The IPSI BgD Transactions on Internet Research

Multi-, Inter-, and Trans-disciplinary Issues in Computer Science and Engineering

A publication of

IPSI Bgd Internet Research Society
New York, Frankfurt, Tokyo, Belgrade
July 2012 Volume 8 Number 2 (ISSN 1820-4503)

Table of Contents:

The IPSI BgD Internet Research Society

The Internet Research Society is an association of people with professional interest in the field of the Internet. All members will receive this TRANSACTIONS upon payment of the annual Society membership fee of €100 plus an annual subscription fee of €1000 (air mail printed matters delivery).

Member copies of Transactions are for personal use only

IPSI BGD TRANSACTIONS ON ADVANCED RESEARCH

www.internetjournals.net

Navigation, Motion Planning and Control of Autonomous Wheeled Mobile Robots in Labyrinth Type Scenarios

Rodić Aleksandar

Abstract—*The paper regards navigation, motion planning and control as well obstacle avoidance in unknown and unpredicted environments including collision avoidance of mobile robots in evolving surroundings. The problem relates to searching the techniques how to navigate toward a goal in an unknown, confined or cluttered environment when the obstacles to avoid are discovered in real time. The tools developed to address this issue thus combine motion planning and control theory techniques including a non-linear model-based approach. The advantage of such reactive obstacle avoidance technique is to compute motion by introducing the sensor information within the control loop, used to adapt the motion to any contingency incompatible with initial plans. Being the global reasoning is required, a trap situation could occur but despite to this limitation, obstacle avoidance techniques are mandatory to deal with mobility problems in unknown and evolving surroundings as presented in the paper.*

Key words—Fuzzy reasoning, mobile wheeled robots, motion planning and control, obstacle avoidance

1. Introduction

Control of wheeled mobile robots is the subject of numerous research studies as reported in [1]. In particular, non-holonomy constraints associated with these systems have motivated the development of highly nonlinear control techniques. For the sake of simplicity, the control methods are developed mainly for unicycle-type and car-like mobile robots. Most of the results can in fact be ex tended/adapted to other type mobile robots (e.g. holonomic), in particular to systems with trailers. A complementary problem to control motion of mobile robots concerns with global motion planning and obstacle (mobile and/or immobile) avoidance in variety of different static as well as dynamic scenarios. The problem considers sensor-based motion to face the physical issues of a r eal system navigating in a real world. The problem relates to searching the techniques how to navigate toward a goal in a confined, troublesome or cluttered environment when the obstacles to avoid are discovered in real time? This is the question that addresses simultaneous motion planning and control as well obstacle avoidance.

The motion planning problem for a non-holonomic system can be stated as follows: given a map of the environment with obstacles in the workspace, a robot subject to non-holonomic constraints, an initial configuration and a goal configuration, and an adm issible collision-free path between the initial and goal configurations. Solving this problem requires to take into account both the configuration space constraints due t o obstacles and t he non-holonomic constraints of robotic system. The tools developed to address this issue thus combine motion planning and control theory techniques.

The objective of motion planning techniques is to compute a collision-free trajectory to the target configuration that complies with the vehicle constraints. That assumes a perfect model of the robot and scenario. The advantage of these techniques is that they provide complete and global solutions of the problem. Nevertheless, when the surroundings are unknown and unpredictable, these techniques fail. A complementary way to address the motion problem is obstacle avoidance. The objective is to move a vehicle towards a target location free of collisions with the obstacles detected by the sensors during motion execution. The advantage of reactive obstacle avoidance is to compute motion by introducing the sensor information within the control loop, used to adapt the motion to any contingency incompatible with initial plans. The main cost of considering the reality of the world during execution is locality. In this instance, if global reasoning is required, a trap situation could occur. Despite this limitation, obstacle avoidance techniques are mandatory to deal with mobility problems in unknown and ev olving surroundings.

The methods that combine both the global point of view of motion planning and t he local point of view of obstacle avoidance have been already developed. How to consider robot perception at the planning level? This is the so-called sensor-based motion planning. Several variants exist, such as the BUG algorithms

Manuscript received on December 27th, 2010. This work was supported by Serbian Ministry of Science and in part by the program of the Hungarian National Development Agency TÁMOP-4.2.2/08/1/2008-0008.

Aleksandar Rodić. Author is with the Robotics laboratory, Mihajlo Pupin Institute, University of Belgrade, Serbia (e-mail: aleksandar.rodic@pupin.rs).

initially introduced in [2]. However none of them consider the practical context of non-holonomic mobile robots.

