SAE TECHNICAL PAPER SERIES

941000

AUTOPASS: Automated Parking Support System

D. Gorinevsky, A. Kapitanovsky, A. Goldenberg
Robotics and Automation Laboratory
The University of Toronto
Ontario, CANADA

SAE The Engineering Society For Advancing Mobility Land Sea Air and Space® INTERNATIONAL

International Congress & Exposition
Detroit, Michigan
February 28-March 3, 1994

400 COMMONWEALTH DRIVE, WARRENDALE, PA 15096-0001 U.S.A. Tel: (412) 776-4841 Fax: (412) 776-5760

93-0031PG

941000

AUTOPASS: Automated Parking Support System

D. Gorinevsky, A. Kapitanovsky, A. Goldenberg
Robotics and Automation Laboratory
The University of Toronto
Ontario, CANADA

ABSTRACT

This paper describes the development of an efficient automated parking support system for passenger cars. By using recent advances in the Artificial Neural Network technology and a classical combination of linear feedback and feedforward control, we propose a novel design of the parking motion controller.

The paper presents the results of the controller design and analysis, including parking problem analysis, feedback controller stability analysis, formulation and optimal solution of the parking trajectory planning problem, and design of a novel parking motion planning architecture based on a Radial Basis Function network. Three general cases of backward parking considered in this work are emulated using the proposed controller. The emulation results reveal high efficiency of the presented approach and demonstrate that the proposed system can be implemented on a typical passenger car.

1 INTRODUCTION

Control technology will be extensively used in the next-generation passenger cars for enhancing driver's safety and reducing the driving stress. Active suspension and noise compensation control are already an integral part of some cars; other automotive applications of control technology are currently being developed. This work presents a new approach to automated control of car parking. Many automobile companies may be potentially interested in a practical working system for car parking support. For example, a prototype of the parallel parking support system has been demonstrated by Volkswagen during the 19-th Canadian International Autoshow in 1992.

The present paper considers the problem of designing a controller for the automated parking support system (AUTOPASS). The system is currently being developed in the Robotics and Automation Laboratory at the University of Toronto. The AUTOPASS is expected to make the car parking process simpler and stress-free for a typical car driver. In addition to parallel parking shown in Figure 1(a), AUTOPASS can deal with more complicated cases of car parking such as shown in Figure 1(b-c).

A passenger car is a typical example of wheeled mobiled vehicles and, as a such, its motion is characterized by the presence of nonholonomic (non-integrable) constraintes that arise due to the three-dimensional rolling of wheels. Automatic control of a nonholonomic system is a difficult problem. There has been significant research performed on nonholonomic motion planning and control during the last years. The issue of the local and global controllability has been studied in [2], [7], [17], and [21]. A condition for global controllability is given by Chow's theorem (see [17] for reference). Necessary conditions for stabilizability of a nonholonomic system by a smooth state-dependent feedback have been formulated in Brockett's theorem [2]. This theorem relates also to results in [21]. The existence of a smooth time-variant feedback for the asymptotic stabilization of a nonholonomic system has been recently established in [7]. Basically, most of the available results on the stabilization of nonholonomic systems can be classified into two main groups: (i) stabilization to equilibrium manifolds [1] and [19]; and (ii) stabilization at the origin. The latter can be sub-classified as: *optimal open-loop control* in [3], [8], and [22]; *constructive open-loop control* in [3], [25], and [33]; *smooth feedback control* in [2], [16], [20], [28], and [32]; *discontinuous feedback control* in [1]; and *particular control designs* in [31] and [35].

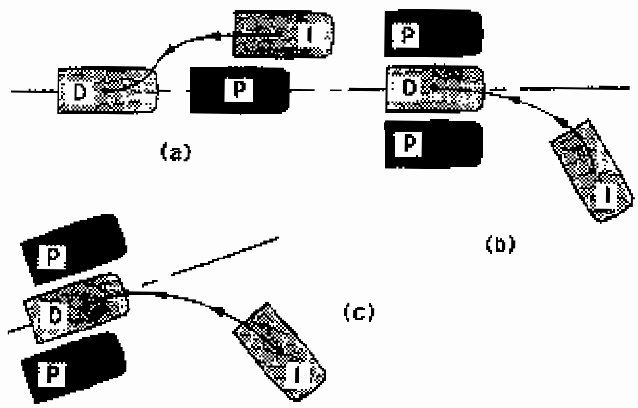

Figure 1: Parking tasks.

