EXPERIMENTAL KINEMATIC COMPARISON OF BEHAVIORAL APPROACHES FOR MOBILE ROBOTS

Gianluca Antonelli Filippo Arrichiello Stefano Chiaverini

Dipartimento di Automazione, Elettromagnetismo,
Ingegneria dell'Informazione e Matematica Industriale,
Università degli Studi di Cassino
Via G. Di Biasio 43, 03043 Cassino (FR), Italy
`{antonelli,f.arrichiello,chiaverini}@unicas.it`

Abstract: An experimental comparison among the main behavioral control techniques for autonomous mobile robots is presented in this paper. The behavioral algorithms tackle the problem in a similar way: they decompose the mission planning of the robot in simple tasks; each task is then approached independently from the others and, at the end, the behavioral algorithms arrange the tasks' output to give an univocal motion command to the robot. The algorithms differ mainly in the latter point: the way they manage the output of the tasks. Three of these approaches have been implemented on the autonomous mobile robot Khepera II to show the main differences in a practical case. *Copyright ©2005 IFAC*

Keywords: Behavioral Control, Mobile Robots, Path Planning

1. INTRODUCTION

During a mission, autonomous mobile robots need to achieve several tasks such as exploration, map building, moving through predefined points and, simultaneously, avoiding obstacles and preserve robot's integrity. Thus, even a simple mission can be seen as composed by elementary sub-problems known in literature as *tasks*, *behaviors*, *functional modules* or *motor schemas*. We will use these terms like synonyms and we will call *behavioral approaches* any method aimed at properly composing elementary behaviors.

Among the behavioral approaches, seminal works are reported in the papers Brooks (1986) and Arkin (1989). A behavioral approach designed for exploration of planetary surfaces has been investigated in Gat et al. (1994), while in Langer et al. (1994) the experimental case of an off-road navigation is presented. Lately, behavioral approaches have been applied to the formation control of multi-robot systems as in, e.g., Balch and Arkin (1998). The use of techniques inherited from inverse kinematics for industrial manipulators is described in Bishop (2003); Antonelli and Chiaverini (2003); Yang et al. (2003). In Saffiotti and Wasik (2003) a hierarchical behavior-based system that performs several vision-based manipulation tasks by using different combinations of the same set of basic behaviors is presented.

In this paper a behavior is expressed through a function of the robot configuration that measures the degree of fulfilment of the task (e.g., a cost or a potential function); thus, in a static environment the task is achieved when its output is constant at a value that minimizes the task function. In presence of multiple behaviors, each task output may achieve its specific goal, but there is no guarantee that a single command to the robot can accomplish all the assigned behaviors at the same time. In particular, when a motion command cannot reduce simultaneously all the task functions there is a *conflict* among the tasks that must be solved by a suitable policy.

The aim of this paper is to underline the different compositions of the tasks in case they need to be achieved at the same time and conflict one with the other.

In the following three different behavior control techniques that solve, in different ways, the combination of multiple behaviors, are compared, namely:

- *Layered control system;*
- *Motor schema control;*
- *Null-space-based control.*

The mission taken into consideration is a move to goal task in a very simple scenario with only one obstacle present; several experiments have been run with the mobile robot Khepera II.

2. COMPARED ARCHITECTURES

In this section the previously cited behavioral approaches are described. These approaches differ in how they work when multiple tasks have to be achieved. This class of problems, known in literature as *Behavioral Coordination* (Arkin (1998)), include two main different approaches, namely, *Competitive Methods* and *Cooperative Methods*. In the competitive methods, on the basis of sensors information, only one task is instantaneously active and the control tries to solve only the chosen task. In the cooperative methods, instead, a supervisor elaborates each task and gives as output an intermediate solution, calculated as the sum of all velocity vectors (one for each task) opportunely multiplied by a gain vector. The supervisor, on the the basis of sensor information, dynamically changes the gain vector, giving instantaneously more or less weight to each task.