Taxonomy of obstacle avoidance techniques and some representative methods can be described here. First there are two groups according [1]: methods that compute the motion in one step and that do it in more than one. One-step methods directly reduce the sensor information to a motion control. There are two types:

- o The heuristic methods were the first techniques used to generate motion based on sensors. The majority of these works derived from classic planning methods [2]-[5].
- o The methods of physical analogies assimilate the obstacle avoidance to a known physical problem. The representative of them is the potential field methods [6,7]. Other works are variants adapted to uncertain models [8] or that use other analogies [9]-[11].

Methods with more than one step compute some intermediate information, which is processed next to obtain the motion.

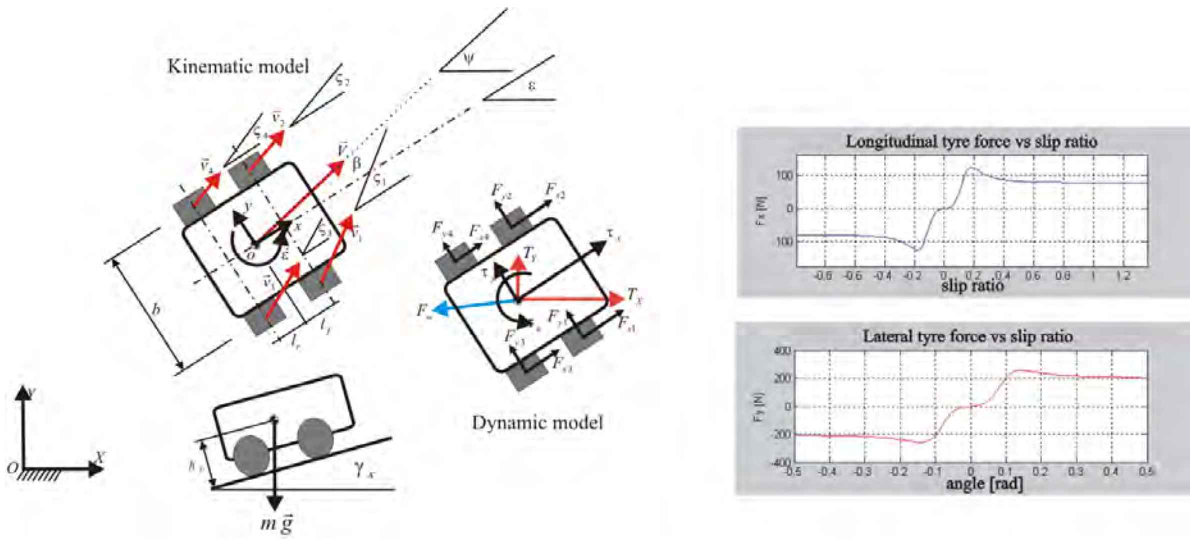
- o The methods of subset of controls compute an intermediate set of motion controls, and next choose one of them as a solution. There are two types: (i) methods that compute a subset of motion directions. The vector field histogram [12] and the obstacle restriction method [13] can be mentioned as representatives. Another method is presented in [14]. (ii) Methods that compute a subset of velocity controls. The dynamic window approach [15] and the velocity obstacles [16] can be taken as examples. Another method based on similar principles but developed independently is the curvature velocity method [17].
- o Finally, there are methods that compute some high-level information as intermediate information, which is translated next in motion. The nearness diagram navigation [18,19] is the representative of this method.

To summarize the previous consideration, the usage of an obstacle avoidance technique with a vehicle in a given scenario is highly dependent on the scenario nature (static or dynamic, unknown or known, structured or not, or its size for example). Usually, this problem is associated with the integration of motion planning and obstacle avoidance. All the methods outlined briefly in this section have advantages and disadvantages depending on the navigation context, like uncertain worlds, motion at high speeds, motion in confined or troublesome spaces, etc. Unfortunately, there is no metric available to measure the performance of the methods quantitatively. In that sense, the considerations to be conducted in the paper will demonstrate a methodology of motion control of mobile robots that combines adhoc motion planning and obstacle avoidance together with model-based, non-linear motion control.

2. *MODELING OF WHEELED ROBOTS*

For the purpose of motion control development and simulation, a non-holonomic wheeled mobile robot with differential (skid) steering is considered (Fig. 1). This kind of steering is able to move more directly (than car-like, i.e. Ackerman steering [20]) from one point to different point. However it would only be able to execute a single surge displacement (forward/backward) and a single rotation (turn/spin). Thus it can execute only two controlled DOFs which is one fewer than are available in its task space what is characteristic for non-holonomic systems.

