A Motion Generation Approach for an Omnidirectional Vehicle

Igor E. Paromtchik*, Hajime Asama
The Institute of Physical and Chemical Research (RIKEN)
Hirosawa 2-1, Wako-shi, Saitama 351-0198, Japan

Abstract

This paper deals with the motion generation approach developed for our omnidirectional mobile robots. Our objective, overall control architecture of the mobile robot and its motion controller are considered. The key idea and advantages of the proposed motion generation approach are discussed. The approach is implemented and tested on the omnidirectional mobile robot and the experimental results obtained are presented. The operation of the control system is illustrated by a video on the remote control of the robot and the visually-coupled motion of the two robots.

1 Introduction

The objective of our research work is to develop a control system which ensures an accurate, fast and smooth motion of the omnidirectional mobile robot to a given goal. The system must be suitable for remote control (teleoperation) and capable to operate in the presence of delays in receiving the control commands. The control system is to provide the vehicle to track a specified object that can be detected and localized by processing visual data gathered by an on-board color CCD camera. The control system has to be capable of reactive and cooperative control in a multi-robot environment.

The control architectures of various complexity are known: a centralized architecture was considered in [1], a distributed behavioral architecture was proposed in [2], and an hybrid architecture was discussed in [3]. The hybrid architecture involves a functional distributed level and a centralized level and is well suited to operate with a combination of synchronous processes (e.g. servo-control) and asynchronous ones (e.g. processing visual data about the environment, remote control). Various methods of motion generation are developed and make use of spline-interpolation [4], e.g. polynomial spline-interpolation was considered in [5, 6]. Obtaining the reference trajectories which are feasible for the servo-systems of the robot was investigated in [7].

Our paper focuses on the motion generation approach as well as the control architecture developed and implemented in the control system of the omnidirectional mobile robot. The paper is organized as follows. The omnidirectional mobile robot and its kinematic model are presented in section 2. The architecture of the control system is discussed in section 3. The motion generation approach is considered in section 4. The experimental setup and results obtained on the omnidirectional mobile robot are discussed in section 5. The conclusions are given in section 6.

2 The Mobile Robot

The mobile robot is shown in Fig. 1. The robot is equiped with four omnidirectional wheels which allow it to perform motions in two directions and rotate simultaneously. The robot is equiped with three DC motors and a transmission mechanism to provide

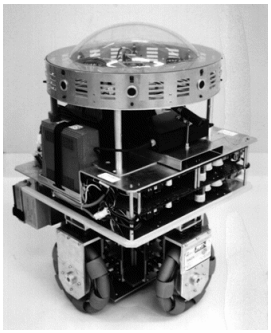

Figure 1: The omnidirectional mobile robot

the omnidirectional motion [8]. Three servo-systems execute the motion control commands issued by the control system of the robot and ensure attaining the

*Currently at the Institut National de Recherche en Informatique et en Automatique (INRIA), 655 avenue de l'Europe, 38330 Montbonnot Saint Martin, France

commanded position, orientation and velocity.

The sensor system of the robot involves a color CCD camera, eight sonar range sensors which gather data about the local environment in order to evaluate relative distances between the robot's frame and environmental objects. Processing the color data about the local environment allows it to detect and localize a specified object, e.g. another mobile robot. The sensor system also involves three encoders used in a feedback loop of the servo-systems.

The control system processes the data obtained from the sensor system and the servo-systems and issues the motion control commands which are subsequently executed by the servo-systems. The robot is capable to follow a specified object (e.g. a mobile robot) while keeping a safe distance to the object as a function of the motion velocity. The robot can also be controled by a remote operator from a graphical user interface [9]. In this case, the target positions are set by the operator and transmitted to the on-board control system of the robot via a wireless Ethernet. The infrared communication system is used for inter-robot communication when the robots are assigned to perform cooperative tasks [10].