Each of the approaches has its advantages and drawbacks. For instance, open-loop strategies handle collision constraints rather well, and generally result in algorithms which are computationally more efficient. On the other hand, feedback approaches result in solutions which are more robust, and the specification of an initial configuration is not necessary. Some published works on control of nonholonomic systems consider the specific problem of controlling a wheeled vehicle using open-loop control [22], feedback control [28] and [31], and neural networks [26].

In this paper, we further facilitate the available techniques by applying an engineering approach to the problem of designing the control system for AUTOPASS. This approach emphasizes the design applicability and implementation issues. In AUTOPASS, all complicated and computationally expensive motion planning needed for automated car parking is done off-line (i.e., prior to programming the on-board computer of a car), and the resulting parking programs are stored in the on-board computer memory. A parking program is regarded herein as the time-dependence of car control commands defined over a desired time interval and stored as a single high-dimensional memory array. In general, each combination of initial and final positions of a car corresponds to a different parking program. All relevant combinations of initial and final parking positions which may occur during the car service are generally unknown *a priori*. AUTOPASS uses a highly efficient and very accurate approximation technique in order to find a parking program for any given initial and final conditions. The pre-computed data are stored as the weights of a Radial Basis Function (RBF) network. An RBF network is an Artificial Neural Network (ANN) architecture that has been recently acknowledged as one of the most efficient and promising ANN-based techniques for the approximation of continuous mappings [4], [5], [6], [24], [27], and [36].

The RBF network-based approximation technique embodied into AUTOPASS overcomes the problem of computational burden associated with open-loop control and, at the same time, has moderate computer memory requirements. The car computer may be concurrently used for other purposes, such as automatic control of the transmission and suspension systems. Related control methods using Polynomial Associative Memories and RBF networks have been successfully tested in numerical simulations and control experiments with advanced electromechanical systems [9], [10], [13], and [14]. To execute a parking program in the real-world conditions, we have provided AUTOPASS with a tracking controller which is based on a combination of linear feedback and feedforward control. The results presented in this paper demonstrate that the proposed ANN-based parking control system architecture may indeed be very efficient in dealing with many complicated cases of passenger car parking.

2 STATEMENT OF THE CONTROL PROBLEM

This section outlines a basic control problem solved by AUTOPASS. Let us consider in-plane motion of a front wheel drive car. The car body position is described by two coordinates x_1 and x_2 of the car center, and the angle ϕ between the car body axis and the horizontal axis. This paper addresses the problem of car motion planning and control during the backward parking process. In practice, car motion during the backward parking process is very slow and, therefore, car body dynamics as well as dynamics of the car motor and steering mechanism may be neglected. We further assume that an automatic car control system or, as a simpler alternative, a driver assisted by AUTOPASS can set the steering angle as desired. Under the assumptions presented above, a model for the in-plane motion is given in the following general form:

$$\begin{aligned} \dot{x}_1 &= \Psi_1(q, u_1, u_2), \\ \dot{x}_2 &= \Psi_2(q, u_1, u_2), \\ \dot{\phi} &= \Psi_3(u_1, u_2), \end{aligned} \tag{1}$$

where $\{\Psi_i \,|\, i = 1, 2, 3\}$ are nonholonomic nonlinear functions that can be obtained based on [22]; the car body center coordinates x_1 and x_2 and the yawing angle ϕ comprise the system state vector $q = \mathrm{col}(x_1, x_2, \phi)$; the driving speed u_1 and the steering angle u_2 are regarded as two control inputs.

Let us regard a car parking problem as the problem of driving the car from an initial state q^i into a desired (parked) state q^d. The initial state is defined as

$$q(t=0) = [\ x_1(0)\ x_2(0)\ \phi(0)\ ]^{\top} \triangleq q^i \in \mathcal{D}, \tag{2}$$

where $\mathcal{D}$ is a domain of possible initial configurations of the car. In this paper, we pursue an engineering approach to the control system design: instead of proposing another general solution for the car control problem, we are designing a control system which is capable of reliable parking process control in the real world. Following the normal driving practice, we consider an initial position of the car to be close to a standard one (i.e. the position from which parking is practically feasible). In other words, the initial configuration domain (2) is a relatively small bounded set. We also require the parking process to be completed by a single movement, without backward and forward iterative movements. For the clarity of exposition, we concentrate in this paper on the backward parking and assume that $u_1 < 0$ during the parking process. The modification of the proposed algorithm for the forward car motion ($u_1 > 0$) is straightforward.