According to the previous consideration a 4WD mobile robot platform with differential steering is assumed in the paper as system representative. A 3D model of mobile robot is considered taking into account that robot can move on the sloped



a) b)

Figure 1. Model of non-holonomic wheel-based robot considered in the paper: a) Kinematical and dynamic

surface, too. In the general case, surface inclination angle can appear in both, longitudinal γ_x as well as lateral γ_y direction of motion (Fig. 1a). Direction of forward (transport) speed vector $\vec{V}$ (Fig. 1a) depends on tyre angular velocities, robot parameters as well as tyre-ground interaction parameters and ground surface conditions. Robot motion is considered in the inertial $OXYZ$ coordinate system attached to the ground surface. The local coordinate system $oxyz$ is attached to the center of mass of the mobile robotic platform and it is mobile, too. Motion of robot platform is consequence of the independendent differential driving (rotation) of robot wheels. Corresponding longitudinal $F_{xi}, i = 1,4$ and lateral $F_{yi}, i = 1,4$ tyre forces produce desired robot motion and desired forward speed. Forward speed $\vec{V}$ in general case is not colinear with direction of longitudinal robot axis of symetry. The angle between velocity vector $\vec{V}$ and longitudinal axis of symetry *x* is defined by angle β known as a slip angle [20]. Particular wheels perform corresponding rotational as well tranlsational movements. Linear, i.e. translational tyre velocities presented in Fig. 1a are signed as $\vec{v}_i, i = 1,4$. These linear tyre velocities do not coincide with corresponding direction of motion in general case. The conseqence is appearance of the tyre slip angles $\varsigma_i, i = 1,4$ as presented in Fig. 1a. Some important geometry parameters of rover are presented in Fig. 1a: b - track of the rover, l_r - rear wheels axis distance from the center of mass, l_f - front wheels axis distance from center of mass, and h_C - height of the center of mass with respect to the ground surface. Vector of state variables that define robot position in the inertial coordinate system $OXYZ$ is in general case of 6×1 order [20]. The planar vehicle dynamics (vehicle dynamics in the plane of motion) is considered in the paper although the terrain surface inclination will be taken into account. Assuming the previous simplification, the robot state variables including three coordinates can be defined in the vector form:

$$
\begin{aligned}
q &= [X \ Y \ \varepsilon]^T \\
\dot{q} &= [\dot{X} \ \dot{Y} \ \dot{\varepsilon}]^T
\end{aligned}
\tag{1}
$$

where *X* and *Y* are translatory displacements in corresponding coordinate diractions and ε is yaw angle about the vertical axis passes through center of mass. The dynamical model of the 4WD rover presented in Fig. 1a, defined by corresponding geometry parameters, kinematical variables and vector of state variables (1) can be defined as:

$$
T = H(q) \cdot \ddot{q} + h_{ccg}(q, \dot{q}) - F_w \tag{2}
$$

where $T \in \Re^{3\times1}$ is the vector of generalized (traction/braking) forces and torques that act in the robot center of mass and has three commponents in the main coordinate directions (Fig. 1a): two generalized forces T_X and T_Y (T_Z is not considered in (2) bearing in mind that *Z* is not a state variable (1)) and yaw torque T_ε; $H \in \Re^{3\times3}$ is the inertia matrix of the rover; $h_{ccg} \in \Re^{3\times1}$ is vector of centrifugal, Coriolis and gravity forces acting upon the system and $F_w \in \Re^{3\times1}$ is vector of external resestance forces and torques including aerodynamic resistance, rolling ressistence, Coulomb friction, etc. Vector of driving/traction forces and troques acting in the vehicle center of mass, expressed in the local coordinate system oxy z (Fig. 1a) attached to the center of mass (MC) can be defined in the form:

$$
\tau = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_\varepsilon \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{4} F_{xi} \\ \sum_{i=1}^{4} F_{yi} \\ M_Z \end{bmatrix} \tag{3}
$$

$$
\begin{aligned}
M_Z = {} & (F_{x1} + F_{x3} - F_{x2} - F_{x4}) \cdot \frac{b}{2} + \\
& (F_{y1} + F_{y2}) \cdot l_f - (F_{y3} + F_{y4}) \cdot l_r
\end{aligned}
\tag{4}
$$

Traction/braking forces at the vehicle tyres are calculated by use of the non-linear Pacejka tyre model known as „*magic formula*“ tyre model [21]. The longitudinal and lateral commponents $F_{xi}, F_{yi}, i = 1,4$ depends on two kinematical variables: tyre slip ratio s_i and tyre slip angle α_i. Both variables can be calculated according to the [20,21] from the following relations:

$$
s_i = \frac{v_i \cos(\alpha_i) - r_i \, \omega_i}{r_i \, \omega_i} \tag{5}
$$

$$
\alpha_i = \delta_i - \varsigma_i \tag{6}
$$

$$
\begin{aligned}
v_1 &= \sqrt{(\dot{y} + l_f \dot{\varepsilon})^2 + (\dot{x} - b/2\,\dot{\varepsilon})^2} \\
v_2 &= \sqrt{(\dot{y} + l_f \dot{\varepsilon})^2 + (\dot{x} + b/2\,\dot{\varepsilon})^2} \\
v_3 &= \sqrt{(\dot{y} - l_r \dot{\varepsilon})^2 + (\dot{x} - b/2\,\dot{\varepsilon})^2} \\
v_4 &= \sqrt{(\dot{y} - l_r \dot{\varepsilon})^2 + (\dot{x} + b/2\,\dot{\varepsilon})^2}
\end{aligned}
\tag{7}
$$