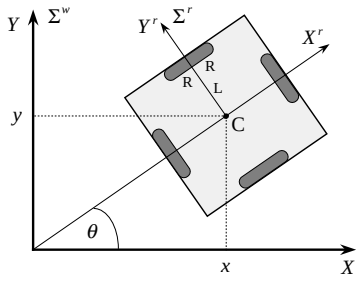

Figure 2: A kinematic model of the mobile robot

The kinematic model of the robot is depicted in Fig. 2. The coordinates of the robot relative to a reference coordinate system $\mathbf{\Sigma}^w$ are denoted as $\mathbf{x} = (x,\, y,\, \theta)^T$, where $x = x(t)$ and $y = y(t)$ are the coordinates of the intersection point C of the imaginary axes connecting the wheel centers of each pair of the parallel wheels, $\theta = \theta(t)$ is the orientation of the robot, and t is time. Let R denote the wheel radius and L denote the distance between the point C and the wheel center. The robot's velocity relative to $\mathbf{\Sigma}^r$ is

$$\dot{\mathbf{x}}^r = \mathbf{J}^r\, \dot{\mathbf{q}}, \tag{1}$$

where $\dot{\mathbf{x}}^r = (\dot{x}^r,\, \dot{y}^r,\, \dot{\theta}^r)^T$, $\dot{\mathbf{q}} = (\dot{q}_1,\, \dot{q}_2,\, \dot{q}_3)^T$ represents velocities of the actuators, and $\mathbf{J}^r$ denotes the Jacobi matrix with respect to $\mathbf{\Sigma}^r$:

$$\mathbf{J}^r = \begin{pmatrix} k_1 R & 0 & 0 \\ 0 & k_1 R & 0 \\ 0 & 0 & k_2 \frac{R}{L} \end{pmatrix}, \tag{2}$$

where k_1 and k_2 are the gear coefficients. Because $\mathbf{J}^r$ is diagonal, the translation along each direction and rotation with respect to $\mathbf{\Sigma}^r$ are decoupled and can be driven by the corresponding actuators [8]. The robot's velocity relative to $\mathbf{\Sigma}^w$ is

$$\dot{\mathbf{x}} = \mathbf{J}(\theta)\, \dot{\mathbf{q}}, \tag{3}$$

where $\mathbf{J}(\theta)$ is the Jacobi matrix with respect to $\mathbf{\Sigma}^w$:

$$\mathbf{J}(\theta) = \begin{pmatrix} k_1 R \cos\theta & -k_1 R \sin\theta & 0 \\ k_1 R \sin\theta & k_1 R \cos\theta & 0 \\ 0 & 0 & k_2 \frac{R}{L} \end{pmatrix}. \tag{4}$$

The equation (3) allows the evaluation of the translational and rotational velocities of the robot and the estimation of its location with respect to Σ^w. Equation (3) is valid for a vehicle moving on flat ground with a pure rolling contact without slippage between the wheels and the ground.

3 Control Architecture

The overall control architecture of the mobile robot is shown in Fig. 3. It involves three main parts: (1) - the vehicle with its actuators and sensors, (2) - the on-board control system, (3) - the remote control interface.

The reactive control ensures an autonomous operation of the robot in a dynamic environment [10, 11]. The collision avoidance algorithm operates with a rule matrix obtained from an adaptive behavior acquisition scheme which is based on reinforcement learning. The reactive control processes data gathered by the ultrasonic and infrared sensors as well as the color CCD camera. Based on the sensors configuration, eight possible directions of motion are considered [11, 12]. The reactive control algorithm ensures a collision-free motion to a given target position.

The motion controller provides coordinated translation and rotation of the omnidirectional vehicle when moving to a target position, as well as to attain a desired velocity of motion. The omnidirectional motion (holonomic case) or a constrained motion (nonholonomic case) can be performed according to the assigned task. For instance, vision-based tracking of a dynamic object requires the object to be kept in the

