} + \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} \quad (20)$$

where k_1 and k_2 are the constants of integration, assuming that the matrix is rank 2. Using (18) and (20), we can easily express the velocities V_x and Ω_y :

$$\begin{cases} V_x &= \frac{V}{\Delta l}(-l_{21}\theta + l_{11}\rho) + k_2 V \\ \Omega_y &= -\frac{V}{L}\delta \end{cases} \quad (21)$$

with:

$$\Delta l = l_{11}l_{22} - l_{21}l_{12}$$

By considering equations (17) and (21), we are led to the linear 2D model used for the steering control system. We introduce in the control loop the visual features θ and ρ of the line. After development, the state representation of the system with initial conditions null ($k_2 = 0$ if $\theta^* = 0$ and $\rho^* = 0$) is given by :

$$\begin{pmatrix} \dot{\theta} \\ \dot{\rho} \end{pmatrix} = \frac{V}{\Delta l} \begin{bmatrix} -l_{11}l_{21} & l_{11}^2 \\ -l_{21}^2 & l_{11}l_{21} \end{bmatrix} \begin{pmatrix} \theta \\ \rho \end{pmatrix} - \frac{V}{L} \begin{bmatrix} l_{12} \\ l_{21} \end{bmatrix} \delta \quad (22)$$

The continuous-time state-space form of the model (17) becomes:

$$\dot{\underline{s}} = A \underline{s} + B \delta \quad (23)$$

where:

- \underline{s} is the visual information vector to be computed at each iteration by image processing,
- δ is the control variable to be injected into the system at each step of the servoing task,
- A and B are constant matrices.

The control law is synthesized using a pole assignment technique by assimilating the behaviour of the system to a second order system having ξ as a damping ratio and ω_0 as its frequency. The parameters ξ and ω_0 can be chosen by using an LQ approach, when the control law $\delta = -G \underline{s}$ enables the poles of the closed-loop dynamics matrix to be placed at the desired locations. Finally, the gain matrix is expressed as follows:

$$G = (g_1 \ g_2) \quad (24)$$

with :

$$\begin{cases} g_1 = \frac{L \omega_0}{V^2 \Delta l} (2 V \xi l_{21} - \omega_0 l_{22}) \\ g_2 = -\frac{L \omega_0}{V^2 \Delta l} (2 V \xi l_{11} - \omega_0 l_{12}) \end{cases} \quad (25)$$

2.4 Second Law : Regulation of two Task Functions

In the modelling of the scene, we considered a visual primitive given by the vector \underline{s} . Then we can define a task function \underline{e} such as :

$$\underline{e} = C(\underline{s} - \underline{s}^*) \quad (26)$$

\underline{e} is a six dimensional vector which corresponds to the six degrees of freedom, $\underline{s} - \underline{s}^*$ is a vector of two components corresponding to the visual informations, and C is a matrix of six rows and two columns which is the combination matrix.

The visual information used in our application is as shown in Figure 2(b):

- θ represents the angle of the line detected in the image,
- θ^* represents the angle of the reference line,
- ρ represents the distance with respect to O of the line detected in the image,
- ρ^* represents the distance with respect to O of the line reference.

We can write:

$$C = (C_1, C_2) \tag{27}$$

From equation (26), we have :

$$\underline{e} = C_1(\theta - \theta^*) + C_2(\rho - \rho^*) = \underline{e}_\theta + \underline{e}_\rho \tag{28}$$

Our application focuses on servoing the robot with respect to a straight line, and for this purpose we consider that the line does not move with respect to the camera. This simplifying hypothesis is not satisfied for every trajectory. Nevertheless, we consider that it is realistic for a straight trajectory or for a piecewise-linear trajectory.

Adopting standard practice in vision-based control, we take the kinematic screw, representing the camera movements, as a control law (Samson et al. 1991; Espiau et al. 1992). In the case where the object stays fixed on the camera frame, a possible control law is :

$$T_c = -\lambda \underline{e} \tag{29}$$

We choose a similar control law given by (Debain 1996):

$$T_c = -\lambda(\beta \underline{e}_\theta + \underline{e}_\rho) \tag{30}$$

λ and β two non-zero positive real values.

This amounts to introducing a supplementary gain on one part of the error, the one concerning the heading error. In fact, the different types of drive of a mobile robot mean that we can give different degrees of importance to the lateral error and the heading error.

A precise but abrupt drive requires a lot of energy, favouring lateral error, whereas a smooth, non-precise drive emphasizes heading error. Consequently, we first impose an exponential decay on \underline{e}_θ and another on \underline{e}_ρ by writing:

$$\dot{\underline{e}} = -\lambda(\beta\underline{e}_\theta + \underline{e}_\rho) \quad (31)$$

The task function \underline{e} is then based on an exponential function together with its decay which will be taken to be equal to the smaller of the two terms $\lambda\beta$ and λ . This behaviour imposes a second-order system with the response time and the overshoot conditioned by the λ and β values.

By using the classical methods of the form of vision-based control, we can calculate the control matrix. We differentiate the equation (28) as follows :

$$\dot{\underline{e}} = C\dot{\underline{s}} = C_1\dot{\theta} + C_2\dot{\rho} \quad (32)$$

where λ and β are scalar and are in R^{*+} and with (31), we obtain :

$$\dot{\underline{e}} = -\lambda C \begin{pmatrix} \beta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \theta - \theta^* \\ \rho - \rho^* \end{pmatrix} \quad (33)$$

Finally, (32) and (33) give :

$$C\dot{\underline{s}} = -\lambda CB(\underline{s} - \underline{s}^*) \quad (34)$$

with

$$B = \begin{pmatrix} \beta & 0 \\ 0 & 1 \end{pmatrix}$$

As is given in (Chaumette et al. 1991), a possible value of C in order to satisfy the conditions of convergence near the desired position is

$$C = L_{(\underline{s}=\underline{s}^*)}^{T+} \quad (35)$$

Here, L^{T+} is the pseudo-inverse of the interaction screw calculated for the equilibrium position $\underline{s} = \underline{s}^*$.

$$L_{(\underline{s}=\underline{s}^*)}^{T+} L_{(\underline{s}=\underline{s}^*)}^T = I_6 \quad (36)$$

Near the the equilibrium position, the modelling of the scene gives us the following :

$$\dot{\underline{s}} = L_{(\underline{s}=\underline{s}^*)}^T T_c \quad (37)$$

T_c is the kinematic screw represented by six degrees of freedom of the camera. We can deduce from equations (34), (35) and (37) :

$$T_c = -\lambda L_{(\underline{s}=\underline{s}^*)}^{T+} B(\underline{s} - \underline{s}^*) \quad (38)$$

T_c gives us a velocity control of our camera and having the expression of Ω_y we deduce the value of the control input of our system expressed in the modelling of the vehicle (eqn. 14). It is given by :

$$\delta = -\frac{\lambda L}{V} \left[\beta \frac{\sin \alpha \cos \alpha}{1+h^2} (\theta - \theta^*) + \frac{h^2 + \cos^2 \alpha}{1+h^2} (\rho - \rho^*) \right] \quad (39)$$

3 Neural Approach

3.1 Neural Model

Using the lateral ρ_i and heading θ_i errors defined in the image reference at the equilibrium situation, the steering controller has to compute steering commands to obtain the minimum difference between the vehicle and the line followed. We chose a multilayer neural network because it has been used successfully in multiple areas such as classification, modelling and automation (LeCun 1987). The learning algorithm is the classical gradient method which aims at reducing the following error function (Rumelhart 1986; Abbassi and Sayed 1991):

$$E_{rr} = \sum_{k=1}^n E^k \quad (40)$$

with $E^k = \sum_1^m (\delta d_i^k - \delta r_i^k)^2$, n the number of example, m the number of outputs, δd_i^k the expected output and δr_i^k the real output.