$$
\begin{aligned}
tg(\varsigma_1) &= \frac{\dot{y} + l_f \dot{\varepsilon}}{\dot{x} - b/2\,\dot{\varepsilon}}, \quad tg(\varsigma_2) = \frac{\dot{y} + l_f \dot{\varepsilon}}{\dot{x} + b/2\,\dot{\varepsilon}} \\
tg(\varsigma_3) &= \frac{\dot{y} - l_r \dot{\varepsilon}}{\dot{x} - b/2\,\dot{\varepsilon}}, \quad tg(\varsigma_4) = \frac{\dot{y} - l_r \dot{\varepsilon}}{\dot{x} + b/2\,\dot{\varepsilon}}
\end{aligned}
\tag{8}
$$

For every particular tyre $i = 1,4$, the angle δ_i represents corresponding steering angle (in the case when the rover has such capability to change orientation angles of tyres), v_i is corresponding translatory speed of centre of mass of the i-th particular tyre, and ς_i represents tyre speed angle with respect to the longitudinal direction of motion of robot platform *x*. Angles of tyre velocities are calculated from the following relations (8) according to [20].

Bearing in mind the non-linear character of tyre model (Fig. 1b), the tyre forces are non-linear functions of their arguments as explained in [20,21] and have the forms:

$$F_{xi} = f_1(s_i, \alpha_i), \quad F_{yi} = f_2(s_i, \alpha_i) \tag{9}$$

The dependency between the generalized forces and torques expressed in the absolute coordinate system and the local system attached to the MC of the rover (Fig. 1a) can be ex pressed in the following way:

$$\begin{aligned} T_X &= \tau_x \cos(\varepsilon) - \tau_y \sin(\varepsilon), \\ T_Y &= \tau_x \sin(\varepsilon) + \tau_y \cos(\varepsilon), \\ T_\varepsilon &= \tau_\varepsilon \end{aligned} \tag{10}$$

According to [20] and assuming that mobile robot in this case is considered as planar mechanism, corresponding matrix and vectors given in the model (2) can be assumed in the form:

$$H = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_Z \end{bmatrix} \tag{11}$$

$$h_{ccg} = \begin{bmatrix} -m\,\dot{y}\,\dot{\varepsilon} + m\,g\sin(\gamma_x) \\ m\,\dot{x}\,\dot{\varepsilon} + m\,g\sin(\gamma_y) \\ 0 \end{bmatrix} \tag{12}$$

$$F_w = F_a + F_{tr} \tag{13}$$

where *m* is robot mass, *Iz* is robot's axial moment of inertia with respect to the axis *z*, *g* is the gravity acceleration. Resultant vector of aerodynamic resistance as well as rolling resistance forces and torques is determined according [20] and has the form:

$$F_w = \begin{bmatrix} -K_x\dot{x}^2 - \sum_{i=1}^{4} f_{r_i} F_{z_i} \cos\zeta_i \\ -K_y\dot{y}^2 - \sum_{i=1}^{4} f_{r_i} F_{z_i} \sin\zeta_i \\ -K_\varepsilon\dot{\varepsilon} \end{bmatrix} \tag{14}$$

where K_x, K_y represents corresponding air resistence coefficients obtained experimentaly for the particular robot and K_ε is a yaw-rate damping coefficient depending on t yre-ground conditions; f_{r_i} is a r olling resistance coefficient of the i-th tyre, and F_{z_i} represents corresponding i-th tyre payload.

Model determined by relations (2)-(14) is used in this paper for synthesis of the model-based (dynamic) control of wheeled mobile robot.

3. NAVIGATION, PATH PLANNING AND OBSTACLE AVOIDANCE

A control system structure of mobile robots is proposed in the paper. It represents a hiearachy system consists of two functional levels: (i) high-level, and (ii) low-level. High level module is designed to enable cognitive tasks performing such as: navigation in presence of obstacles, path planning and c olision avoidance with surrounding obstacles. At the low-level, distribution of control efforts per particular tyres is provided. Control effort is desired to be uniformly distributed among the robot wheels. For the sake of simplicity, the wheels from the same side of mobile robot have equal angular velocities of rotation. Non-holonomic wheeled robots represent overactuated systems.

