Proceedings of the 1992 IEEE
International Conference on Robotics and Automation
Nice, France - May 1992

# Wall-following Control of a Mobile Robot

P. van Turennout G. Honderd L.J. van Schelven
Control Laboratory, Dept. of Electrical Engineering, Delft University of Technology
P.O. Box 5031, 2600 GA Delft, The Netherlands

## Abstract

*In this paper the wall-following control of a mobile robot is discussed. The wall-following control problem is characterized by moving the robot along a wall in a desired direction while maintaining a constant distance to that wall. From ultrasonic distance measurements the distance and the orientation of the robot with respect to the wall can be calculated. This is solved by the use of an observer: the distance and orientation are estimated using a robot model and corrected by sensor measurements. Since a wall may not be available continuously (e.g. an open door), the robot must be able to navigate on its dead-reckoning as well. The feedback controller has been set up in such a way that it can handle both the observer data and dead-reckoning data. The controller has been verified by means of experiments. The results show a good performance with an absolute error of a few millimetres from the desired distance to the wall.*

## 1 Introduction

There are several reasons why autonomous mobile robots must be able to follow walls, or in a more general sense, to follow the contours of an object. This depends, of course, on the type of mobile robot and its application. Robots operating in an unknown, unstructured environment (usually outdoors) use their sensors to perceive the surroundings and (re)plan their motions or trajectories accordingly. Perspective information is very important in unstructured environments and hence vision systems (very often stereo vision systems) are mainly used. Examples of these kinds of applications are (extra)terrestial exploration [6, 8, 14] and road following [2, 3, 10, 15]. The data of the sensor systems is used to analyze the environment in front of the vehicle and to plan the next, local part of the trajectory. There is no direct link between the sensor output and the trajectory control. There is also no need to build a map of the environment. At least, not a global map containing the information in the representation the robot uses to plan future motions (it may be a map for human interpretation, though). The second category consists of robots moving in a structured environment, i.e. an (indoors) environment designed for humans (buildings, offices, homes) or especially designed for mobile robots (automated assembly halls, factories or warehouses). This environment can be known: a map is present (either a priori or *after* a stage of world modelling), or unknown: no map present (either intentional or *before* the world modelling phase). Also in this case, vision systems are used to look ahead and (re)plan local trajectories.

Applications which require a 'look-ahead strategy' to plan the next part of the trajectory are not considered here. This paper concentrates on the case in which sensor data directly influence the trajectory control actions. This can be the case in the following situations.

- *Obstacle avoidance.* When the sensors can not provide an overview of the shape or the size of the obstacle, it is not possible to plan an evasive route. Then it becomes necessary to follow the contour of the obstacle until such is possible or until the original route can be resumed.
- *Following an unknown wall.* When there is little or no information about the environment, the trajectory may be specified as 'follow the wall on the right until the first doorway'. When world modelling is the specific goal of the robot, it may be necessary to follow a wall to model it completely. The position of the robot (relative to its starting point) can be calculated by means of dead-reckoning. If (and only if!) this position is known accurately during the manoeuvre, the wall can be placed in the map.
- *Following a known wall.* When the trajectory of the robot has been planned, it can be followed by means of dead-reckoning. However, dead-reckoning methods suffer from accumulating errors. A way to keep these errors small is by tracking a wall [5, 1]. The planning must include the availability of walls in the route description in that case (like in [7, 12]). This is the reverse of the previous case: now the position of the wall has to be known accurately.

0-8186-2720-4/92 $3.00 ©1992 IEEE

In all cases there is a cooperation between the use of dead-reckoning and the use of external sensors. The switching between both is done by a higher level decision process which monitors the sensor data. For example, the detection of a doorway while following a wall causes the control to switch from the use of distance measuring sensors to dead-reckoning. When the wall is found again, the reverse takes place. Also, in the case of obstacle avoidance it must be decided when the originally planned route can be resumed. Because of this switching it is important that the data from dead-reckoning and the sensors are kept consistent.