In our application, for an input (ρ_i, θ_i) the associated steering angle δd_i is unknown. This is known as the credit assignment problem. One way to avoid it is to perform neural training looking at the behaviour of an existing controller. In our application, driver behaviour is used to build a steering controller able to react as effectively as the driver. The driver enables the system to know the correct output δd_i associated with an input (ρ_i, θ_i) . The expected output δd_i is equal to the steering angle. In fact, at time ($t = t_i$), the input (ρ_i, θ_i) is computed and the steering angle δd_i is read. The set $(\rho_i, \theta_i, \delta d_i)$ is then used by the back-propagation algorithm in order to modify the synaptic coefficients and minimize the overall error. In Figure 4, the driver is used as an *ideal* steering controller to perform neural network training.

The sets of examples $(\rho_i, \theta_i, \delta d_i)$ are available at a regular rate. With the sampled data, the back-propagation algorithm used in the gradient method can lead to local minima (McInerney et al. 1989; Dayhoff 1990). In order to avoid them, the learning step is divided into two parts (Figure 5) :

- *learning set creation* : sets of examples $(\rho_i, \theta_i, \delta d_i)$ are stored for later use.

- *neural network learning* : general extraction of sets $(\rho_i, \theta_i, \delta d_i)$ enables the system to reduce the local minima problem during training.

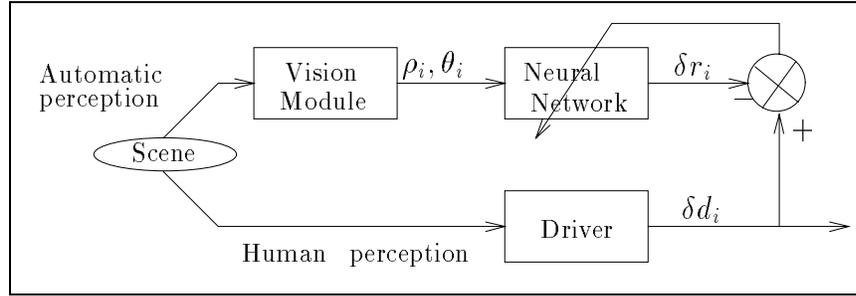


Figure 4: Principle of neural network training

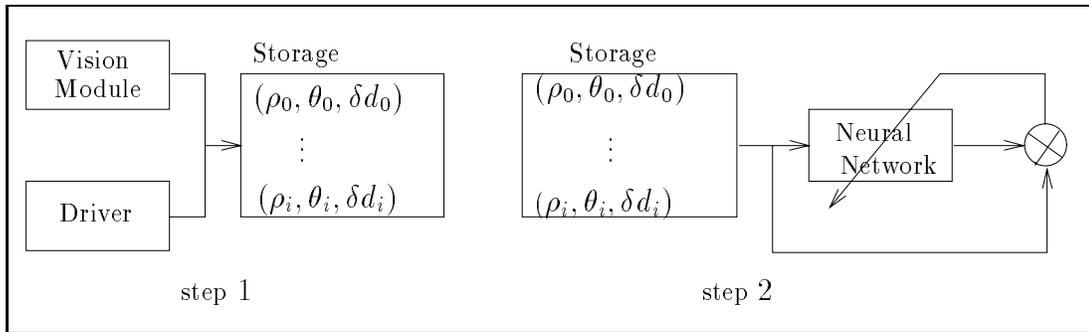


Figure 5: Data storage (step 1) and learning set creation (step 2)

Step 1 is represented in Figure 5, where sets of data $(\rho_i, \theta_i, \delta d_i)$ are stored in a database. In step 2, the learning set created in step 1 is used to perform the off-line neural network training. As there is no real time training, this method is called off-line neural network learning. The neural network can then be introduced into the control loop.

3.2 Choice of Neural Network

We opted for a multi-level gradient back-propagation network. The implementation of this type of network with regular structure is relatively straightforward. We also notice that unlike classification applications, which generally require large networks, our application needs about ten neurons.

The choice of the size of network (number of levels and number of neurons per level) is still a difficult issue, since for a given problem we do not know the optimal network, just as for a given network we do not know the optimal number of neurons per level. The risk is double edged:

- a smaller number of neurons leads to insufficient modelling.
- too many neurons leads to excessive modelling which can reduce the performance of the network.

The final choice of network in our guidance application was found experimentally. We settled on a network with two inputs (heading and lateral position of the line detected in the image), one output (the steering angle) and two hidden levels. The first has two neurons and the second has five neurons (see figure 6).

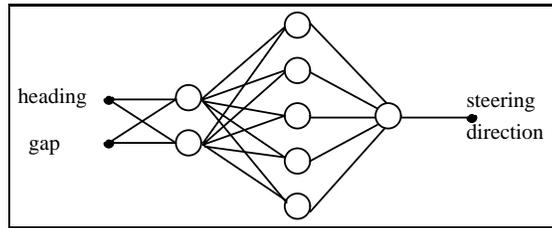


Figure 6: Choice of neural network architecture

3.3 Training the Network

The network learns from examples constituted by the human supervisor model. If the examples come from a domain of reduced functioning of the system, the network will not learn how to respond correctly outside this domain. In the same way, if the supervisor provides erroneous information, the network will learn an erroneous model. In order to create the data from training, a scene composed of two straight lines and one curve is used. The radius of curvature r of this track to follow by the vehicle, defines the maximum domain of controller functioning (see Figure 7).

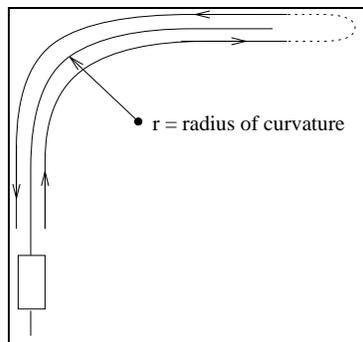


Figure 7: Creation of training data

For some radii of curvature less than r , the neural network will give correct steering control. During this phase of training through manual driving, we record the heading and the position of the line detected in the image and thus the corresponding wheel angle. The training data set created is used to achieve the convergence of the network. This training is not achieved “on line” because the use of regular sets with the algorithm of back-propagation of gradient could fall into a local minimum (McInerney et al. 1989).