When an autonomous mobile robot moves

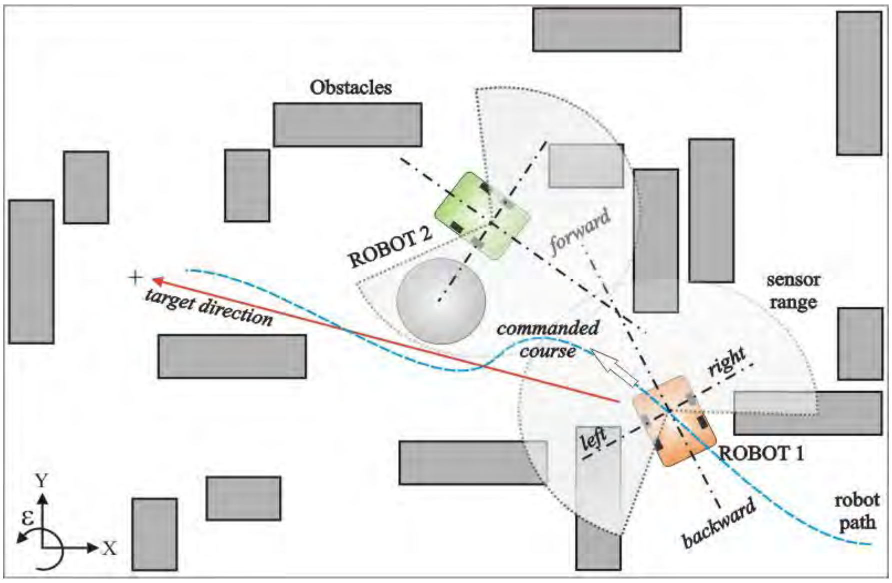


Figure 2. Model of navigation and motion planning in presence of mobile and immobile obstacles

towards some target position and its sensors detect obstacle(s), corresponding avoiding strategy should be activated. In that sense, robot motion can be described as compromise between avoiding obstacles and moving towards the target position as presented in Fig. 2. Autonomous robots react to both of the sensed variables (target direction and collision free direction of motion with respect to obstacles) to manage its motion. Corresponding sensed variables used for navigation and motion planning in presence of obstacles in unknown environment are (Fig. 2): (i) distances to surrounding obstacles within the corresponding sensor range (front-rear and right-left side obstacles), (ii) azimuth angle with respect to the obstacle (angle measured with respect to the robot longitudinal axis of symmetry), and (iii) relative position of the robot with respect to the target point. Moving towards the target point (Fig. 2) and avoiding obstacles in its surrounding, a mobile robot changes its orientation in the task space and its forward velocity. When the obstacle is detected by sensors, mobile robot slows down and changes its direction of motion according to the actual conditions of motion. Navigation strategy of mobile robot is determined in such a way to enable guidance of the robot in presence of obstacles (mobile and immobile) and tracking target direction. For this purpose robot uses acquired sensor information about the target point position. For the case of spatial reasoning in unknown and unpredicted environments, appropriate fuzzy inference system (FIS) is commonly used [2]-[19], [22] to perform such kind of robot tasks. In this paper, two FIS are designed (FIS-1 and FIS-2, Figs. 3 and 4) : one for obstacle avoidance and other for collision avoidance of mobile obstacles or other robots.

The novelty presented in this paper regards with utility of the robot model (2)-(14) and corresponding complementary FISs presented in Figs. 3 and 4. Model-based control and cognitive knowledge-based algorithms are combined within the robot controller to enable accurate navigation and reliable motion control in presence of obstacles and in unknown and unpredicted environments. Corresponding model-based (dynamic) control is designed in the paper to improve corresponding dynamic performances of robotic system during its motion. Assumed model-based approach to control of system motion provides extended universality of the proposed control technique considered in this paper. Specifity of any fuzzy inference system is that it is valid for one particular system (or quite similar one) but not for systems that are significantly different. The reasons lay in fact that membership functions and fuzzy rules, characteristic for one particular FIS, are designed to fit behaviour (model) of this specific system The FIS designed in such a way is not of general purpose to fit other particular systems, too. The way to overcome this drawback of the FIS specifity lay in combination of two techniques - fuzzy logic reasoning and model-based control. The consequence of such approach is that fuzzy inference system (i.e. FIS block) provides two referent (commanded) variables: (i) referent forward speed V_0, and (ii) referent yaw-rate $\dot{\varepsilon}_0$. Moving through the unknown environment in presence of mobile and immobile obstacles, robot has to determine the strategy how to avoid these obstacles and navigate towards the target location. Because of that, two FIS systems (Figs. 3 and 4) are needed to be engaged to determine referent variables V_0 and $\dot{\varepsilon}_0$. In the case when the robot meets mobile obstacles, then the FIS-2 block should be favorized (dominant) to calculate referent variables V_0 =*CS-2* and $\dot{\varepsilon}_0$=*CYR-2* (Fig. 3). In other case (meeting of immobile obstacles), the following rules shoulod be assumed V_0 =*CS-1* and $\dot{\varepsilon}_0$=*CYR-1* (Fig. 4). Modified robot velocity V_0 and referent yaw-rate $\dot{\varepsilon}_0$, calculated from FIS cognitive blocks FIS-1 or FIS-2 (Figs. 3 and 4) represent new-calculated referent values that are used as the inputs q_0 and $\dot{q}_0$ to the low-level control that represents model-based control algorithm according to (1). In the case of obstacle avoidance, reference direction, front, right, left and rear distnces (proximities) to obstacles and motion modus (forward or backward) determine referent variables (Fig. 3). In other case (collision avoidance), distance to collision point of meeting robots, intersection angle, speeds of colising robots, traffic rules (priority) as well as right, left and rear distances to obstacle are relevant for