The general form of equation (1) is such that a change of the time scale is equivalent to scaling the driving velocity u_1 [10]. In the practical sense, this property means that one can drive the car along the same path with an arbitrary varying driving velocity. Therefore, without loss of generality we can assume that $u_1 = \text{const} < 0$. Hence, the *problem of parking control* reduces to the problem of finding the steering angle input $u_2(t)$ that drives the system to the desired state q^d.

Major factors of complexity in finding the steering input $u_2(t)$ are the requirements of boundedness of the steering angle and avoidance of obstacles (e.g., other parked cars). Let u_* be the maximal steering angle. We will call the initial position q^i *admissible*, if there exists a bounded steering angle control input $\{u_2(\cdot) \in C^1; |u_2(t)| \leq u_*\}$ that drives the car into the desired state q^d under the constraint $u_1 = \text{const} < 0$. To fulfill the two requirement stated above, the problem of parking control is divided into two inter-linked sub-problems:

Problem 1: Check the admissibility of a given initial state $q^i \in \mathcal{D}$.

Problem 2: If q^i is admissible, find a steering angle control input $u_2(t)$ that solves the problem of parking control.

The domain of admissible initial configurations of a car usually has a complex shape and, therefore, a solution to Problem 1 is not trivial. At the same time, finding a solution of Problem 1 is indispensable, as a parking assistance system must be capable of immediately advising the driver that parking from the current position is impossible.

The solution for Problems 1 and 2 embodied into AUTOPASS is based on the fact that the proposed network architecture is capable of solving fast the problem of finding an optimal steering angle input $u_2(t)$ that drives the system to a desired state q^d. Therefore, at first AUTOPASS computes the optimal (i.e., minimal-magnitude) steering angle input $u_2(t)$ that drives the system to the desired state q^d. If the computed optimal steering angle input exceeds the feasible bounds, the initial state is reported by AUTOPASS to be not admissible. Otherwise, the obtained steering angle input $u_2(t)$ is used as the solution to Problem 2.

If we apply the computed steering angle input $u_2(t)$ to a real-world car, the car is most likely to reach the final state q^d with an error due to perturbations and imprecision in the car dynamic model. Following the standard control engineering practice, we compensate for this error by computing a steering angle input as a sum of the feedback and feedforward terms. The AUTOPASS control sub-system, which stabilizes the car motion along a reference trajectory despite disturbances and imprecise modeling, is based on the efficient combination of feedforward and feedback control, as described in [15].

The solution of Problems 1 and 2, as well as the computation of the feedforward term are carried out with the help of a specially designed RBF network architecture described in the next section.

3 APPROXIMATING A PARAMETRIC FAMILY OF CAR PARKING PROGRAMS

In this section, we present the most interesting and innovative aspect of AUTOPASS controller design. We consider a general problem of computing a parametric family of feedforward control programs U and reference trajectories Q that solve the parking control problem.

3.1 The problem formulation

The parking motion patterns considered in this paper depend on the 3-dimensional task parameter vector p of the form:

$$p = \begin{bmatrix} x_1^i - x_1^d \\ x_2^i - x_2^d \\ \phi^i - \phi^d \end{bmatrix} = q^i - q^d, \qquad q^i = \begin{bmatrix} x_1^i \\ x_2^i \\ \phi^i \end{bmatrix} = q\,(t=0). \tag{3}$$

The parameter vector (3) depends on the initial and desired values of vector x (i.e., $x\,(0)$ and $x\,(t)$ respectively) for the control problem formulated in Section 2. The control task considered in Section 2 can be described in terms of the parametric family of input/output nonlinear mappings and the desired parameter-dependent state vector $q^d(p)$. Based on the technique proposed in [34] and further elaborated in [11], we can compute optimal control input vectors $U_*(p)$ for some (discrete) values of the task parameter vectors p. Using this finite set of control programs, we can obtain a solution for any value of p by solving the following *approximation problem*:

Approximation problem: *Given the sets of vectors* $p^{(k)}$ *and* $U_*^{(k)}$, *and matrices* $Q^{(k)}$ as $\{p^{(k)},\, U_*^{(k)} = f(p^{(k)}),\ Q^{(k)} = g(p^{(k)})\}_{k=1}^{K}$ *find an approximation for the smooth mappings* $U = f\,(p)$ and $Q = g\,(p)$ *over a given domain of the task parameter vector* p.