The layered control system, proposed by Brooks (1986), is a classical example of a competitive method, while the motor schema control, proposed by Arkin (1989), is one of the cooperative methods. The null-space-based control, proposed by Antonelli and Chiaverini (2003), is a recently proposed approach that comprehends, as particular cases, the former two algorithms.

2.1 Layered Control System

The layered control system (Brooks (1986)) is an architecture for controlling mobile robots that lets the robot working at increasing levels of competence. Each task is related to a layer that is an asynchronous module that communicates over low-bandwidth channel. In particular, each behavior is represented using an augmented finite state machine (AFSM) model. On the basis of sensors information (or *of the perception of the word*), each layer, working independently from the others, elaborates an output that is a motion command for the robot, i.e., a direct input to the actuators or a desired velocity vector to give to the low-level control.

Layers have different priority levels and higher-level layers can subsume the lower levels by suppressing their outputs. With this hierarchic architecture, the higher-priority task is always pursued and, on the basis of the instantaneous context, the task to be executed dynamically changes. With this approach, only the higher task is properly achieved and it is possible to add new layers simply choosing the positions in the hierarchy. In the case of tasks conflicting one each other, the higher priority task subsumes the lower one.

This architecture, known in literature also as *subsumption architecture*, needs the use of a priority-based coordination function. The function can take the form of a fixed prioritization network in which a strict behavioral dominance hierarchy exists. This architecture uses a subsumption language (Behavior Language) that is an abstraction of the AFSMs using a single rule set to encode each behavior.

In the older version of this algorithm the overall robot behavior is simply achieved by collecting elementary behaviors; later, an intermediate level has been proposed and the overall robot behavior is obtained by selecting an abstract behavior that is, itself, a collection of elementary behaviors (Arkin (1998)).

From a practical aspect, the subsumption can follow two mechanisms: the inhibition, used to prevent that a signal is transmitted to the actuators, and the suppression, in which a signal is replaced by an higher-priority one.

2.2 Motor Schema Control

The motor schema control (Brooks (1986)), strongly motivated by the biological sciences (as neuroscientific, psychological and robotic sources), is one of the cooperative methods for behavioral approaches. A motor schema is "*the basic unit of motor behavior from which complex actions can be constructed. It consists of both the knowledge of how to act as well as the computational process by which it is enacted*" (Arkin (1989)). The output of a navigation motor schema is a vector representing the desired advancing velocity vector. The navigation behavior is obtained by the proper combination, by an high-level supervisor, of the task functions. This architecture, thus, needs the presence of a supervisor that dynamically selects the active behaviors and outputs the robot motion command by a normalized sum of the active tasks.

The supervisor realizes the selection of active behaviors properly changing the weights of each task. This approach implements a hierarchy among the tasks, where a null weight deactivates a behavior while the active ones are taken into account in proportion to the respective weight factor. The weights are usually context-dependent (depending on word perception and interpretation) and, thus, can dynamically change the relative priority of the tasks.

The main difference with the competitive methods is that this approach realizes a linear combination of the outputs elaborated for each task. In this way no task is completely achieved but, on the basis of gain vectors, a compromise solution is found.

2.3 Null-Space-Based Control

The null-space-based approach for mobile robots arises from a strategy devoted at kinematic control for redundant industrial manipulators proposed in Chiaverini (1997). This approach, presented in Antonelli and Chiaverini (2003, 2004) for coordinated control of platoons of autonomous vehicles, is specially designed for kinematic control of mobile robots to achieve simultaneously multiple tasks. At the best of our knowledge, the first idea to use technique inherited from inverse kinematics of redundant manipulator is given in Bishop (2003); Bishop and Stilwell (2001) where a different merging of the tasks' output is proposed.