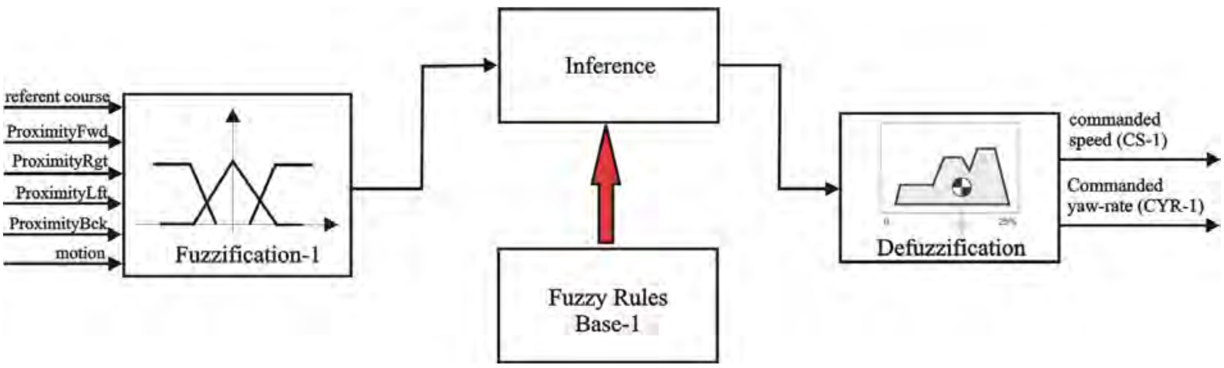


Figure 3. Block scheme of the Fuzzy Inference System FIS-1 developed for obstacle avoidance in unknown environment

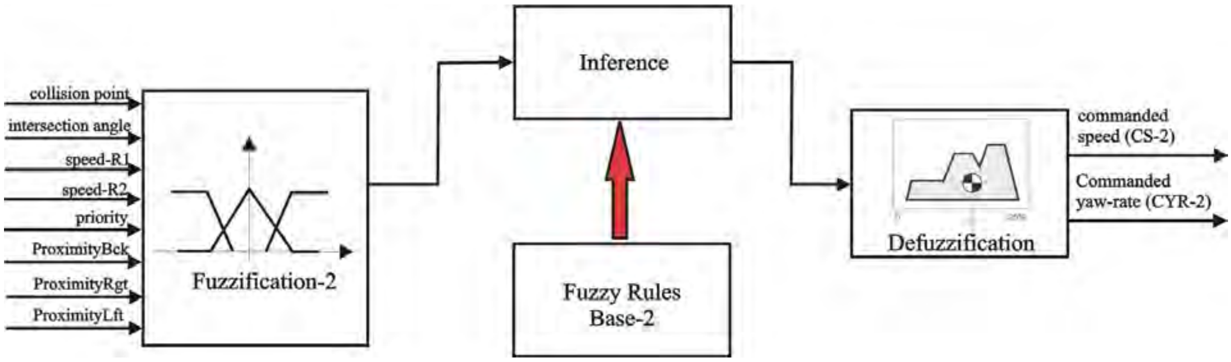


Figure 4. Block scheme of the Fuzzy Inference System FIS-2 developed for collision avoidance in evolving environment

calculation of referent variables V_0 and $\dot{\varepsilon}_0$. Fuzzy reactive navigation strategy of collision-free motion and velocity control of mobile wheeled robots in an unknown environment with obstacles is proposed in the paper. The intelligent robot reactive behavior is formulated by appropriately designed membership functions and corresponding fuzzy rules. The procedure is well described in the open l iterature [2]-[19] and details will not be specially elaborated here. The input variables, presented in Figs. 3 and 4, are expressed by corresponding linguistic labels and appropriate Gaussian membership functions shaped on the basis of experience and simulation tests.