In the ANN literature, the sets $\{p^{(k)}, U_*^{(k)}\}_{k=1}^{K}$ and $\{p^{(k)}, Q^{(k)}\}_{k=1}^{K}$ are called *training sets*. In the following section we propose a solution to the Approximation problem stated above.

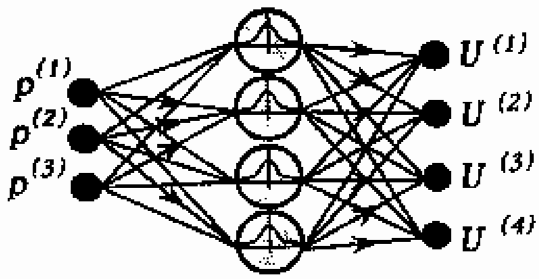

Figure 2: The Radial Basis Function network.

3.2 Radial Basis Function network architecture

The Approximation problem above comprises a vector-valued and a matrix-valued mappings $p \mapsto U$ and $p \mapsto Q$, whereas most of the previous work on similar problems concentrate on scalar-valued mappings.

We solve the approximation problem with the help of a Radial Basis Function (RBF) network approximation. RBF approximation has been recently demonstrated to provide better precision and training rates than many other ANN and Associative Memory approximation techniques. For example, the RBF approximation has shown to be superior in accuracy and training time to Multilayered Perceptron Networks used in many ANN-based systems [4], [5], [12], [18], and [24]. Radial Basis Function networks were recently acknowledged to possess advantageous spatial filtration properties [27], [29], and [30].

For AUTOPASS, we use a matrix form of an RBF network that is particularly efficient for solving multi-variable vector- and matrix-valued approximation problems, such as that stated in the previous section. The RBF network is shown schematically in Figure 2. This network has a single hidden layer of nodes with a bell-shaped activation function.

Several forms of the radial functions are used in RBF networks. For AUTOPASS design, we choose to use the *Gaussian function* that was recently proved to have advantageous spatial filtration properties [30]. The Gaussian radial function has the following form $h(r) = \exp\left(-r^2/d^2\right)$, where the parameter d is the interaction radius. In numerical simulations, we have obtained the best results with the interaction radius d chosen to be 1.6 times the distance between the network nodes $p^{(j)}$. We compute the vector weights $W^{(j)}$ of the RBF network from the *exact interpolation conditions*, so that the approximation error is zero for each pair $\{p^{(j)}, U^{(j)}\}$ in the training set. Based on [23], the weights $W^{(j)}$ of the RBF network can be computed using the known results from the advanced linear algebra theory.

3.3 Approximating the reference trajectory

The AUTOPASS controller requires that a reference (parking) trajectory, which determines the feedforward control, is specified. This reference trajectory is defined by the matrix Q. For the node centers $p^{(n)}$, matrices $Q^{(n)}$ are computed in the course of the iterative optimization procedure along with the control vectors $U^{(n)}$. By storing these matrices, we obtain the training set $\left\{p^{(n)}, Q^{(n)}\right\}_{n=1}^{K}$ and the approximation problem similar to the one considered above. Therefore, for each column of the matrix Q we can apply an RBF approximation method of the form considered in the previous section.

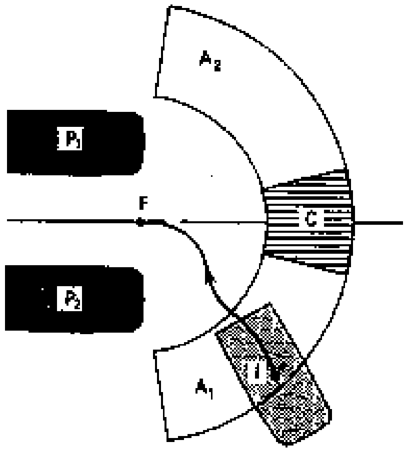

Figure 3: Initial configuration domain in regular parking.