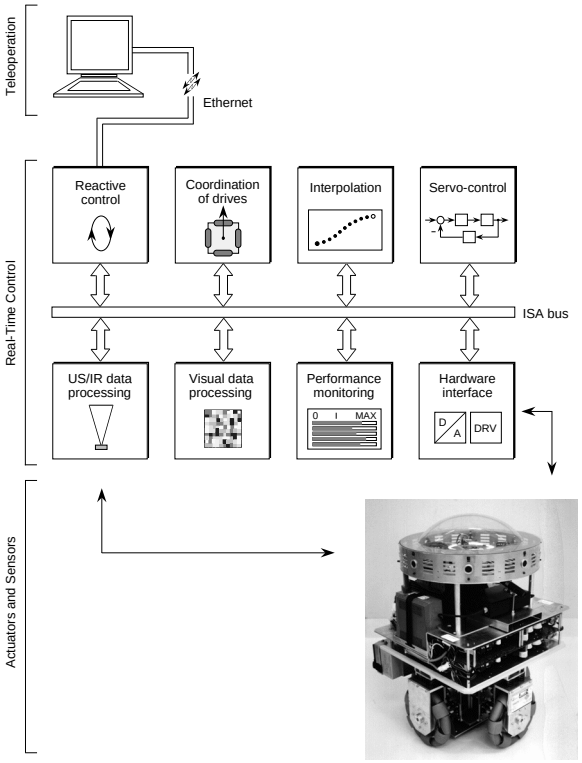

Figure 3: The overall control architecture

view field of the on-board CCD camera and is achieved by the coordinated control of the translational and rotational coordinates of the robot.

The motion controller is shown in Fig. 4, where: `SPL` denotes spline-interpolation, `SC` - servo-controllers, `D/A` - digital/analog converters, `M` denotes motors and $\mathbf{F}(\theta)$ is an orthogonal matrix of the coordinate transformation from $\mathbf{\Sigma}^r$ to $\mathbf{\Sigma}^w$:

$$\mathbf{x} = \mathbf{F}(\theta)\mathbf{x}^r. \tag{5}$$

The desired position $\mathbf{x}_d = (x_d,\, y_d,\, \theta_d)^T$ and motion velocity $\dot{\mathbf{x}}_d = (\dot{x}_d,\, \dot{y}_d,\, \dot{\theta}_d)^T$ are set by the operator or evaluated from the sensor data about the environment, e.g. during vision-based tracking of an object. The spline-interpolation is performed to achieve the coordination of the drives and generate the reference functions to control the position and orientation $\mathbf{x}_i = (x_i,\, y_i,\, \theta_i)^T$ and the velocity $\dot{\mathbf{x}}_i = (\dot{x}_i,\, \dot{y}_i,\, \dot{\theta}_i)^T$ of the robot. The servo-controllers ensure the accurate execution of the commanded reference functions.

The performance monitoring aims to increase the reliability and safety of the robot operation. The monitoring of measurable signals allows for fault detection and diagnosis in a closed-loop during the operation, e.g. measuring the capacity level of the electric batteries of the robot and, if needed, requesting the control system to interrupt execution of the on-going task and direct the robot to a location where the electric batteries can be replaced.

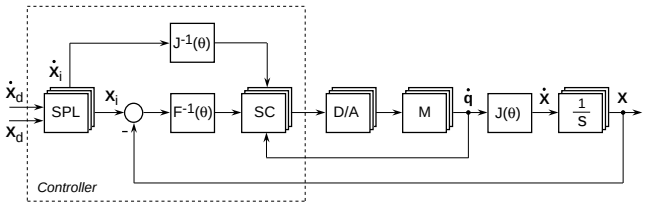

Figure 4: A motion controller of the robot

4 Motion Generation

The motion generation approach makes use of a modified cubic spline-interpolation [13]. In this section, the motion generation is discussed for one coordinate of the robot. In the control system, the approach is extended for the three coordinates. Let us consider a cubic polynomial

$$x(t) = \sum_{i=0}^{3} a_i \left(\frac{t}{T}\right)^i, \qquad 0 \le t \le T, \tag{6}$$

where a_i are real coefficients and $T > 0$ is an interpolation period. Let the boundary conditions be

$$\begin{cases} x(0) = x_0, & x(T) = x_T, \\ \dot{x}(0) = \dot{x}_0, & \dot{x}(T) = \dot{x}_T \end{cases} \tag{7}$$

and $\Delta x_T = x_T - x_0$.