4. MOTION CONTROL

Control algorithm that has to provide accurate trajectory tracking and s atisfactory dynamic perfromances of robotic system is based on the robot model (2). Then, the control algorithm can be written in the following form as used with autonomous automotive systems [20]:

$$\begin{aligned} T &= H(q)\cdot\hat{\ddot{q}} + h_{ccg} - F_w, \\ \hat{\ddot{q}} &= \ddot{q}_0 - K_d\left(\dot{q}-\dot{q}_0\right) - K_p\left(q-q_0\right) \end{aligned} \tag{15}$$

where K_p and K_d are corresponding proportional and differential control gains. Vectors q_0, $\dot{q}_0$ and $\ddot{q}_0$ are calculated on the basis of the referent values V_0 and ε_0 obtained from the higher control level, i.e. from the cognitive block. They provide the following variables regarding speed of motion in three coordinate direction / longitudinal, lateral and yaw one:

$$\begin{aligned} \dot{x}_0 &= V_0\cos(\varepsilon_0) \\ \dot{y}_0 &= V_0\sin(\varepsilon_0) \\ \dot{\varepsilon}_0 &= \dot{\varepsilon}_0 \end{aligned} \tag{16}$$

where ε_0 are determined from :

$$\varepsilon_0 = \int \dot{\varepsilon}_0 \cdot dt \tag{17}$$

Now, the nominal speed vector $\dot{q}_0$ can be expressed in the form:

$$\dot{q}_0 = \left[\dot{x}_0 \;\; \dot{y}_0 \;\; \dot{\varepsilon}_0\right]^T \tag{18}$$

And corresponding nominal acceleration is obrained from the relation:

$$\ddot{q}_0 = \frac{d\,\dot{q}_0}{dt} \tag{19}$$

From (15) the control (traction/braking) forces T_X, T_Y and control (yaw) torque T_ε can be calculated with respect to the absolute coordinate system (Fig. 1). From (10) driving forces and torques in the longitudinal τ_x, lateral τ_y and yaw τ_ε direction are determined. Thsese values can be calculated from the relations:

$$\begin{aligned} \tau_x &= T_X\cos(\varepsilon) + T_Y\sin(\varepsilon), \\ \tau_y &= -T_X\sin(\varepsilon) + T_Y\cos(\varepsilon), \\ \tau_\varepsilon &= T_\varepsilon \end{aligned} \tag{20}$$

Control forces/torques τ_x, τ_y and τ_ε are produced by corresponding tyre forces distributed in an appropriate way among the four wheels. 4WD mobile robotic platform is "overactuated" since in general case there are four driving wheels and m otion is perfromed in three coordinate directions: *x*, *y* and ε. Theoreticaly, there are eight tyre force commponents $F_{xi}, i=1,4$ and $F_{yi}, i=1,4$ that should to enable desired motion of robot body. Practically, that requires the unknown forces $F_{xi}, i=1,4$ and $F_{yi}, i=1,4$ should be determined from the relations (3) and (4) taking into account pre-determined generalized forces/torques τ_x, τ_y and τ_ε. In order the calculation to be mathematicaly determined the certian assumptions must be assumed in order to simplified calculation of unknown control variables. The following simplifications can be assumed without loosing system maneouverability and fine dynamic performnaces:

$$\begin{aligned} &F_{x3} \equiv F_{x1},\, F_{x4} \equiv F_{x2}, \\ &F_{y2} = \kappa_2\cdot F_{y1},\, F_{y3} = \\ &\quad = \kappa_3\cdot F_{y1},\, F_{y4} = \kappa_4\cdot F_{y1} \end{aligned} \tag{21}$$