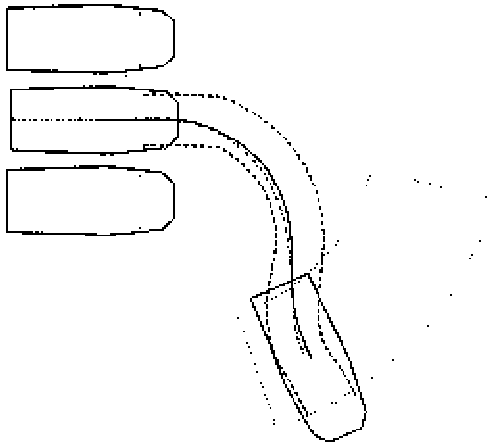

Figure 4: Backward regular parking using AUTOPASS.

4 IMPLEMENTING THE PARKING CONTROL SYSTEM

The most basic control implemented in AUTOPASS – parallel parking presented in Figure 1(a) – was described in [15]. In this section we describe the design of a *backward parking* control also implemented in AUTOPASS. The layout of the regular (backward) parking problem is presented in Figure 1(b). The goal is to park the car in the parking slot between two previously parked cars P_1 and P_2. The task is often considered as stressful and difficult for many car drivers.

AUTOPASS is designed to assume the parking control when the car is located in the ring segment shown in Figure 3. The segment center is located at the point F on the centerline in between the neighboring cars P_1 and P_2. Initial orientation of the car is such that its rear is aimed in the direction of point F with some allowed deviation. Such a layout of the initial configuration domain $\mathcal{D}$ (2) is based on three main considerations regarding the parking task in question. Firstly, it is impossible to carry out the parking if the controlled car is initially either located too close to the parked cars, or oriented with too large a deviation from the direction to point F. Secondly, there is no sense in starting an automated parking control too far from the desired parking slot. Thirdly, the initial configuration domain $\mathcal{D}$ outlined in Figure 3 allows reliable recognition of the desired parking slot with the help of sensors mounted at the rear of a car.

AUTOPASS divides the parking process into two phases. The first, more difficult and important phase is to place the car center close to the point F while the car body axis is approximately parallel to (or coincides with) the centerline of the parking slot. At the second phase, the car moves deeper in between the neighboring cars while keeping its centerline close to the centerline of the parking slot. This second phase is carried out using the feedback control of the car motion along the centerline of the parking slot, as described in Section 2.

Let us elaborate the first phase of the parking process. We classify possible initial positions of the car into three sub-cases that correspond to initial locations of a car in the segments A_1, A_2, and C (see Figure 3). If a car is initially located in the segment C, adjacent to the parking slot centerline, the entire parking control task is solved with the help of the feedback tracking (as described in Section 2) of the centerline regarded as a reference trajectory. The feedback gains and the size of the segment C are chosen so that the transient process is completed before the car arrives to the point F.

The most sophisticated control design is required if the parking is to start from the segments A_1 and A_2. AUTOPASS solves this complicated problem by first designing the sub-optimal steering control feedforward program and the reference trajectory as described in Section 3, and then tracking the reference trajectory as described in Section 2. Instead of computing each parking control program from scratch, which can take about quarter an hour for 1 Mflops computer, AUTOPASS uses an RBF network architecture for approximating the desired parking control program as described in Section 3. The network computes approximations in less that 100 msec using the same computer. The network training is done off-line.

Figure 4 shows the parking emulation results for backward regular parking controlled by AUTOPASS. The dashed lines represent the front wheel track and the solid curve represents the car body center trajectory, and dash-dotted curve shows the reference parking trajectory computed by the network. The dotted line marks the initial configuration segment A_1.

The control algorithm described in this section can be applied, with minor modifications, to the parking problem shown in Figure 1(c). To make this fact clear, let us notice that a comparison of Figures 1(b) and 1(c) reveals the similarity of the parking task layout at the both sides of the parking slot centerline. Therefore, to solve the parking problem shown in Figure 1(c), one can use the controller design described in this section, with the only modification that the upper and lower segments of the initial configuration domain in Figure 3 are shifted along the parking slot centerline in the opposite directions.