Key idea. Let us study the case, when for $t \ge 0$ the coefficients a_i are recomputed with a sampling period $T^\star$ such that $T_s < T^\star \ll T$ while $T_s > 0$ and T_s, T, $\dot{x}_T$, Δx_T are constant. Let the appropriations $x_0 := x(kT^\star)$, $\dot{x}_0 := \dot{x}(kT^\star)$, $x_T := x(kT^\star) + \Delta x_T$, $k = 1, 2, \ldots$ ensure the continuity of $x(t)$ and $\dot{x}(t)$ at the boundary between subsequent $(k-1)$ and k computations.

The goal is to obtain a smooth function $x(t)$ by interpolating from $x(kT^\star)$, $k = 0, 1, 2, \ldots$ to a given x_d when

$$|\, x_d - x(kT^\star)\,| \gg |\, \Delta x_T\,|, \tag{8}$$

and ensure, that for a given $\dot{x}_d$ the following condition holds for $n > k$:

$$|\, \dot{x}_d - \dot{x}(nT^\star)\,| < \varepsilon, \tag{9}$$

where $\varepsilon > 0$ is a small constant, as well as

$$|\ddot{x}(t)| \leq \ddot{x}_{max}, \tag{10}$$

where $\ddot{x}_{max} > 0$ is a given constant.

The recomputation of the coefficients a_i terms in the proximity of the given x_d:

$$|\, x_d - x(kT^\star)\,| \leq |\, \Delta x_s\,|, \tag{11}$$

where $\Delta x_s = \Delta x_s(\dot{x}_d,\, \ddot{x}_{max})$ and $|\, \Delta x_s\,| > |\, \Delta x_T\,|$.

A name *virtual running point* can be used to indicate that each k-th interpolation step is performed to a virtual point situated at a distance Δx_T and this point is shifted with a sampling period $T^\star$ in the direction of x_d, as it is illustrated by Fig. 5.

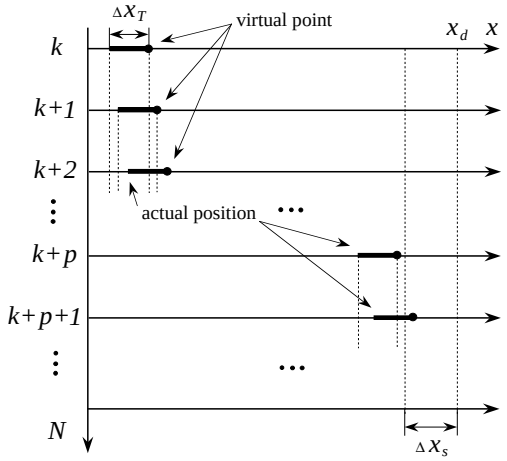

Figure 5: A virtual running point

First derivative. When the coefficients a_i are recomputed with the sampling period $T^\star$, one can derive from (6) and (7):

$$\dot{x}(kT^\star) = b^k\, \dot{x}_0 + c\,(1 + \sum_{i=1}^{k-1} b^i), \tag{12}$$

where

$$b = 1 - 4\,\tau + 3\,\tau^2, \tag{13}$$

$$c = 6\,(1-\tau)\,\tau\,\frac{\Delta x_T}{T} - 3\,(2-\tau)\,\tau\,\dot{x}_T \tag{14}$$

and $\tau = \frac{T^\star}{T}$. Because $0 < b < 1$, according to [14]:

$$\lim_{k\to\infty} \dot{x}(kT^\star) = \frac{c}{1-b}. \tag{15}$$

Let us denote

$$v = \frac{\Delta x_T}{T}. \tag{16}$$

Then, according to (13), (14) and (15), the following equation is obtained:

$$\lim_{k\to\infty} \dot{x}(kT^\star) = \frac{6\,(1-\tau)\,v - 3\,(2-\tau)\,\dot{x}_T}{4 - 3\,\tau}. \tag{17}$$

Because $\dot{x}_T$ is an auxiliary constant in such computations, one can choose $\dot{x}_T = 0$ and rewrite (17) as

$$\lim_{k\to\infty} \dot{x}(kT^\star) = \frac{6\,(1-\tau)}{4 - 3\,\tau}\, v. \tag{18}$$

Hence, if v is computed as

$$v = \frac{4 - 3\,\tau}{6\,(1\,-\,\tau)}\, \dot{x}_d, \tag{19}$$

the condition (9) is satisfied according to (18).