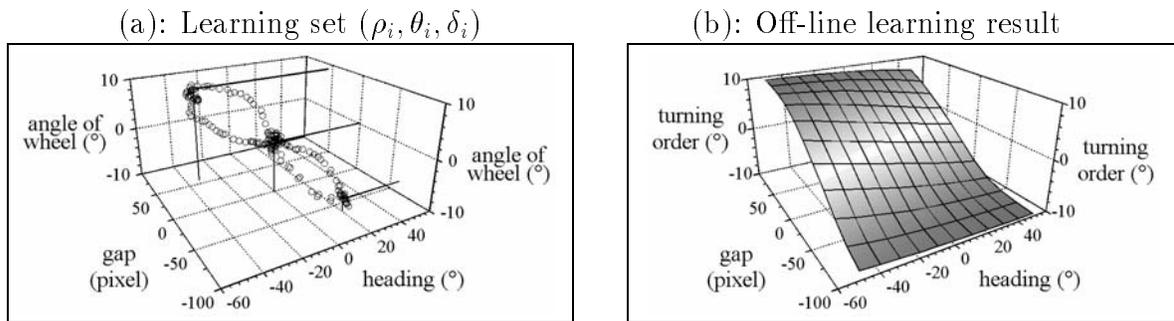


Figure 8: 3D data training

Figure 8(a) represents an example of the learning set. The dots represent a straight line and right and left turn. The beginning and end of turns also appear with different trajectories. This set is then used in order to train the network. Figure 8(b) gives the result of supervised training. The network is able to interpolate the commands in the domain of variation of heading and lateral defined the phase of training.

At the end of the training phase, the neural network can be introduced into the steering control loop. Apart from the problems related to the sensor, the performance obtained is similar to that achieved by a human pilot.

4 Experimental Results

4.1 Implementation Considerations

In order to test our three control laws, we laid out a trajectory composed of a bend, a step of one meter and a straight line as shown in Figure 9(a). A camera was placed on the left side of the cabin (Figure 9(b)). This supplies an image of the white line whose parameters are transmitted to the control algorithm by the image processing task. These parameters are the distance (gap) and the angle (heading) of the line detected (Figure 2(b)).

In order to measure the quality of our control laws, we positioned a second camera in front of the vehicle so that the camera optical axis was perpendicular to the ground (Figure 9(b)).

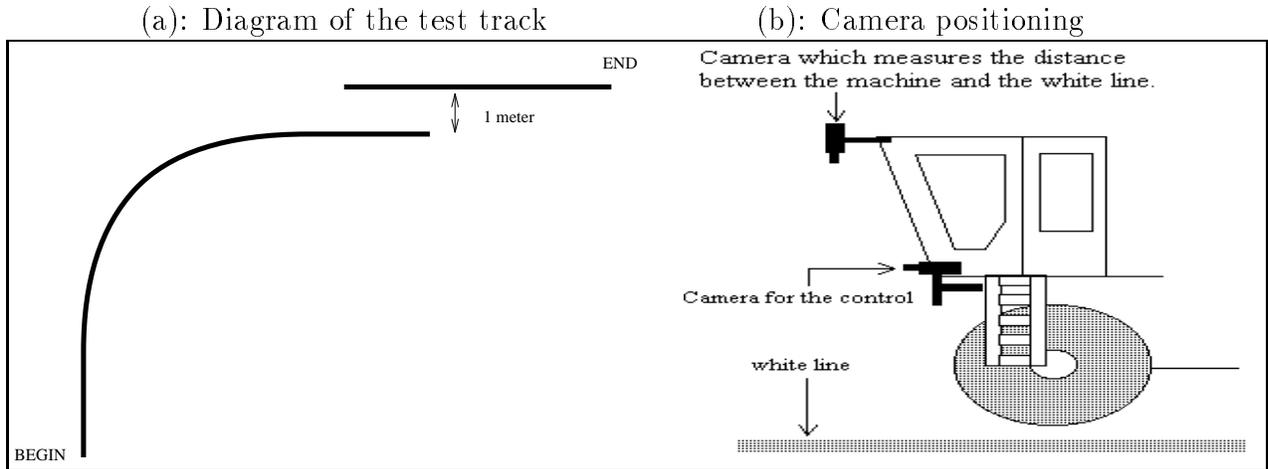


Figure 9: Test procedure

This camera continuously measures the distance between the white line and the machine. A second image processing algorithm is used to enable us to reconstruct the trajectory of the machine. It is calibrated to be able to measure the performance of the guidance system with an accuracy of ± 5 cm.

For each control law, we carried out the same tests under the same conditions. The first test consisted of following the trajectory at several speeds. We then added noise to the visual measurements. The noise corresponds approximately to what can be observed in a natural environment (± 16 pixels in ρ). Most often, it is induced by the terrain which is not perfectly flat like bitumen, or else by the trajectory to be followed.

The test is composed of four parts:

- beginning of bend,
- following the bend (curve with constant turning radius),
- leaving the bend and response to a step of one meter,
- following the straight line.

The servoing task to be accomplished is : first follow a bend with curvature r , then determine the response to a step of 1 meter distance between the two lines, and finally follow the line with good precision. The target is represented by the straight line centered in the image plane and characterized by $\theta^* = \rho^* = 0$.

We employed an architecture composed of two VME microprocessor boards (Figure 10). It is composed of three parts : a low level module, a feature extraction module and a control module. The problem caused by such pipeline architecture consists of the distribution and organization of

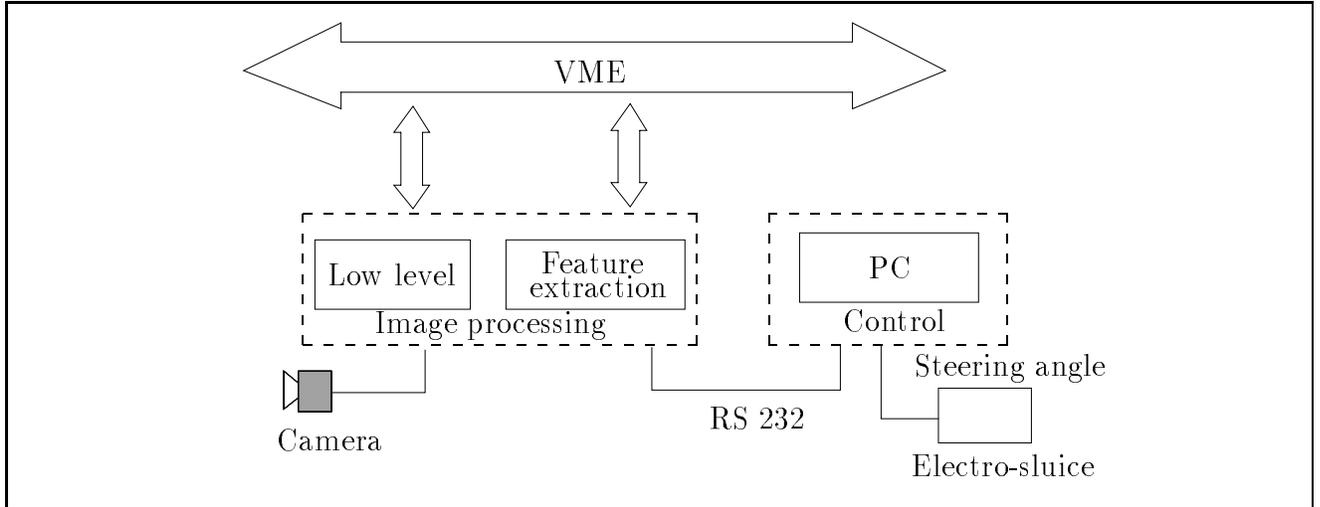


Figure 10: Parallel architecture

all the concurrent tasks which constitute the parallelized algorithm. To manage all the problems arising from the resources shared between the different processors, we chose the OS9 operating system. From the different mechanisms for synchronization and communication between tasks offered by this operating system, we made particular use of signals, communication pipes and data modules. Knowledge of *a priori* information and the parallelized algorithm enables us to achieve real time operation. The image processing results are sent to the personal computer which controls the steering angle. The range period of the image processing calculation time is about 200 ms, which is close to that encountered in a natural environment.