5 CONCLUSIONS

In this paper, we have presented a controller design for an automated parking support system (AUTOPASS). To design AUTOPASS, we have performed a comprehensive analysis of the parking problem, and desired features of a parking support system and a parking controller. The proposed controller design is based on the Radial Basis Function (RBF) architecture for the reference trajectory planning and feedback/feedforward control arrangement for tracking of this reference trajectory.

We have demonstrated that AUTOPASS can deal not only with the simplest case of parallel parking control, but with more complicated cases of car parking as well. The car parking emulation results demonstrate that the proposed control architecture yields an elegant and sufficiently accurate solution to the automated parking control problem. AUTOPASS can be implemented on a typical on-board computer of a passenger car.

This paper has concentrated on the AUTOPASS control algorithm. Other pertinent issues, such as sensory-based data acquisition and processing, and design of the AUTOPASS driver interface are currently being studied. Furthermore, the design is currently in progress to build on-line learning ability into AUTOPASS. This learning ability will allow for the adaptation of parking programs and the tracking controller to the inaccuracy and changes in time of the vehicle dynamic model and the external environment conditions.

REFERENCES

[1] Bloch, A.M., Reyhanoglu, M., and McClamroch, N.H. "Control and stabilization of nonholonomic dynamic systems," *IEEE Trans. on Automatic Control*, vol.37, no.11, pp.1746-1757, 1992.

[2] Brockett, R.W. Asymptotic stability and feedback stabilization, in *Differential Geometric Control Theory*, (Birkhauser, Basel-Boston,) 181-191, 1983.

[3] Brockett, R.W. "Control theory and singular riemannian geometry," in P. Hilton and G. Young, Eds., *New Directions in Applied Mathematics*, Springer-Verlag, 1981.

[4] Broomhead, D.S., and Lowe, D. "Multivariable functional interpolation and adaptive networks," *Complex Systems*, no.2, pp.321–355, 1988.

[5] Chen, S., and Billings S.A. "Neural networks for nonlinear dynamic system modelling and identifications," *Int. J. of Control*, vol.56, no.2, pp.319-346, 1992.

[6] Chen, S., Cowan, C.F.N., and Grant, P.M. "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. on Neural Networks*, vol.2, no.2, pp.837–863, 1991.

[7] Coron, J.M. "Global asymptotic stabilization for controllable systems without drift," to appear in *Math. Control Sign. Systems*, 1992.

[8] Fernandes, C., Gurvits, L., and Li, Z.X. *Foundations of nonholonomic motion planning*, Technical report TR 577 RR 253, Robotics Research Laboratory, Courant Institute of Mathematical Sciences, 1991.

[9] Gorinevsky, D.M. "Adaptive learning control using Radial Basis Function network approximation over task parameter domain," *Proc. IEEE Int. Symp. on Intelligent Control*, Chicago, IL, August 1993.

[10] Gorinevsky, D.M., Kapitanovsky, A., and Goldenberg, A.A. "Control of nonholonomic space systems using Radial Basis Function network," *ASME Winter Annual Meeting, Proc. of the Symp. on Advances in Robust and Nonlinear Control*, New Orleans, LA, 1993.

[11] Gorinevsky, D.M. "On the approximate inversion of linear system and quadratic-optimal control," *J. Comput. Syst. Sci. Int.*, vol.30, no.6, pp.6-23, 1992.

[12] Gorinevsky, D.M., and Connolly, T.H. "Comparison of some neural network and scattered data approximations: The inverse manipulator kinematics example," *Neural Computation*, (to appear), 1993.

[13] Gorinevsky, D.M. "Learning and approximation in database for feedforward control of flexible-joint manipulator," *ICAR '91: 5th Int. Conf. on Advanced Robotics*, pp.688–692, Pisa, June 1991.

[14] Gorinevsky, D.M. "Experiments in direct learning of feedforward control for manipulator path tracking," *Robotersysteme*, vol.8, pp.139–147, 1992.

[15] Gorinevsky, D.M., Kapitanovsky, A., and Goldenberg, A. "An architecture for control of car motion and application to parking," *Proc. of the 4th Symp. on Advanced Automotive Technologies, ASME Winter Annual Meeting*, New Orleance, Dec. 1993.