Second derivative. For an instant $t = kT^\star$, $\dot{x}_T = 0$ and recomputation of the coefficients a_i with the sampling period $T^\star$, one can obtain from (6), (7), (16) and (19):

$$\ddot{x}(kT^\star) = \frac{(4 - 3\,\tau)\,\dot{x}_d}{6\,(1-\tau)\,\Delta x_T} \cdot \left(\frac{4 - 3\,\tau}{1-\tau}\,\dot{x}_d - 4\,\dot{x}(kT^\star)\right). \tag{20}$$

From the equation (20) it follows, that a value of $\ddot{x}(t)$ depends on a deviation between $\dot{x}_d$ and $\dot{x}(t)$. Taking into account (9), one can conclude that this deviation has its extremal value at the moment when a new $\dot{x}_d$ is given, i.e. in the beginning of the interpolation step. According to (20), for given $\dot{x}_d$ and $\dot{x}(kT^\star)$, there can always be found such $\Delta x_T \neq 0$ that

$$|\ddot{x}(kT^\star)| \leq \ddot{x}_{max}, \tag{21}$$

where $\ddot{x}_{max} > 0$ is a given constant.

Recurrent equations. When the coefficients a_i are recomputed with a sampling period $T^\star$, the equation (6) can be rewritten as

$$x(t) = \sum_{i=0}^{3} a_{i,k} \left(\frac{t - kT^\star}{T}\right)^i, \tag{22}$$

where $t \in [\,kT^\star,\ (k+1)T^\star\,]$, $k = 0, 1, 2, \ldots$ and the coefficients are:

$$\begin{cases} a_{0,k} = x(kT^\star), \\ a_{1,k} = \dot{x}(kT^\star)\,T, \\ a_{2,k} = 3\,\Delta x_T - 3\,\dot{x}_T\,T - 2\,\dot{x}(kT^\star)\,T, \\ a_{3,k} = -2\,\Delta x_T + \dot{x}_T\,T + \dot{x}(kT^\star)\,T. \end{cases} \tag{23}$$

The features of the developed motion generation are:

- $x(t)$ and $\dot{x}(t)$ are continuous,
- $\dot{x}(t)$ tends to a given value $\dot{x}_d$ and the speed of the convergence depends on the value of Δx_T,
- $\ddot{x}(t)$ is limited and tends to zero while $\dot{x}(t)$ tends to $\dot{x}_d$.

These features allow one to apply the method to various robotic tasks where trajectory generation in real time is needed to control the motion of the mobile robots in a dynamic environment (e.g. collision avoidance or tracking an object).

Simulation example. Let us compare the motion generation based on (6), (7) with the proposed approach based on (22), (23). An example of the motion generation based on (6), (7) is shown in Fig. 6, where the boundary conditions are: $x_0 = \dot{x}_0 = \dot{x}_T = 0$ and $x_T = 5\ m$. The maximal velocity is set to $0.15\ m/s$.

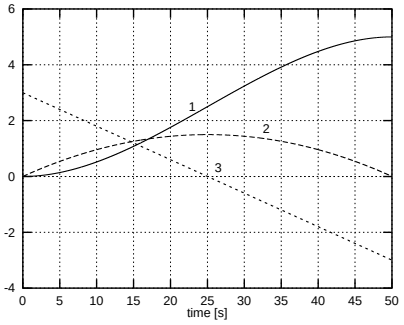

Figure 6: An example of motion generation without recomputation of the coefficients, where 1 - $x(t)$ in m, 2 - $\dot{x}(t)$ in 10 m/s and 3 - $\ddot{x}(t)$ in 0.1 m/s^2

The proposed approach to the motion generation based on (22), (23) and recomputation of the spline coefficients with a sampling period $T^\star$ is illustrated by Fig. 7 with the same boundary conditions. The desired value of the velocity was $\dot{x}_d = 0.15\ m/s$ and $\Delta x_T = 0.15\ m$. As it is seen from Fig. 7, the proposed method ensures a near trapezoidal profile of $\dot{x}(t)$ that allows the total time to be shortened in comparison with the case of Fig. 6.