Each task is described by a variable $\sigma \in \mathbb{R}^m$ expressed by a function $\boldsymbol{f}$ of the robot position $\boldsymbol{p} \in \mathbb{R}^3$:

$$\boldsymbol{\sigma} = \boldsymbol{f}(\boldsymbol{p}) \tag{1}$$

and by the corresponding differential relationship:

$$\dot{\boldsymbol{\sigma}} = \frac{\partial \boldsymbol{f}(\boldsymbol{p})}{\partial \boldsymbol{p}} \boldsymbol{v} = \boldsymbol{J}(\boldsymbol{p})\boldsymbol{v}\,, \tag{2}$$

where $\boldsymbol{J} \in \mathbb{R}^{m\times 3}$ is the configuration-dependent task Jacobian matrix and $\boldsymbol{v} \in \mathbb{R}^3$ is the vehicle velocity in a general 3-dimensional motion.

The proposed approach finds an inverse solution to the kinematic problem (1) starting from the desired values $\boldsymbol{\sigma}_d(t)$ of the task function. Thus, inverting the (locally linear) equation (2) the reference desired position $\boldsymbol{v}_d(t)$ is obtained. The Jacobian matrix is a task depending matrix and, in general, it is a low-rectangular matrix, thus a unique solution to the inverse of problem (2) does not exist. A typical requirement is to purse minimum norm velocity, leading to the least-squares solution:

$$\boldsymbol{v}_d = \boldsymbol{J}^\dagger \dot{\boldsymbol{\sigma}}_d = \boldsymbol{J}^{\mathrm{T}} \left(\boldsymbol{J}\boldsymbol{J}^{\mathrm{T}}\right)^{-1} \dot{\boldsymbol{\sigma}}_d\,. \tag{3}$$

At this point, the vehicle motion controller needs a reference position trajectory besides the velocity reference; this can be obtained by time integration of $\boldsymbol{v}_d$. However, discrete-time integration of the vehicle's reference velocity would result in a numerical drift of the reconstructed vehicle's position; the drift can be counteracted by a so-called Closed Loop Inverse Kinematics (CLIK) version of the algorithm, namely,

$$\boldsymbol{v}_d = \boldsymbol{J}^\dagger \left(\dot{\boldsymbol{\sigma}}_d + \boldsymbol{\Lambda}\widetilde{\boldsymbol{\sigma}}\right), \tag{4}$$

where $\boldsymbol{\Lambda}$ is a suitable constant positive-definite matrix of gains and $\widetilde{\boldsymbol{\sigma}}$ is the task error defined as $\widetilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}_d - \boldsymbol{\sigma}$. This solution is relative to only one task. When two or more tasks need to be achieved at the same time a geometric approach is proposed. This technique consists in assigning a relative priority to each task and using a null space projection operation to combine higher priority task with lower ones. The idea is to use all the motion capability that does not affect the higher-priority task to fulfill the lowers.

In case of multiple tasks, the contributions of each task are calculated following the formula (4) and, starting from the lowest level, each contribution has to be properly projected onto the null-space of the higher-priority task. Denoting with $\boldsymbol{v}_i$ the generic contribute of the i-th task quantity

$$\boldsymbol{v}_i = \boldsymbol{J}_i^\dagger \left(\dot{\boldsymbol{\sigma}}_{i,d} + \boldsymbol{\Lambda}_i\widetilde{\boldsymbol{\sigma}}_i\right), \tag{5}$$

and supposing that i denotes also the priority order, then, the formula for three tasks is:

$$\boldsymbol{v}_d = \boldsymbol{v}_1 + \left(\boldsymbol{I} - \boldsymbol{J}_1^\dagger \boldsymbol{J}_1\right)\left[\boldsymbol{v}_2 + \left(\boldsymbol{I} - \boldsymbol{J}_2^\dagger \boldsymbol{J}_2\right)\boldsymbol{v}_3\right]. \tag{6}$$

This approach, whose sketch is given in Figure 1, differently from the competitive methods and the comparative method, pursues the highest-priority task and, when possible, fulfill the lower-priority tasks.