In this paper the wall-following control of a mobile robot is considered with the use of ultrasonic sensors for distance measurement. In literature, only mobile robots using vision systems have been found to achieve this (for example, [4, 9, 11]). The following of a wall is characterized by keeping a certain (constant) distance to the wall. Ultrasonic sensors are a simple means of measuring this distance. Thus, there is no need for complex and expensive vision systems. Further, as it concerns relatively small distances (typically $\leq 1$ metre) which are measured (almost) perpendicular to the wall, the main disadvantages of these sensors, being the poor lateral resolution and the large beam width, do not play an important role. The robot considered in this paper is a differentially steered robot; it has two independently driven wheels on the centre line. Caster wheels for support are placed in front and at the back. On each side of the robot is one Polaroid ultrasonic sensor, also on the centre line, just above the drive wheel. A dynamical model for this robot is presented in the next section. Section 3 explains the feedback controller based on this model. The use of the ultrasonic sensors in wall-following has resulted in an observer structure. This observer is treated in section 4. The controller including the observer has been simulated and verified by experiments. The results are given in section 5. The paper ends with conclusions and a perspective for future research.

## 2 Robot model

The robot position in the Cartesian space is given by two planar coordinates $x$ and $y$ and the orientation $\Psi$ of the robot with respect to some reference frame. The dynamic model of the robot is presented here, showing how the position depends upon the control inputs. The desired trajectory of the robot is assumed to be a straight line. A state feedback controller is used to keep the robot on this line; it is explained in the next section.

The mobile robot can be considered as a system with two inputs and three outputs. The two inputs are the voltages to the power units of the DC-motors. The outputs are the position coordinates $x$, $y$ and $\Psi$. Each DC-motor can be modelled as a first order system with the voltage as input and the rotation speed as output ($U_r$, $\omega_r$ and $U_l$, $\omega_l$ for the right and left wheel, respectively). Not the wheel speeds are of interest, however, but the translation and rotation speeds of the robot as a whole. These can easily be found as

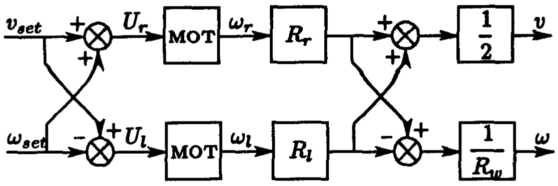

*Figure 1. First part of robot model.*

$$
\begin{aligned}
v &= \frac{1}{2}(R_r\omega_r + R_l\omega_l) && (1)\\
\omega &= \frac{1}{R_w}(R_r\omega_r - R_l\omega_l) && (2)
\end{aligned}
$$

with $R_r$, $R_l$ the wheel radii and $R_w$ the distance between the wheels. The input voltages are redefined as the set-points $v_{set}$ and $\omega_{set}$ corresponding to the speeds $v$ and $\omega$ (assuming that a simple scaling operation in software is used to remove the gain factors in the DA-converters and the power units). In figure 1 the first part of the robot model has been depicted. The second part of the model concerns the derivation of the position coordinates from the robot speeds. For this purpose a reference frame must be defined. This reference frame is chosen such that the $x$-axis is aligned with the desired trajectory and the $y$-axis is perpendicular to it. This trajectory consists either of a straight line or a wall. The latter case represents the following of a straight line with an offset in the $y$-direction. The orientation is defined as the angle between the translation speed $v$ of the robot and the $x$-axis. Figure 2 illustrates these definitions. The speeds of the position coordinates are found as

$$\dot{x} = v\cos\Psi, \qquad \dot{y} = v\sin\Psi, \qquad \dot{\Psi} = \omega \qquad (3)$$