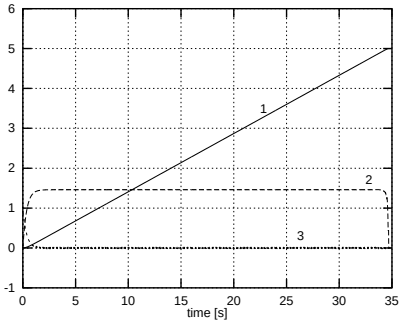

Figure 7: An example of motion generation with recomputation of the coefficients, where 1 - $x(t)$ in m, 2 - $\dot{x}(t)$ in 10 m/s and 3 - $\ddot{x}(t)$ in m/s^2

5 Experiments

The developed control system is implemented and tested on the omnidirectional mobile robot shown in Fig. 1. The software is developed in C language on a SUN-workstation. A VxWorks real-time operation system is used. The compiled code is transmitted via a wireless Ethernet to the on-board control system based on a Pentium processor and ISA bus.

The operation of the control system is illustrated by an example shown in Fig. 8 where the initial coordinates of the vehicle are (0.7, 1.1). In this example, the desired velocity is $0.15\ m/s$, and the default rotational velocity is $0.28\ rad/s$. Firstly, the robot moves in the direction of an intermediate position (0.2, 2.7). Then, after a period of 5 s used in this example, a new goal position $(-1.6, 2.2)$ is given so that the robot has to change its motion direction. The corresponding motion of the robot is depicted in Fig. 8 and the velocity profiles are shown in Fig. 9.

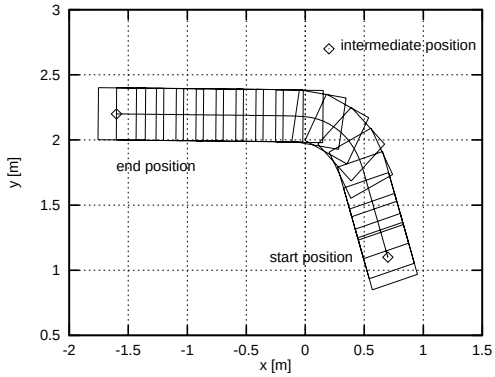

Figure 8: Motion on the (x, y)-plane

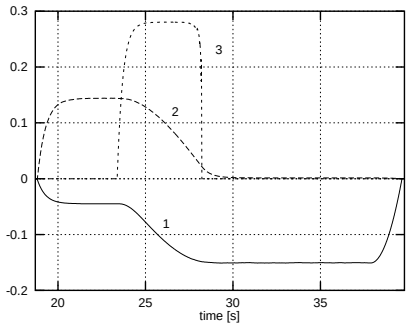

Figure 9: Velocity profiles: 1 - $v_x(t)$ in m/s, 2 - $v_y(t)$ in m/s and 3 - $v_\theta(t)$ in rad/s

The control system was also tested on two visually-coupled mobile robots. Our experimental setup is

shown in Fig. 10. One robot is assigned as a leader and the other as a follower. A remote human operator uses a graphical user interface and sets goal positions and motion velocity of the leader within an environmental map shown on the display. During the motion, the leader's position is depicted on the map and the operator views the visual data gathered by the on-board CCD camera of the robot about the local environment. The follower is visually-coupled to the leader and moves as a visually-connected trailer. Our video illustrates the experiments performed.