It is worth noticing that, with this approach, the dimension of the tasks, independently from their priority, that can be fulfilled at the same time is equal to the degrees of freedom of the system: 2 for a ground mobile robots. Obviously, this reduces the number of tasks that can be fulfilled. However, in case of multi-robot systems, the degrees of freedom significantly increase and a large number of tasks in a complex scenario can be fulfilled. In Antonelli and Chiaverini (2004), a caging mission of multiple robots, with the presence of obstacles and failure of vehicles is performed with this approach.

2.4 Layered control system and motor schema control as particular cases of null-space-based control

When the tasks handled by a layered control system or by a motor schema control have a smooth analytic expression, so that the relative Jacobian matrices can be computed, it can be shown that the former schemes can be seen as particular cases of the null-space-based control.

By considering the extended task vector obtained by stacking all the single defined task vectors

$$\boldsymbol{\sigma} = \left[\boldsymbol{\sigma}_1^{\mathrm{T}} \ldots \boldsymbol{\sigma}_n^{\mathrm{T}}\right]^{\mathrm{T}}, \tag{7}$$

the corresponding extended Jacobian matrix can be written as:

$$\boldsymbol{J} = \mathrm{diag}\{\boldsymbol{J}_1, \ldots, \boldsymbol{J}_n\}\,. \tag{8}$$

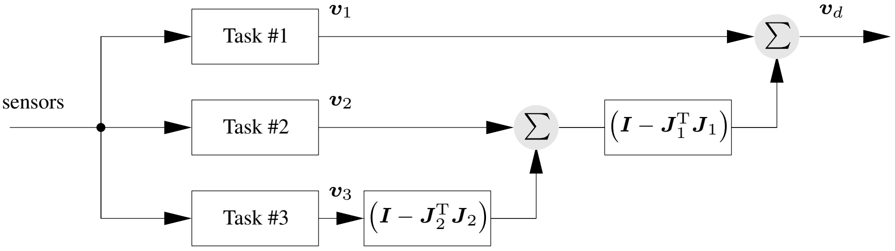

Fig. 1. Sketch of the null-space-based kinematic control in a 3-task example. A supervisor (not represented here) may change the relative priority of the tasks.

At this point, by using a weighted pseudoinverse in eqs. (3)–(4) with the weight matrix

$$\boldsymbol{W} = \mathrm{diag}\{\alpha_1, \ldots, \alpha_n\}\,, \tag{9}$$

the motor schema control is recovered when $\alpha_i \in [0, \infty]$ and the layered control system is obtained when $\alpha \in \{0, 1\}$ and $\sum \alpha_i = 1$. In fact, the layered control system satisfies one task selectively, while the motor schema control satisfies a weighted sum of the tasks.

3. EXPERIMENTAL RESULTS

In this section the experiments performed with a Khepera II mobile robot are presented. For each one of the proposed architectures a move to goal mission has been performed.

The overall mission is divided in two elementary tasks, namely:

- Task #1: *obstacle-avoidance;*
- Task #2: *move-to-goal.*

The obstacle-avoidance is the highest priority task in all the schemas because its achievement is of crucial importance to preserve the integrity of the vehicle. In presence of an obstacle in the advancing direction, its aim is to keep the robot on a safe distance from the obstacle. Thus, its implementation elaborates as output a velocity, in the robot-obstacle direction, that keep the robot to a safe distance from the obstacle. Therefore, it is:

$$\begin{aligned}\sigma_1 &= \|\boldsymbol{p} - \boldsymbol{p}_o\| \\ \sigma_{1,d} &= d \\ \boldsymbol{J}_1 &= \hat{\boldsymbol{r}}^{\mathrm{T}}\,,\end{aligned}$$

where $\boldsymbol{p}_o$ is the obstacle position and

$$\hat{\boldsymbol{r}} = \frac{\boldsymbol{p} - \boldsymbol{p}_o}{\|\boldsymbol{p} - \boldsymbol{p}_o\|}$$

is the unit vector aligned with the obstacle-to-vehicle direction. According to (6), the primary-task velocity is then

$$\boldsymbol{v}_1 = \boldsymbol{J}_1^{\dagger} \lambda_1 \left(d - \|\boldsymbol{p} - \boldsymbol{p}_o\|\right).$$