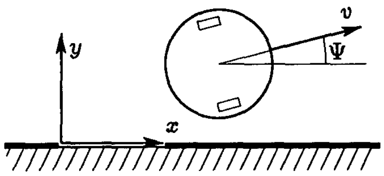

*Figure 2. Reference frame for robot positions.*

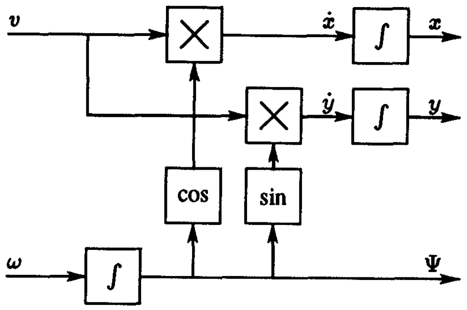

*Figure 3. Second part of robot model.*

The position coordinates themselves are found by integration. The second part of the model is summarized in figure 3. Both parts can be concatenated to form the complete model. However, some simplifications can be made. First, the double cross couplings in the first part cause coupling terms between the translation and rotation speeds. Assuming equal wheel radii and identical motor control units, these terms can be neglected. The transfer functions between $v_{set}$ and $v$ and between $\omega_{set}$ and $\omega$ are approximated by first order systems with gains equal to one and time constants $\tau_v$ and $\tau_\omega$ respectively. Second, as it is the task of the control to keep the robot on a straight line, the orientation $\Psi$ may be assumed small. As a consequence, $\cos\Psi \approx 1$ and $\sin\Psi \approx \Psi$. Keeping the orientation smaller than the maximum beam width of the ultrasonic sensors is also necessary to prevent the sound waves from being reflected away and hence make it impossible to measure the distance. Third, the translation speed can be taken as a constant while driving along a straight line. This implies that the change in $y$-direction is proportional to the orientation: $\dot{y} = v\Psi$, in which $v$ can be considered as a gain factor. Otherwise, $\dot{y}$ would be obtained by the product of two time dependent variables, which is a mathematically unpleasant situation in the controller design. With these simplifications the complete robot model can be represented in the Laplace domain by the block diagram in figure 4.

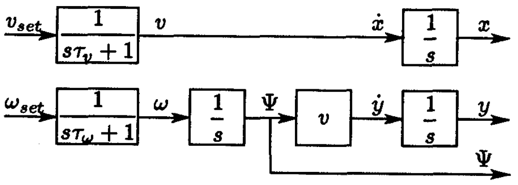

*Figure 4. Simplified dynamic model of a mobile robot.*

## 3 Feedback control

The task of the controller is to supply the robot with setpoints for $v$ and $\omega$ based upon reference values and measurements of the robot state. The reference values originate from a planning stage, either based upon a route-searching algorithm, use of look-ahead perception strategies, or upon a combination of both. The measurement of the robot state can be done by means of dead-reckoning or by use of the ultrasonic sensors. Preferably, the controller should be the same for both cases. Depending on the control strategy the inputs of the controller are switched. This has the following advantages. The feedback gains in the controller need to be optimized only once. Also, each control strategy benefits from any future improvements. Finally, it saves implementation space.

Figure 4 suggests two independent controllers: one for the $x$-coordinate and one for the combined $y$-coordinate and orientation $\Psi$. The $x$-coordinate equals the displacement of the robot. The displacement is controlled by means of position and velocity feedback.

$$v_{set} = K_s(x_{set} - \hat{x}) - K_v\hat{v} \tag{4}$$

where $x_{set}$ is the desired amount of displacement. The velocity feedback is mainly used to meet performance requirements. The displacement and the translation speed can be obtained from the shaft encoders. The lower part of figure 4 shows a double pure integration in the transfer function for the $y$-coordinate. This necessitates the feedback of both $y$ and $\dot{y}$ in order to keep the process stable. However, in case of zero speed $v$, it must still be possible to control the orientation. Hence, $\Psi$ is fed back instead of $\dot{y}$. The feedback of $\hat{\omega}$ serves again to meet performance requirements.

$$\omega_{set} = K_y(y_{set} - \hat{y}) - K_\Psi\widehat{\Psi} - K_\omega\hat{\omega} \tag{5}$$

The values $\hat{x}$, $\hat{y}$, and so on, indicate measured or estimated values. Both setpoints for the speeds are limited to the maximum allowed robot speeds. The feedback gains have been optimized for the default speeds of the robot. In practice, this controller still performs satisfactory when the assumption of constant translation speed does not hold. As $y$ and $\Psi$ are strongly dependent, the values of the feedback gains in equation (5) are a compromise for the best performance of the control of both $y$ and $\Psi$.

## 4 Observer design

From the ultrasonic measurements the distance (i.e. the $y$-coordinate) of the centre point of the robot to the wall can easily be calculated.