The experimental results obtained by the three control laws under the different conditions will be presented in the following sections.

4.2 Results using the first Law (state space design)

In Figure 11 we present results with parameters adapted as a result of simulations in order to obtain good response to a second order system. We adjust the two parameters ξ and ω_0 with regard to the average speed of the machine which is fixed at 4 km/h . The results concern the input δ and the lateral position x of the machine obtained in real experimentation, and the features errors θ and ρ used as visual informations. The experimental lateral position was reconstructed by using a second camera perpendicular to the scene taking into account the camera calibration. We note that the machine cut across the bend because the camera sees the scene 6.5 m in front of the machine.

$\omega_0(rd/s)$	ξ	$V(km/h)$
0.14	0.9	4

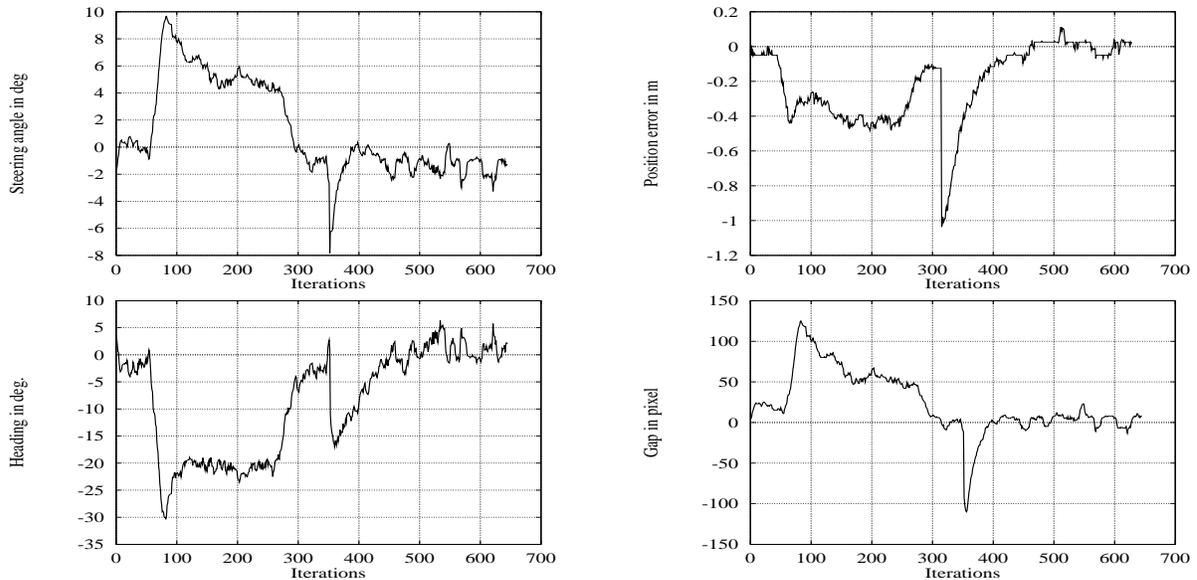


Figure 11: First law: test at 4 km/h with adapted parameters ξ and ω_0

In Figure 12, we tested the robustness of the control law with regard to noise introduced in the measurement parameters ρ and θ . We remark that the task is achieved with good stability and robustness.

Finally, we tested the robustness of the approach to variation in speed. The control is synthesized at a fixed velocity of 4 km/h and the machine is moved at different velocities (4, 6, 8, 10 and 13 km/h). Figure 13 shows that the vehicle reaches the line and continues following it, despite the variation of its speed. But, we remark that increasing speed causes the controlled behaviour to oscillate at steady-state. This is usually an indication that the controller gains are too high.

4.3 Results using the second Law (regulation of two task functions)

The results shown in Figure 14 are obtained at a speed of 4 km/h.

λ	β	$V(km/h)$
0.3	2.6	4

The response time is about 20 s with no overshooting. There are some perturbations on the curves because the camera which measures the distance between the vehicle and the white line