[16] Gurvits, L., and Li, Z.X. Smooth time-periodic feedback solutions for nonholonomic motion planning, in Z.X. Li and J. Canny, eds., *Progress in nonholonomic motion planning*, (Kluwer Academic Publishers), 1992.

[17] Haynes, G.W., and Hermes, H. "Nonlinear controllability via Lie theory," *SIAM J. Contr. and Optimiz.* 8(4) 1970.

[18] Kadirkamanathan, V., Niranjan M., and Fallside F. "Sequential adaptation of radial basis function neural networks and its application to time-series prediction," Moody, J.E., Lipmann, R.P., and Touretzky, D.S., ed., *Advances in Neural Information Processing Systems 3*, vol. 3, pp. 721–727, Morgan Kaufmann, San Mateo, CA, 1991.

[19] Kapitanovsky, A., Goldenberg, A., and Mills, J. "Dynamic control and motion planning technique for a class of nonlinear systems with drift," *Systems and Control Letters*, Vol. 21, pp. 363-369, 1993.

[20] Kapitanovsky, A., Goldenberg, A., and Mills, J. "Design of smooth dynamic feedback for stabilization of nonholonomic systems," *Proc. of the 32nd IEEE Conf. on Decision and Control*, pp. 965-970, San Antonio, TX, 1993.

[21] Krasnoselsky, M.A., and Zabreiko, P.P. *Geometric Methods of Nonlinear Analysis*, Springer Verlag, NY, 1984.

[22] Lafferriere, G., and Sussmann, H. *Motion planning for controllable systems without drift: A preliminary report*, Rutgers University Systems and Control Center Report SYCON-90-04, 1990.

[23] Micchelli, C.A. "Interpolation of scattered data: Distance matrices and conditionally positive definite functions," *Const. Approx.*, vol.2, pp.11–22, 1986.

[24] Moody, J., and Darken, C.J. "Fast learning in networks of locally-tuned processing units," *Neural Computation*, vol.2, no.1, pp.281–284, 1989.

[25] Murray, R.M., and Sastry, S.S. "Nonholonomic motion planning: Steering using sinusoids," *IEEE Trans. on Automatic Control*, vol.38, no.5, pp.700-716, 1993.

[26] Nguyen, D.H., and Widrow, B. "Neural networks for self-learning control systems," *Int. J. Control*, vol. 54, no. 6, pp. 1439-1451, 1991.

[27] Poggio, T., and Girosi, F. "Networks for approximation and learning," *Proc. of the IEEE*, vol. 78, no.9, pp.1481-1497, 1990.

[28] Pomet, J.-P. "Explicit design of time-varying stabilizing control laws for a class of controllable systems without drift," *Systems and Control Letters* 18, 147-158, 1992.

[29] Powell, M.J.D. "The theory of Radial Basis Function approximation in the 1990," Chapter 2 in *Advances in Numerical Analysis*, (Light, W., ed.) vol. 2, Clarendon Press, Oxford, pp. 105-210, 1992.

[30] Sanner, R., and Slotine, J.-J.E. "Gaussian networks for direct adaptive control," *IEEE Trans. on Neural Networks*, vol.3, no.6, pp.837-863, 1992.

[31] Samson, C., and Ait-Abderrahim, K. "Feedback stabilization of a nonholonomic wheeled mobile robot," *Proc. IEEE Int. Workshop on IROS*, pp. 1242-1247, 1991.

[32] Sontag, E.D. Feedback stabilization of nonlinear systems, in M.A. Kaashoek et al., ds., *Robust Control of Linear Systems and Nonlinear Control, Proc. of the Int. Symp. MTNS-89*, (Birkhauser, Basel-Boston), 61-81, 1990.

[33] Sussmann, H., and Liu, W. *Limits of highly oscillatory controls and the approximation of general paths by admissible trajectories*, Rutgers University Systems and Control Center Report SYCON-91-02, 1991.

[34] Tikhonov, A.N., and Arsenin, V.Y. *Methods for Solution of Ill-poses Problems*, Washington D.C., 1977.

[35] Vafa, Z. "Space manipulator motion with no satellite attitude disturbances," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 1990.

[36] Vlassenbroeck, J., and Rene, Van D. "A Chebyshev technique for solving nonlinear optimal control problems," *IEEE Trans. on Automat. Cont.*, vol.33, no.4, pp.333–340, 1988.