$$y_{us} = D_{us} + R_{sensor} + R_{sensor}(\cos\Psi - 1) \quad (6)$$

with $D_{us}$ the measured distance and $R_{sensor}$ the distance between the sensor and the centre point of the robot. The last term is a correction for the displacement of the sensor as the robot turns away from the wall. Obtaining values for the orientation implies the differentiating of the distance data since $\dot{y} = v\Psi$. The differentiation can be approximated by taking the difference of two measurements, divided by the displacement of the robot between those measurements:

$$\Psi_{us} = \arcsin\frac{D_2 - D_1}{\Delta t \cdot v} \approx \frac{D_2 - D_1}{\Delta t \cdot v} \quad (7)$$

in which $D_1$ and $D_2$ are two successive measurements and $\Delta t$ is the time between these measurements. This operation is rather noise sensitive and does not give very accurate results. Use of two sensors instead of one has the obvious drawback of needing an extra sensor. Further, the two sensors may not be used simultaneously as the echos might get mixed. Finally, equation (7) may only be used when both sensors are measuring to the wall. Irregularities in the wall (obstacles or doorways, for example) must be passed by both sensors before a reliable value for $\Psi$ can be obtained. A solution for the problem of calculating the orientation from single sensor measurements has been found in the use of an observer [13].

The basic set up of the observer consists of a robot model, which receives the controller output, and feedback from sensor data. The estimated $\hat{y}$-coordinate of the robot is obtained by the lower part of the model in figure 4. The real $y$-coordinate is measured by the ultrasonic sensor. The difference between these values is fed back through a gain vector $G$ (figure 5). The gains reflect the reliability of the sensor measurements compared to the reliability of the model outputs. They are *static* gains (in contrast to a Kalman filter, for instance) and have to be calculated beforehand. Hence, the observer algorithm is not that complex and can be implemented relatively easy. The observer is shown once again in more detail in figure 6, where the first part of the robot model with the first order transfer functions has been truncated. In this figure, it can clearly be seen how the estimates $\hat{y}$ and $\hat{\Psi}$ are corrected through the observer gains $G_y$ and $G_\Psi$, respectively. Proper values have to be chosen for these gains. Large values will provide a fast convergence and reduction of initial estimation errors and a smaller sensitivity to modelling errors. It increases the sensitivity to sensor noise, however. As this sensor noise turns out to be small, a large value for $G_y$ can be chosen. The value for $G_\Psi$ can not be chosen very large, since an initialisation error of $y_{us} - \hat{y}$ immediately disturbs the estimate $\hat{\Psi}$.

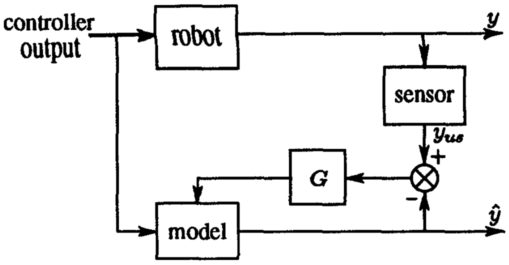

*Figure 5. Global set up of the observer structure.*

The values for both gains have been optimized by means of simulations.

## 5 Experimental results

The controller with observer has been tested by experiments. During these experiments, a constant speed of 0.4 m/s was maintained. The robot started at a distance of 22 cm more than the setpoint distance, with an orientation already parallel to the wall. The results are shown in figure 7. The upper part shows the difference $D$ between the setpoint distance $y_{set}$ and the measured distance $y_{us}$. The lower part shows the convergence of $\hat{\Psi}$ to the orientation $\Psi$ calculated by the dead-reckoning algorithm. It can clearly be seen that the estimated orientation jumps to a positive value due to the large initial distance error. The estimate $\hat{y}$ almost immediately converges to the calculated $y$ by the dead-reckoning; this is not shown. It is important to note that both $D$ and $\hat{\Psi}$ do not converge to zero. There remains a steady state error of about 2 cm in the distance to the wall. The cause of this can be found by investigating the steady state values for the estimates from figure 6. In steady state, the following relations hold:

$$\hat{y} - y_{us} = \frac{\omega_{set}}{G_\Psi} \quad (8)$$

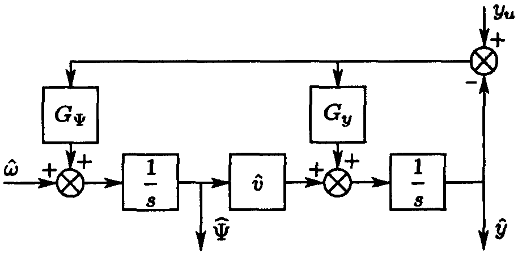

*Figure 6. Observer for distance and orientation.*

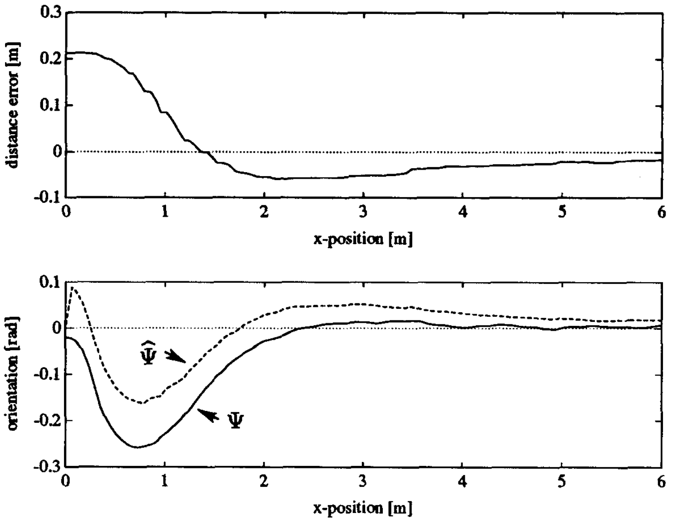

*Figure 7. Response of distance error D (top) and estimated orientation $\hat{\Psi}$ compared to the $\Psi$ calculated by dead-reckoning (bottom) for the initial observer.*

$$\hat{\Psi} = \frac{\omega_{set} G_y}{\hat{v} G_\Psi} \tag{9}$$

Substitution of equation 5 to eliminate $\hat{y}$ yields for the distance error:

$$y_{set} - y_{us} = \omega_{set} \cdot \frac{(K_\omega + 1)\hat{v}G_\Psi + \hat{v}K_y + K_\Psi G_y}{\hat{v}K_y G_\Psi} \tag{10}$$

This error is non-zero for a non-zero value of $\omega_{set}$. When in steady state the robot remains parallel to the wall, $\omega_{set}$ is expected to be zero. However, as mentioned in section 2, there are small coupling terms between the translation and rotation speeds. For a constant speed $v$, there is a small coupling factor $C_{v\omega}$ to $\omega$. In steady state, this term is compensated by $\omega_{set} = -vC_{v\omega}$. The resulting error can be reduced by increasing the gain $G_\Psi$, but this leads to an even larger disturbance in the orientation estimation.

To solve the problem of the steady state errors, the following modification to the observer has been made. As soon as the difference $\hat{y} - y_{us}$ exceeds a certain value (presently at 5 cm), the gain $G_\Psi$ is set to zero for the present sample. Now, the gain can be increased without the danger of large disturbances in the estimation of $\Psi$. Figure 8 shows the results with a gain $G_\Psi$ four times larger than in the previous case. The errors in $D$ and $\hat{\Psi}$ have been significantly reduced. The setpoint $y_{set}$ is now reached within a few millimetres. Also, the estimate $\hat{\Psi}$ converges faster and shows no large disturbances.

## 6 Conclusions

The wall-following control of a mobile robot has been presented using ultrasonic sensors. The sensor data is used in an observer to obtain the estimates of the distance and orientation of the robot. These estimates are utilized in the feedback controller. A dead-reckoning algorithm is also present on the robot. When the wall is no longer available, the controller switches to dead-reckoning data. At first, the observer performs not quite satisfactory: there remains a static deviation of a few centimetres in the distance. By delaying the correction on the orientation and increasing the observer gains, this error can be greatly reduced. In the final experiments the robot has an error of a few millimetres from the desired distance to the wall.

Further research includes the improvement of the observer by a better and more detailed robot model (cross coupling terms). Another possibility is the use of the dead-reckoning algorithm instead of the robot model. A further comparison of the observer with statistical methods and Kalman filtering with respect to performance versus complexity seems appropiate.

## References

[1] Cox, I.J., *Blanche — An Experiment in Guidance and Navigation of an Autonomous Robot Vehicle*, IEEE Journal of Robotics & Automation, Vol. 7, No. 2, pp. 193–204, 1991.