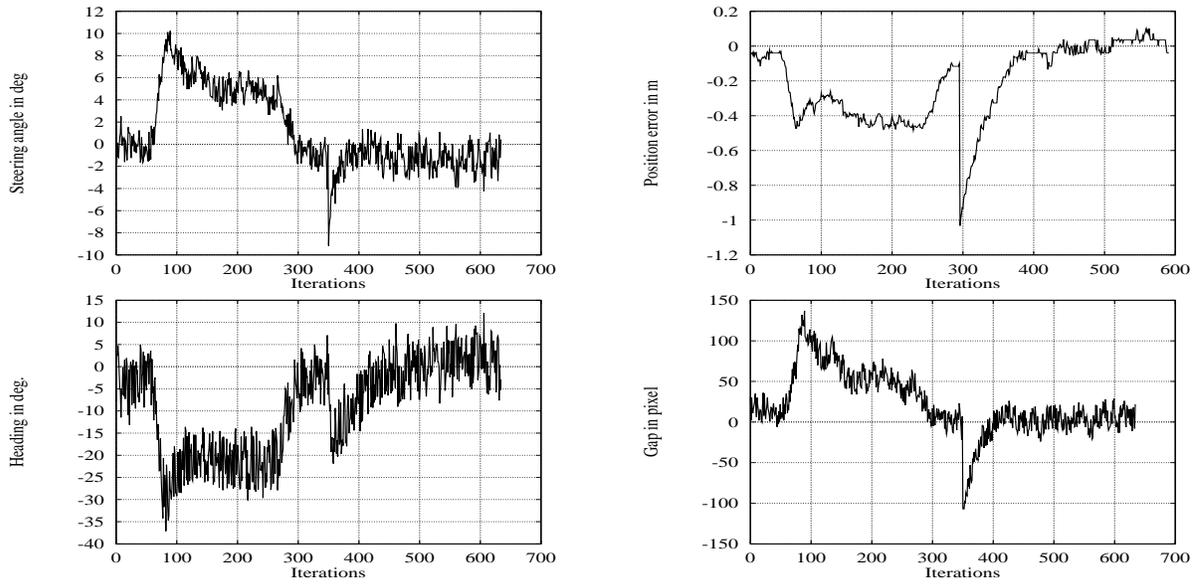


Figure 12: First law : test at 4 km/h with 16 pixels of noise in ρ

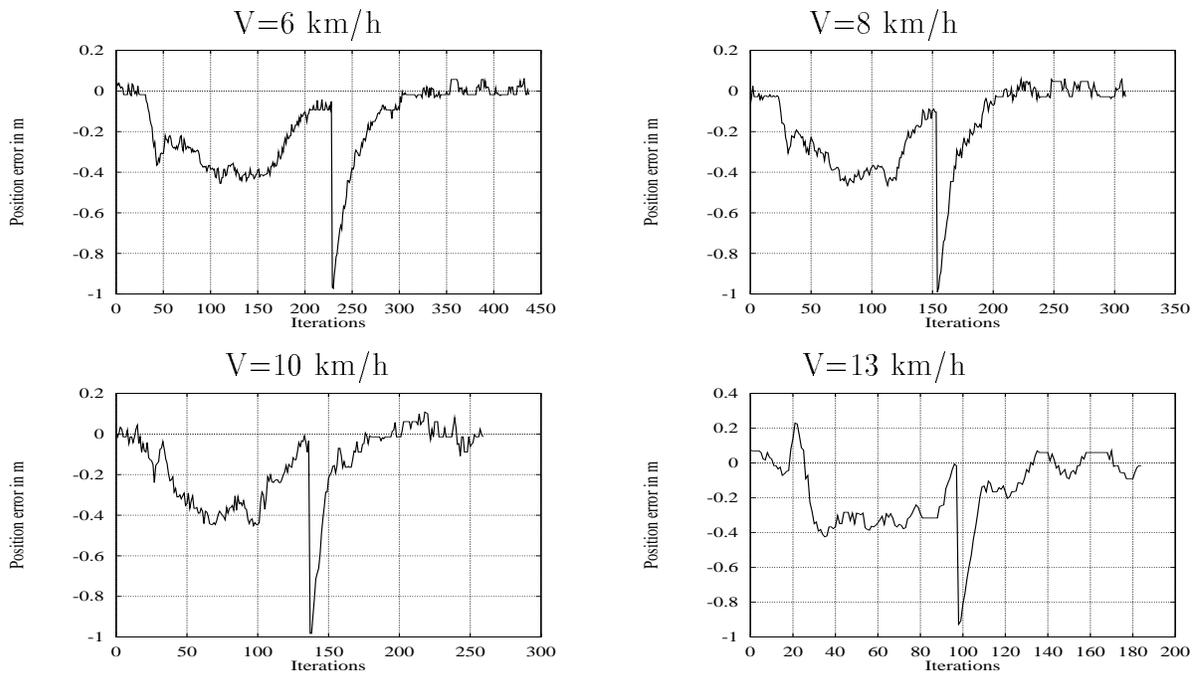


Figure 13: First law : test at different velocities

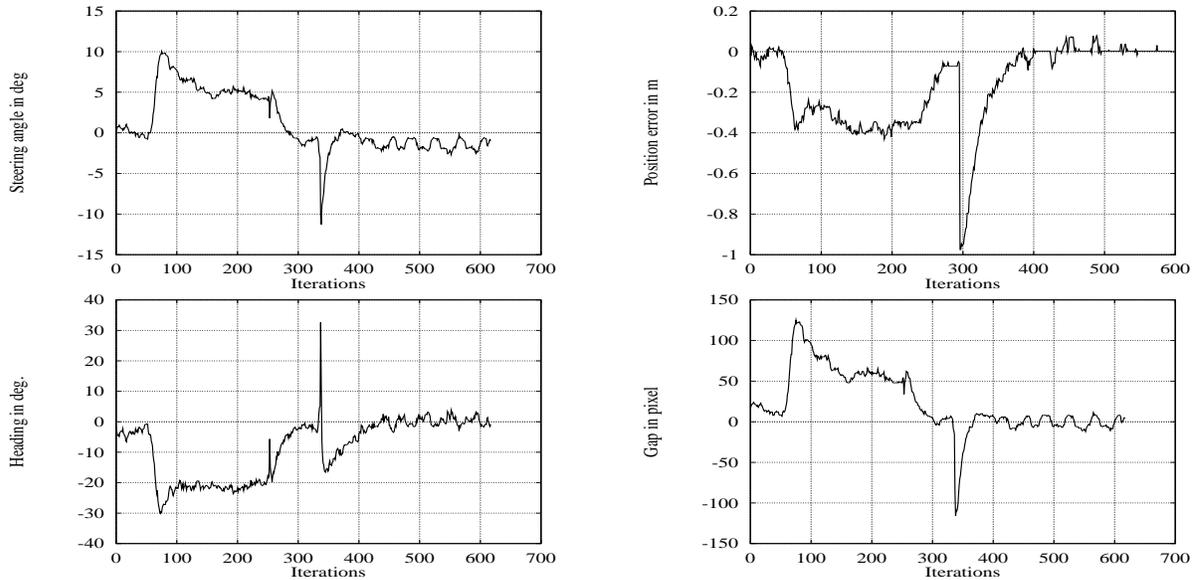


Figure 14: Second law : test at 4 km/h with adapted parameters λ and β

vibrates considerably when the machine runs. The same image processing algorithm is used to control the machine and to measure the error. We will see that noise has no consequence on our system.

Figure 15 presents results obtained in the same conditions, but introducing noise into the image processing (figure 15). The same remark as for the first law can be made in this case about bend-following performance.

It seems that there is no difference between the two cases (with or without noise). In fact, the noise of the algorithm which is used to control the quantity of our system has more consequences than the effect of noise introduced in the result of the image processing.

The test representation in Figure 16 was carried out at different speeds and without noise. Here, the response of the machine is reduced from the first case (test at 4 km/h with adapted parameters). In fact, the machine has the same trajectory but needs less time to do it. The curves show that speed causes the same behaviour as in the first case, since the gains become too high when the speed increases.

4.4 Results using the third Law (neural approach)

In this case, the training phase was carried out at a velocity of 4 km on a training track different from the test track (in particular, the radius of the bend is large). Figure 17 shows the results obtained respecting the conditions of training with regard to velocity. The machine follows the track with good stability.