In the null-space-based approach, moreover, the obstacle avoidance task elaborates also the null-space direction, that is, in this implementation, the direction tangent to the circle. Thus, the velocity component of secondary task has to be projected along the tangent direction. The expression of the corresponding null space, required only for the third approach considered, is

$$\mathcal{N}(\boldsymbol{J}_1) = \boldsymbol{I} - \boldsymbol{J}_1^{\dagger} \boldsymbol{J}_1 = \boldsymbol{I} - \hat{\boldsymbol{r}}\hat{\boldsymbol{r}}^{\mathrm{T}}.$$

It is worth noticing that, in case of the layered control system, the obstacle avoidance task is active and subsumes the lower only close to the obstacles. Similarly, for the other two approaches, the supervisor activates this task when required.

The move-to-goal task is the same in all the schemas and its output is a velocity, in the goal direction, proportional to the distance from the goal $\boldsymbol{p}_g$; therefore, it is

$$\begin{aligned}\boldsymbol{\sigma}_2 &= \boldsymbol{p} \\ \boldsymbol{\sigma}_{2,d} &= \boldsymbol{p}_g \\ \boldsymbol{J}_2 &= \boldsymbol{I}\,.\end{aligned}$$

According to (6), the secondary-task velocity then is

$$\boldsymbol{v}_2 = \boldsymbol{\Lambda}_2 \left(\boldsymbol{p}_g - \boldsymbol{p}\right).$$

For all the algorithms, a saturation is performed on the resulting velocity in order to feed the system with limited amplitude signals.

The experimental set-up is composed by a unicycle-like mobile robot, namely the Khepera II manufactured by K-Team(see Figure 2).

The sampling time is 50 ms, the heading controller is derived from the controller reported in De Luca et al. (2000), the wheels' controllers is a PID developed by the manufacturer, a saturation of 5 cm/s and 130 deg/s have been introduced for the linear and angular velocities, respectively. The desired goal position is $[40 \quad -3]^{\mathrm{T}}$ cm, an obstacle has been considered in $[20 \quad -5]^{\mathrm{T}}$ cm. Since the experiments have been developed to test the algorithms, the obstacle is considered as perfectly known. Being the robot radius approximatively 4 cm, for the first task the safe distance to the obstacle has been set to $d = 8$ cm, while distance at which the obstacle can be recognized has been set to $d = 10$ cm to take into consideration the limited range of the sensors.

Fig. 2. Robot Khepera II of the K-Team.

For all the algorithms the gains have been kept constant, moreover, the priority of the tasks is also constant but the obstacle-avoidance task becomes active when the obstacle is closer than 10 cm to the vehicle and it lies in its advancing direction. The gains have been selected as

$$\lambda_1 = 10$$
$$\boldsymbol{\Lambda}_2 = 1\boldsymbol{I}.$$

In Figure 3, the path followed by the robot controlled by the layered control system is presented. It can be noticed that the robot, while approaching the obstacle, switches to the sole obstacle-avoidance task and resume the sole move-to-goal when far enough from the obstacle.

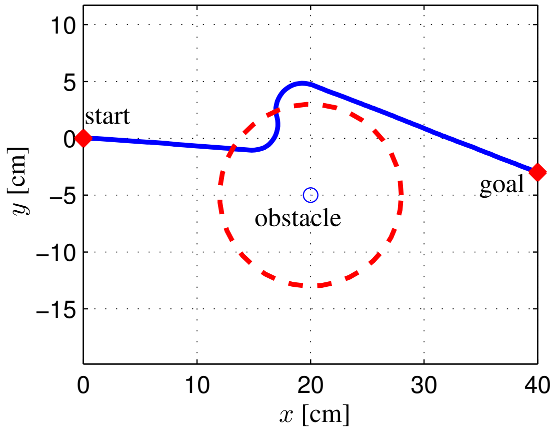

Fig. 3. Path followed under the layered control system for reaching a goal in presence of obstacle.