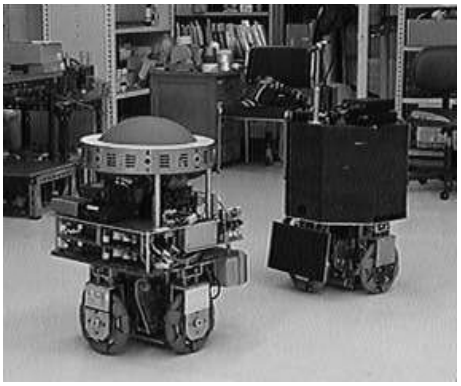

Figure 10: Visually-coupled mobile robots: a leader (left) and a follower (right)

6 Conclusion

The motion generation approach and its implementation in the control system of the omnidirectional mobile robot were considered. The experimental results obtained for the remote control and visually-coupled motion of the two mobile robots have shown effective operation of the control system. The mobile robot will be used in a distributed autonomous robotic system to perform inspection tasks (e.g. to examine equipment of an electric power station) or delivery operations (e.g. in shops or hospitals). The system can also be used for entertainment purposes (e.g. in a robotic soccer competition RoboCup).

Acknowledgements

This research work was partially supported by the STA Fellowship Program (1997-99). Thanks are given to I. Endo, T. Fujii, T. Ishikawa, H. Kaetsu, K. Kawabata, D. Kurabayashi and T. Suzuki for the creative atmosphere during our work.

References

[1] R. G. Simmons, Structured Control for Autonomous Robots, *IEEE Trans. on Robotics and Automation*, Vol. 10, No. 1, 1994, pp. 34-43.

[2] R. A. Brooks, A Robust Layered Control System for a Mobile Robot, *IEEE Journal of Robotics and Automation*, 1986.

[3] R. Chatila, Deliberation and Reactivity in Autonomous Mobile Robots, *Robotics and Autonomous Systems*, Vol. 16, No. 2-4, 1995, pp. 197-211.

[4] K. de Boor, *A Practical Guide to Splines*, Springer Verlag, New York, USA, 1978.

[5] H. Akima, A New Method of Interpolation and Smooth Curve Fitting Based on Local Procedures, *Journal of Assc. for Comp. Mach.,* Vol. 17, No. 4, 1970, pp. 589-602.

[6] N. F. Stewart, On-Line Robotic Trajectory Control Based on Spline Interpolation, *INFOR*, Vol. 23, No. 2, 1985, pp. 159-170.

[7] Y. Bestaoui, On Line Motion Generation with Velocity and Acceleration Constraints, *Robotics and Autonomous Systems*, Vol. 5, No. 3, 1989, pp. 279-288.

[8] H. Asama *et al.*, Development of an Omni-Directional Mobile Robot with 3 DOF Decoupling Drive Mechanism, *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Nagoya, Japan, May 21-27, 1995, pp. 1925-1930.

[9] T. Suzuki *et al.*, A Multi-Robot Teleoperation System Utilizing the Internet, *Advanced Robotics*, Vol. 11, No. 8, 1998, pp. 781-797.

[10] Y. Arai *et al.*, Collision Avoidance among Multiple Autonomous Mobile Robots using LOCISS (LOcally Communicable Infrared Sensory System), *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Minnesota, MN, USA, April 22-28, 1996, pp. 2091-2096.

[11] Y. Arai *et al.*, Adaptive Behaviour Acquisition of Collision Avoidance among Multiple Autonomous Mobile Robots, *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Grenoble, France, Sept. 7-11, 1997, pp. 1762-1767.

[12] I. E. Paromtchik, U. M. Nassal, Reactive Motion Control for an Omnidirectional Mobile Robot, *Proc. of the Third European Control Conference*, Roma, Italy, Sept. 5-8, 1995, pp. 3074-3079.

[13] I. E. Paromtchik, U. Rembold, A Practical Approach to Motion Generation and Control of an Omnidirectional Vehicle, *Proc. of the IEEE Int. Conf. on Robotics and Automation*, San Diego, CA, USA, May 8-13, 1994, pp. 2790-2795.

[14] I. N. Bronshtein, K. A. Semendyayev, *Handbook of Mathematics*, Verlag Harri Deutsch, Van Nostrand Reinhold Co., Edition Leipzig, 1985, 973 p. (see p. 102).