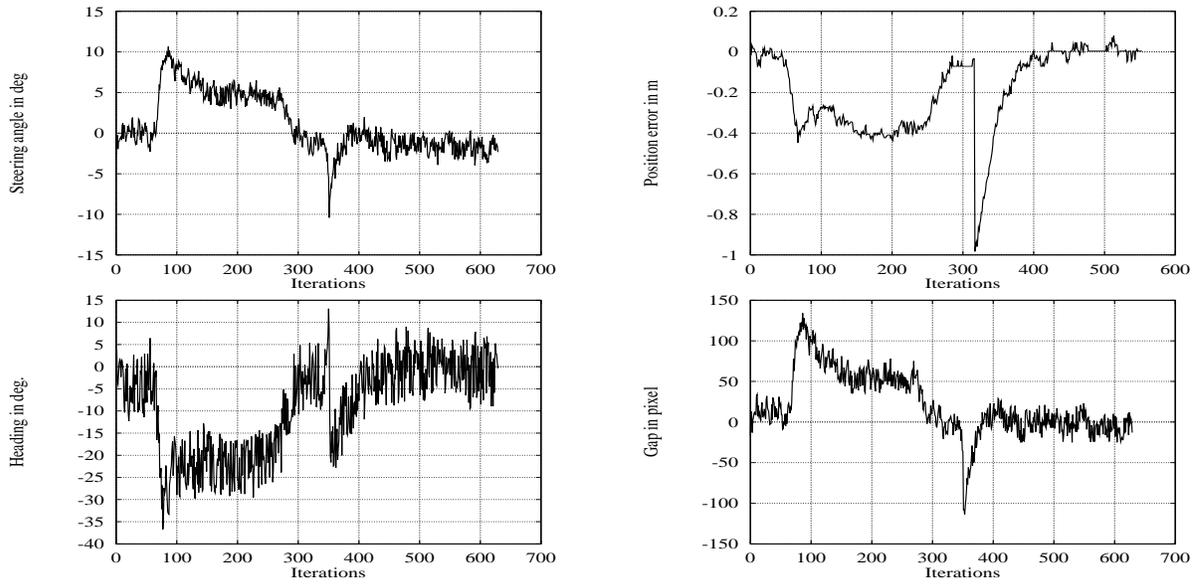


Figure 15: Second law : test at 4 km/h with 16 pixels of noise in ρ

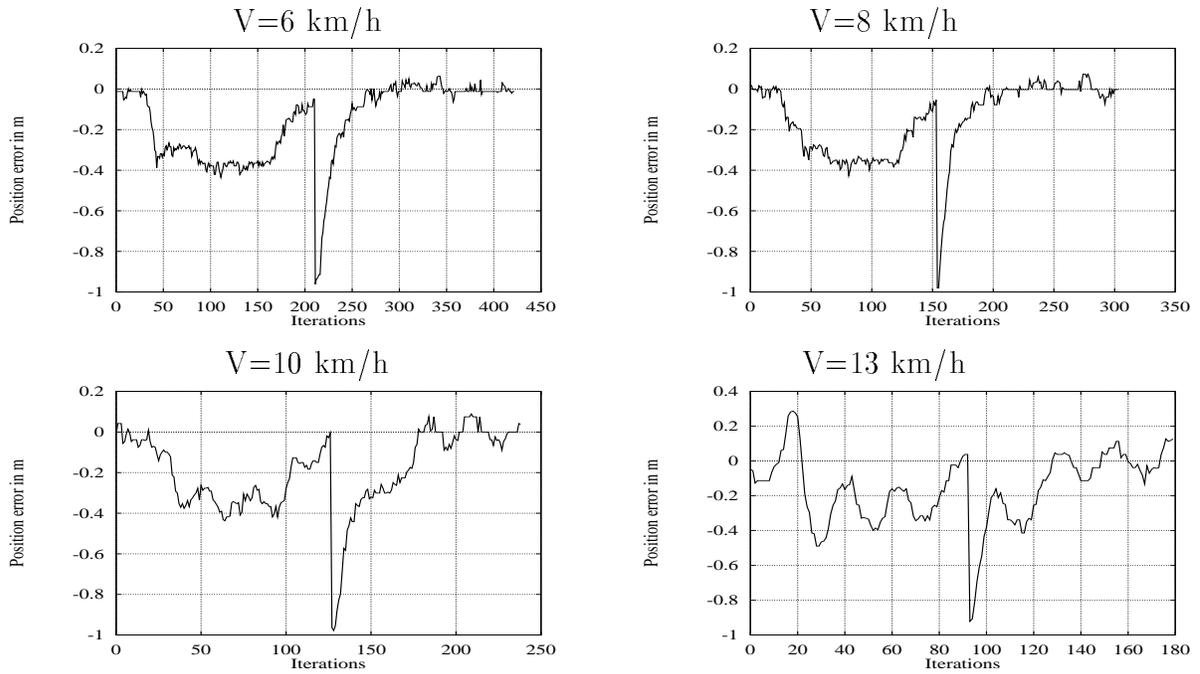


Figure 16: Second law : test at different velocities

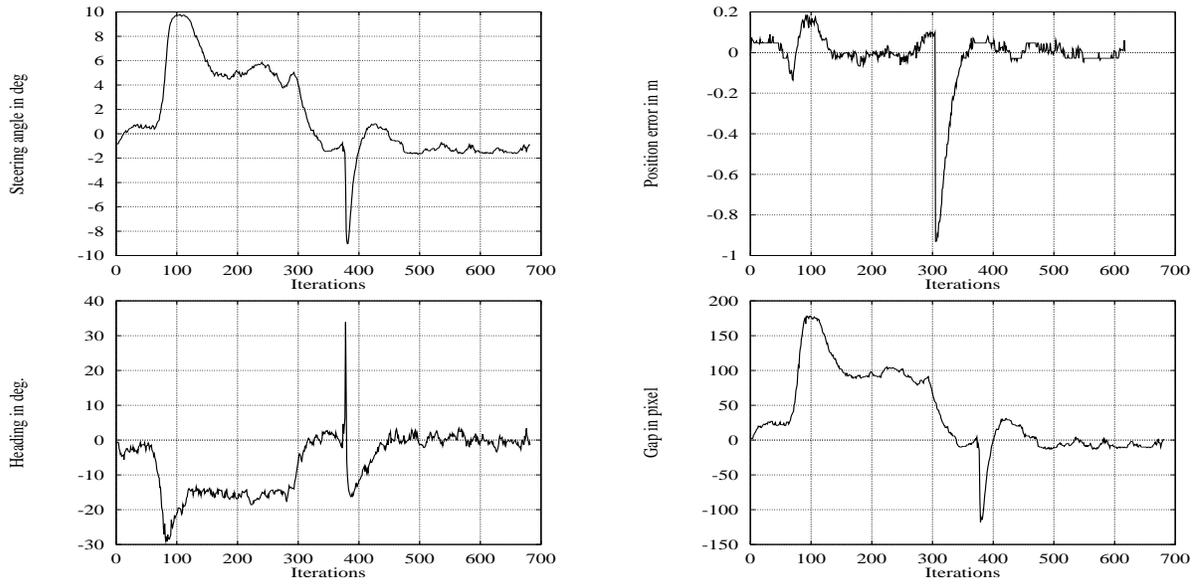


Figure 17: Third law : test at 4 km/h

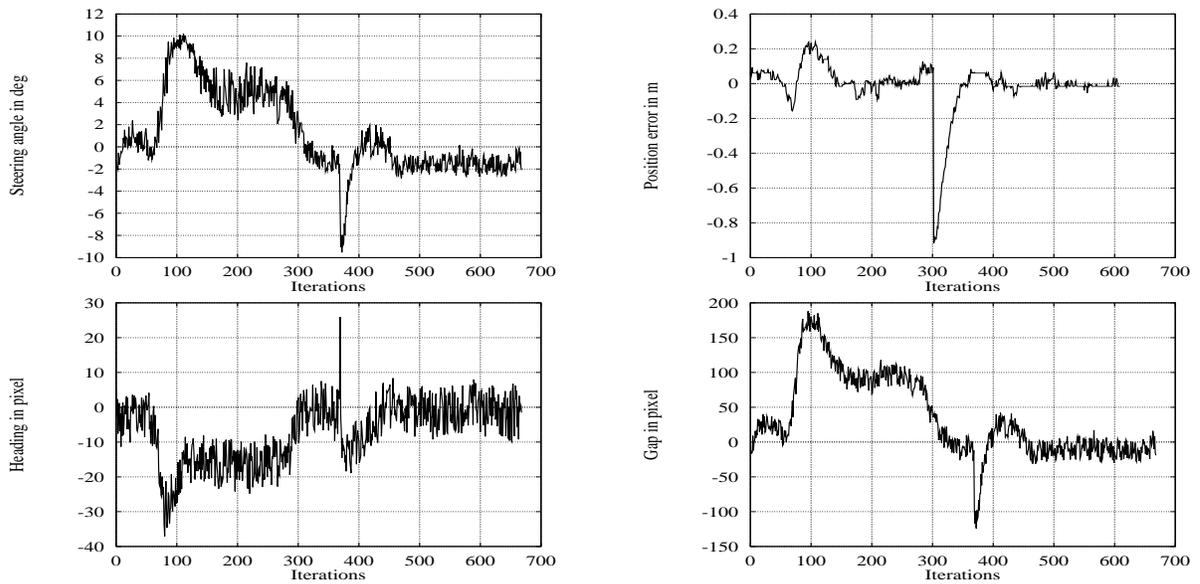


Figure 18: Third law : test at 4 km/h with 16 pixels of noise in ρ

For the same longitudinal velocity of 4 km/h, random noise was added to the gap ρ . The addition of this noise has no effect on the performance of the vehicle in following the line, although the deterioration in the visual signals is clearly visible in the curves (see Figure 18). One possible interpretation of this phenomenon is that a high degree of precision is not required in the orders applied to the steering (wheel angle), because the machine itself constitutes a natural filter for the turning orders.

Three speed values were also tested (4 km/h, 6 km/h and 8 km/h) and gave us a good result. We did not find a notable difference between curves shown in Figure 19. We only note a small performance degradation at the beginning of the bend, when the speed increases. We observe an overshoot of 18 cm at 4 km/h, 25 cm at 6 km/h and 39 cm at 8 km/h.

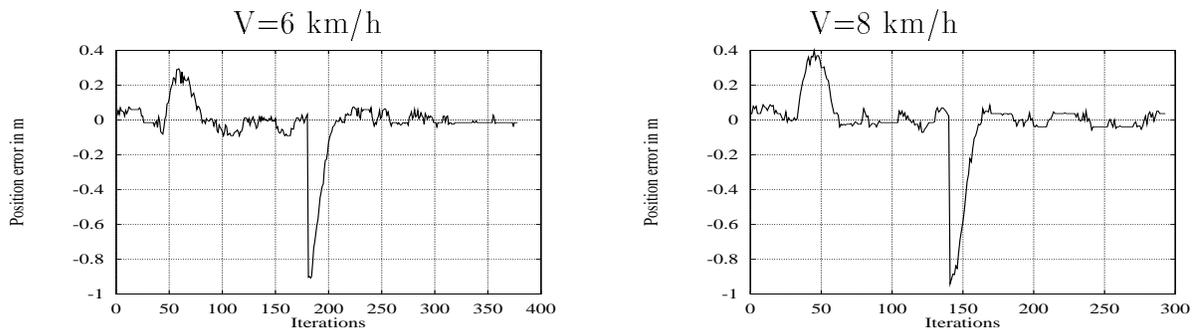


Figure 19: Third law : test at different velocities

5 Conclusion

We have presented a system that provides the driving assistance to an agricultural mobile machine in order to help human beings in repetitive and difficult tasks in a natural environment. The system is based on a vision system using a single camera (Derras 1993). Three vision-based control laws were tested successfully. For each law, the control is directly specified in terms of regulation in the image space. It is noticeable that this approach has the advantage of avoiding the intermediate step of 3D estimation of the environment with regard to the machine. This enables us to introduce into the control loop features directly measured in the same space, which prevents us from making poor estimations of 3D measurements in the case of three dimensional control. In the regulation task function approach (the first and second laws), we need to construct an interaction matrix related to the scene. We also need to know the vehicle model. The neural approach is a “black-box modelling” method and it avoids specific modelling tasks such as explicit camera calibration and vehicle modelling.

With the first law, the gains of the control law developed are adaptive with regard to the desired situation to be reached in the image space and to machine speed. We have to fix the behaviour of the vehicle to a second order system characterized with known parameters (ω_0 and ξ). So we do not need a phase of empirical gain research. We may note that the machine is modelled by a kinematic equation, and we then show that it is amply sufficient for this kind of application. All the results obtained show good convergence and the robustness of our algorithm. They are good enough to demonstrate the feasibility of such an approach which can be extended for any other steerable, mobile, wheeled robot.