In Figure 4 the path obtained under the motor schema control is presented. It is worth noticing that the vehicle enters a 8 cm-circle around the obstacle since the algorithm outputs a linear combination of the two tasks. Obviously, the designer can modify the tasks' weights thus imposing a more or less conservative vehicle behavior.

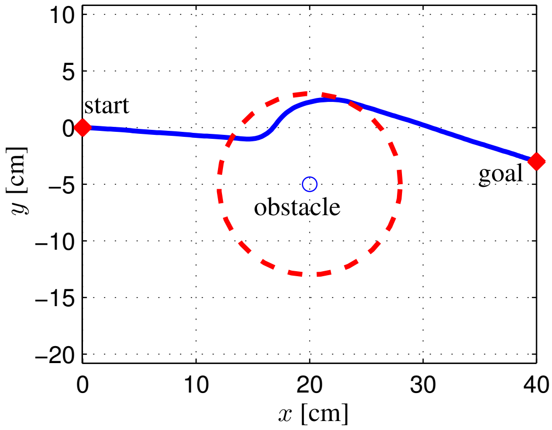

Fig. 4. Path followed under the motor schema control for reaching a goal in presence of obstacle.

In Figure 5, the path followed with the null-space-based control is shown. The geometric characteristic of this approach can be easily verified, during the obstacle avoidance, in fact, the vehicle keeps the safe distance from the obstacle exactly, thus fulfilling the higher priority task, and uses its null space in order to try to fulfill the lower priority. The overall motion is a clean *sliding* of the vehicle around the safe circle. In Figures 6 and 7 the corresponding measured values of the linear and angular velocities are shown.

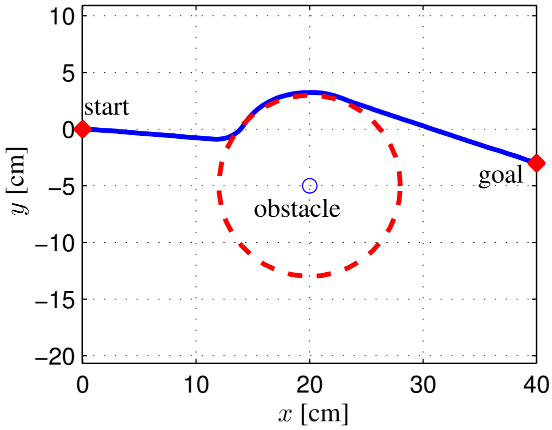

Fig. 5. Path followed under the null-space-based control for reaching a goal in presence of obstacle.

4. CONCLUSIONS

For autonomous robots, the layered control architectures, the motor schema control and the null-space-based behavior control have been implemented on an

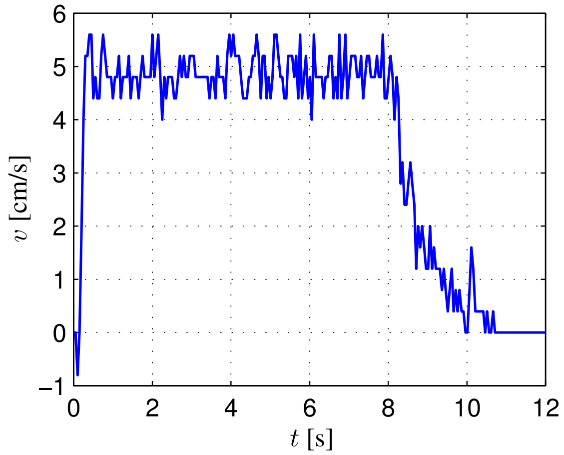

Fig. 6. Odometric measure of linear velocity of the robot for null-space-based control