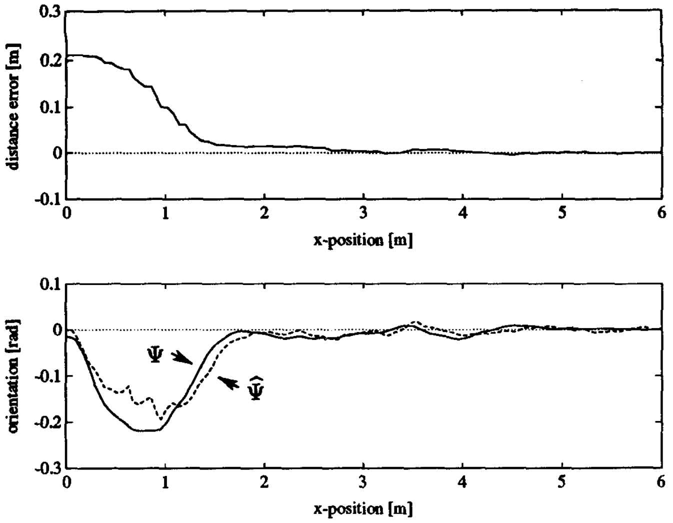

*Figure 8. Response of distance error D (top) and estimated orientation $\hat{\Psi}$ compared to the $\Psi$ calculated by dead-reckoning (bottom) for the observer with switching gain $G_{\Psi}$.*

[2] Davis, L.S., *Visual Navigation at the University of Maryland*, Proc. of the 2nd Int. Conf. on Intelligent Autonomous Systems, Amsterdam, the Netherlands, pp. 1–19, 1989.

[3] Dickmanns, E.D. and T. Christians, *Relative 3D-State Estimation for Autonomous Visual Guidance of Road Vehicles*, Proc. of the 2nd Int. Conf. on Intelligent Autonomous Systems, Amsterdam, the Netherlands, pp. 683–693, 1989.

[4] Drake, K.C., E.S. McVey and R.M. Inigo, *Experimental Position and Ranging Results for a Mobile Robot*, IEEE Journal of Robotics & Automation, Vol. RA–3, No. 1, pp. 31–42, 1987.

[5] Durrant-Whyte, H.F. and J.J. Leonard, *Navigation by Correlating Geometric Data*, Proc. IEEE/RSJ Int. Workshop on Intelligent Robots and Systems (IROS), Tsukuba, Japan, pp. 440–447, 1989.

[6] Gat, E., M.G. Slack, D.P. Miller and R.J. Firby, *Path Planning and Execution for a Planetary Rover*, Proc. IEEE Int. Conf. on Robotics & Automation, Cincinatti, USA, pp. 20–25, 1990.

[7] Habib, M.K. and S. Yuta, *Efficient On-line Path Planning Algorithm and Navigation for a Mobile Robot*, Int. Journal of Electronics, Vol. 69, No. 2, pp. 187–210, 1990.

[8] Hebert, M., E. Krotkov and T. Kanade, *A Perception System for a Planetary Rover*, Proc. 28th Int. Conf. on Decision & Control, Tampa, USA, pp. 1151–1156, 1989.

[9] Kak, A.C., B.A. Roberts, K.M. Andress and R.L. Cromwell, *Experiments in the Integration of World Knowledge with Sensory Information for Mobile Robots*, Proc. IEEE Int. Conf. on Robotics & Automation, Raleigh, USA, pp. 734–740, 1987.

[10] Kuan, D. and U.K. Sharma, *Model Based Geometric Reasoning for Autonomous Road Following*, Proc. IEEE Int. Conf. on Robotics & Automation, Raleigh, USA, pp. 416–423, 1987.

[11] Madarasz, R.L., L.C. Heiny, R.F. Cromp and N.M. Mazur, *The design of an Autonomous Vehicle for the Disabled*, IEEE Journal of Robotics & Automation, Vol. RA–2, No. 3, pp. 117–125, 1986.

[12] Mori, H., M. Nakai, H. Chen, T. Yuguch and S. Nakayama, *Steering Panning Coordination in Stereotyped Motion*, Proc. IEEE/RSJ Int. Workshop on Intelligent Robots and Systems (IROS), Tsukuba, Japan, pp. 155–162, 1989.

[13] Schelven, L.J. van, *Position Estimation with Ultrasonic Sensors for a Mobile Robot*, Control Laboratory, Faculty of Electrical Engineering, Delft University of Technology, Report nr. A91.049 (in Dutch), 1991.

[14] McTamaney, L.S., *Mobile Robots — Real-Time Intelligent Control*, IEEE Expert, Vol. 2, No. 4, pp.55–68, Winter 1987.

[15] Thorpe, C.E., *Outdoor Visual Navigation for Autonomous Robots*, Proc. of the 2nd Int. Conf. on Intelligent Autonomous Systems, Amsterdam, the Netherlands, pp. 530–544, 1989.