With the second law, we show that our theory can be used for many machines. Our gains λ and β are chosen by means of a desired response curve adapted to the machine. All our results show the robustness of our control algorithm. However, the regulation is done in movement and we do not solve the problem of controlling a non-holonomic machine with a final speed equal to zero.

One advantage of the first and the second control laws is due to their adaptability with regard to longitudinal velocity. But their disadvantage is that they do not react correctly in the case of a sharp bend. The principle reason is that the camera sees the scene approximately 6.5 m in front of the machine, and then anticipates its actions. Another reason is that we did not model the bend. In fact the first and the second control expressions are the same, the only difference being the method for selecting the gains. The choice of the parameters, (ξ, ω_0) for the first, and (λ, β) in the second permits to modify the pole location of the system's closed loop.

The third law gives correct results on driving track. The automatic guidance system rivals a human driver in performance. During the tests above 8 km/h, the neural network does not react sufficiently quickly in the beginning of the bend, and then we have a systematic check with loss of detection of the line. This is due first to the modification of the dynamic characteristics of the vehicle with regard to the velocity used in the training phase. So the model built up by the neural network does not correspond sufficiently to reality for higher speeds. Secondly, the system can not take sufficient account of the heading information ρ at the beginning of the bend. In fact, the beginning of the bend is translated first by the modification of the gap of the line detected and then by variation of the heading.

At this level, the best results can be obtained by controlling the contents of the training data, or taking into account the velocity parameter V in the neural network, but this complicates the data training.

Working in an agricultural domain, with industrial partners, we have often adopted a pragmatic approach. Our methods need not be highly sophisticated, but should be easy to implement on different vehicles. We improve algorithms, and adapt them to each application, in order

to obtain fast response and stability, with robustness with regard to speed and curve changes. With our present know-how, we can apply driving assistance systems to different situations and different vehicles. To do that, we are developing robust image segmentation algorithms (Derras 1993; Derras et al. 1995). For instance, in agricultural applications like harvesting we use this kind of approach to extract the reap limit between harvested and unharvested areas modelled as a line whose parameters are used by the control laws previously described. Figure 20 gives examples of reap limit detection for an agricultural application.