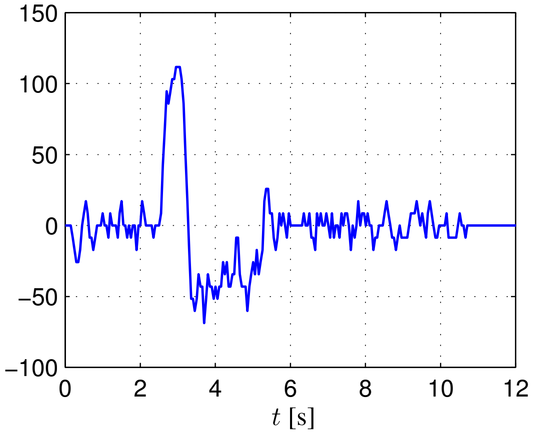

Fig. 7. Odometric measure of angular velocity of the robot for null-space-based control

experimental set-up and their performance evaluated. The comparison concerns principally the differences in the coordination of multiple-tasks and the kinematic interpretation of the different approaches. The null-space-based kinematic control ensures the achievement of the higher-priority task without being affected by the output of the lower-priority tasks, same as for the layered control system. In addition, by exploiting the null-space projection, it can fulfill simultaneously two or more tasks (depending on the task-space dimension) instead of only one, although dynamically selected. As a drawback, due to its analytical nature, the null-space-based kinematic control needs the definition of a suitable task function that admits computation of a proper Jacobian; this may be not obvious for some tasks.

REFERENCES

G. Antonelli and S. Chiaverini. Kinematic control of a platoon of autonomous vehicles. In *Proceedings 2003 IEEE International Conference on Robotics and Automation*, pages 1464–1469, Taipei, TW, Sept. 2003.

G. Antonelli and S. Chiaverini. Fault tolerant kinematic control of platoons of autonomous vehicles. In *Proceedings 2004 IEEE International Conference on Robotics and Automation*, pages 3313–3318, New Orleans, LA, April 2004.

R.C. Arkin. Motor schema based mobile robot navigation. *The International Journal of Robotics Research*, 8(4):92–112, 1989.

R.C. Arkin. *Behavior-Based Robotics*. The MIT Press, Cambridge, MA, 1998.

T. Balch and R.C. Arkin. Behavior-based formation control for multirobot teams. *IEEE Transactions on Robotics and Automation*, 14(6):926–939, 1998.

B.E. Bishop. On the use of redundant manipulator techniques for control of platoons of cooperating robotic vehicles. *IEEE Transactions on Systems, Man and Cybernetics*, 33(5):608–615, Sept. 2003.

B.E. Bishop and D.J. Stilwell. On the application of redundant manipulator techniques to the control of platoons of autonomous vehicles. In *Proceedings 2001 IEEE International Conference on Control Applications*, pages 823–828, México City, MX, Sept. 2001.

R.A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2:14–23, 1986.

S. Chiaverini. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *IEEE Transactions on Robotics and Automation*, 13(3):398–410, 1997.

A. De Luca, G. Oriolo, and M. Vendittelli. Stabilization of the unicycle via dynamic feedback linearization. In *6th IFAC Symposium on Robot Control*, pages 397–402, Wien, A, Sept. 2000.

E. Gat, R. Desai, R. Ivlev, J. Loch, and D.P. Miller. Behavior control for robotic exploration of planetary surfaces. *IEEE Transactions on Robotics and Automation*, 10(4):490–503, 1994.

K-Team. `http://www.k-team.com/`.

D. Langer, J.K. Rosenblatt, and M. Hebert. A behavior-based system for off-road navigation. *IEEE Transactions on Robotics and Automation*, 10 (6):776–783, 1994.

A. Saffiotti and Z. Wasik. A hierarchical behavior-based approach to manipulation tasks. In *Proceedings 2003 IEEE International Conference on Robotics and Automation*, pages 2780–2785, Taipei, Taiwan, Sept. 2003.

Y. Yang, O. Brock, and R.A. Grupen. Exploiting redundancy to implement multi-objective behavior. In *Proceedings 2003 IEEE International Conference on Robotics and Automation*, pages 3385–3390, Taipei, Taiwan, Sept. 2003.