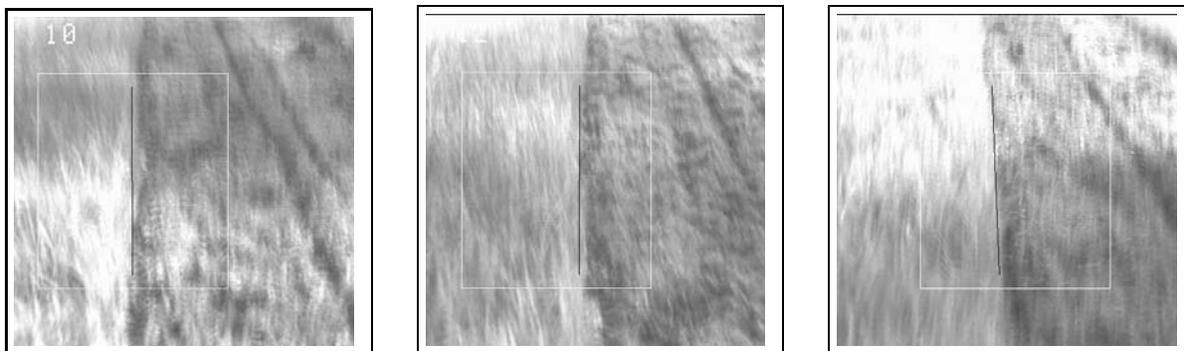


Figure 20: Results with image processing algorithms

6 References

- Amat, J., Battle, J., Fuertes, J.M. and Martinez, A.B. 1993. Vision controlled robot for agricultural application. In *Proceedings of the 24th ISIR*, Tokyo, Japan, pp. 537–542.
- Abbassi, M. and Sayed, M.R. 1991. Class of learning algorithms for multilayer perceptron. In *Proceedings of the SPIE*. International Society for Optical Engineering, 1396:237–242.
- Casals, A., Amat, J. and Grau, A. 1993. Texture parametrization method for image segmentation. In *Proceedings of Second European Conference on Computer Vision ECCV'92*, pp. 160–164.
- Chaumette, F. 1990 (July). La relation vision-commande : théorie et application à des tâches robotiques. Thèse de PhD, IRISA/INRIA-Rennes, Rennes, France.
- Chaumette, F., Rives, P. and Espiau, B. 1991 (April). Positioning of a robot with respect to an object, tracking it and estimating its velocity by visual servoing. In *Proceedings of the IEEE, International Conference on Robotics and Automation*, Sacramento, California, pp. 2248–2253.
- Dayhoff, J. 1990. *Neural Network Architectures: an introduction*, Chap. The exclusive or : a classic problem. VNR Press.

- Debain, C., Derras, M., Berducat, M., Bonton, P. and Gallice, J. 1994. Development of a visual guidance system for an upkeep robot of natural environments. In *Proceedings of the International Symposium on Signal Processing, Robotics and Neural Networks*. IMACS, Lille, France, pp. 121–124.
- Debain, C. 1996 (September). Lois de Commande pour le Contrôle et la Mobilité de Machines Agricoles. Thèse de PhD, Université Blaise-Pascal, Clermont-Ferrand, France.
- Derras, M., Berducat, M. and Bonton, P. 1991 (September). Vision guided mower for the upkeep of natural environment. In *Proceedings of the 1st International seminar of on-machine vision systems for the agricultural and bio-industries*, Montpellier, France, pp. 37–46.
- Derras, M., Berducat, M., Bonton, P., Gallice, J. and Canals, R. 1993. Segmentation texturale originale appliquée au guidage visuel d'un robot d'entretien d'espaces naturels. In *Proceedings of the 14^{ème} colloque sur le traitement du signal et des images*. GRETSI, Juan-les-Pins, France, pp. 771–778.
- Derras, M. 1993 (December). Segmentation non supervisée d'images texturées par champs de Markov : Application à l'automatisation de l'entretien des espaces naturels. Thèse de PhD, Université Blaise-Pascal, Clermont-Ferrand, France.
- Derras, M., Verrat, L., Berducat, M. and Bonton, P. 1995. Image processing and algorithms merging : Real time control of an upkeep machine for natural spaces. In *Proceedings of the Fourth Workshop on Robotics in Agriculture and the Food-Industry*, pp. 93–98.
- Espiau, B., Chaumette, F. and Rives, P. 1992. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3):313–326.
- Faugeras, O. D. 1993. *Three Dimensional Computer Vision : A Geometric Viewpoint*. MA: MIT Press.
- Hager, G.D., Hutchinson, S. and Corke, P. 1996 (April). *Visual Servo Control*. *IEEE International Conference on Robotics and Automation*. Minneapolis, Minnesota, Tutorial TT3.
- Ioos, E., Boulle, M. and Tournassoud, P. 1988 (February). Etude d'un sous-système de navigation pour un robot mobile. Technical report No 783, INRIA Rocquencourt, France.
- Inerney, J.M. Mc, Haines, K.G., Biafore, S. and Hecht-Nielsen, R. 1989. Error surfaces of multi-layer networks can have local minima Technical report No 157, CS89, UCSD, CA.
- Jarvis, R.A. 1990. Omniscient camera based outdoor mobile robot. In *Proceedings of the Workshop on Robotics in Agriculture and the Food Industry*. 1st IARP, Avignon, France, pp. 247–258.
- Jurie, F., Rives P., Gallice, J. and Brame, J.L. 1994. High-speed vehicle guidance based on vision. *Control Eng. Practice*, 2(2):287–297.
- Kelly, A. 1994 (May). A partial Analysis of the High Speed Autonomous Navigation Problem.

- Technical report, The Robotics Institute Carnegie Mellon University.
- Khadraoui, D., Martinet, P. and Gallice, J. 1995. Linear control of high speed vehicle in image space. In *Proceedings of the Second International Conference on Industrial Automation*. IAIA, Nancy, France, 2:517–522.
- Khadraoui, D., Motyl G., Martinet, P., Gallice, J. and Chaumette, F. 1996 (October). Visual servoing in robotics scheme using a camera/laser-stripe sensor. *IEEE Transactions on Robotics and Automation*, 12(5):743–750.
- Khadraoui, D. 1996 (December). La Commande Référencée Vision pour le Guidage Automatique de Véhicules. Thèse de PhD, Université Blaise-Pascal, Clermont-Ferrand, France.
- Klassen, N.D., Wilson, R.J. and Wilson, J.N. 1994. Guidance systems for agricultural vehicles. In *Proceedings of the 12th World Congress on Agricultural Engineering*. International Commission of Agricultural Engineering, Milano, Italia, pp. 1136–1142.
- Le Cun, Y. 1987 (June). *Modèles connexionnistes d'apprentissage*. Thèse de PhD, Université Paris 6, Paris, France.
- Murray, R.M. and Sastry, S.S. 1993 (May). Nonholonomic motion planning: steering using sinusoids. *IEEE Transactions on Automatic Control*, 38(5):700–716.
- Paul, R. P. 1982. *Robot Manipulators : Mathematics, Programming, and Control*. MIT Press. Massachusetts Institute of Technology, Cambridge, Massachusetts 02142.
- Pissard-Gibollet, R. and Rives ,P. 1991 (December). Asservissement Visuel Appliqué à un Robot Mobile: état de l'art et modélisation cinématique. Technical report No 1577, Rapport de recherche INRIA.
- Rumelhart, D. et al 1986. *Parallel and Distributed Processing*. MIT Press.
- Samson, C., Borgne M. Le and Espiau, B. 1991. *Robot Control. The task function approach*. Oxford, ISBN 0-19-8538057.
- Sandini, G., Buemi F., Massa, M. and Zucchini. 1990. Visually guided operations in green houses. In *Proceedings of the Workshop on Robotics in Agriculture and the Food Industry*. 1st IARP, Avignon, France, pp. 69–84.