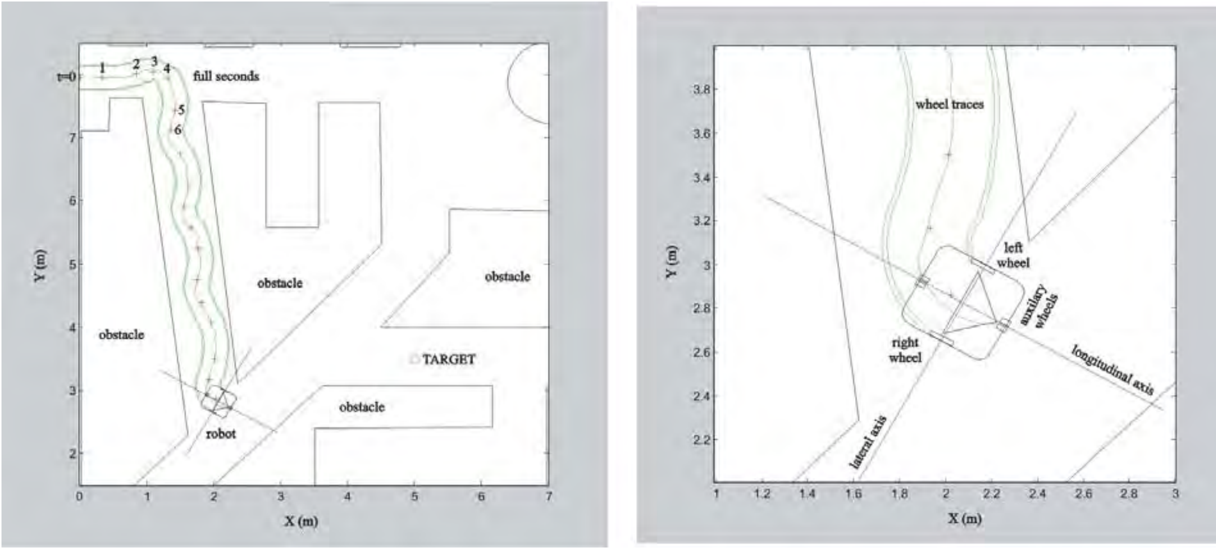


Figure 5. Experimental verification of navigation strategy and motion control algorithms of a wheeled robot in a labyrinth scenario model – macro plan (left) and micro plan (right)

where $\kappa_i = F_{yi} / F_{y1}, i = 1,3$ are corresponding coeficients regarding lateral force amplitudes with respect to the referent value characteristic for the tyre signed as no. 1. Taking into account the assumptions (21) and including them in (3) and (4), then three equations with corresponding three unknown variables F_{x1}, F_{x2} and F_{y1} can be determined:

$$\begin{aligned}
\tau_x &= 2 \cdot F_{x1} + 2 \cdot F_{x2}, \\
\tau_y &= (1 + \kappa_2 + \kappa_3 + \kappa_4) \cdot F_{y1}, \\
\tau_\varepsilon &= b \cdot F_{x1} - b \cdot F_{x2} + \\
&+ \left[(1 + \kappa_2) \cdot l_f - (\kappa_3 + \kappa_4) \cdot l_r \right] \cdot F_{y1}
\end{aligned} \tag{22}$$

From the system of equations (22) the unknown forces can be calculated to satisfy control requirements. Calculated tyre forces represents control variables but the true control variables are particular angular velocities of robot wheels $\omega_i, i = 1,4$. Since, according to the simplifications (21) the rotations of the right-hand side and left-hand side wheels are controled in pair, the wheel angular velocities ω_i are calculated from the kinematical relation (5) which defines the tyre slip ratio. Since the longitudinal $F_{xi}, i = 1,4$ and lateral tyre forces $F_{yi}, i = 1,4$ are non-linear functions (relations (9) and Fig. 1) of tyre slip ratios s_i and tyre slip angles α_i that implies the unknown variables are calculated by solving inverse mathematical problem of the model given by relations (9). In the last step of control signal determination, tyre angular velocities ω_i are calculated from (5) as the first-hand control variables. It is enough to determine corresponding angular velocitios for single, right and left particular wheels since the wheels at the same side rotate with the equal speeds of roation.

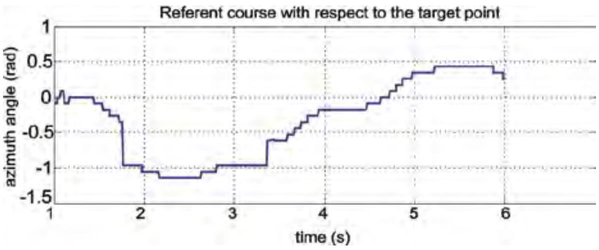


Figure 6. Referent course of motion of mobile robot with respect to the target direction measured from the longitudinal axis

5. VERIFICATION OF CONTROL APPROACH

To verify the proposed control approach, one characteristic labyrinth scenario with several wheeled robots moving around is simulated by use of the Virtual WRSP software toolbox [23]. One typical example of reactive navigation in presence of obstacles, searching for the collision-free corridors and tracking of the target direction is simulated (Fig. 5). A fragment of the first six seconds of motion is chosen (Fig. 5, left plot) to focus attention to the cornering maneuver. Corresponding sensor-based input signal to the FIS-1 (Fig. 3), that represents a referent course of the mobile robot, is presented in Fig. 6. The referent course is a dominant input signal to the control block. It directly influences guidance of the robot towards the assumed target point. Referent course of the wheeled robot as well as proximities of obstacles (Fig. 3) in surrounding determines the command variables: referent velocity V_0 and yaw-rate $\dot{\varepsilon}_0$. Bearing in mind that the wheeled robots considered in the paper use differential (skid) steering, the corresponding angular tire velocities ω_i represent actual control variables and not command variables generated by FIS block shown in Fig. 7. Actual control variables ω_i that correspond to the command variables shown in Fig. 7 are presented in Fig. 8. They are calculated indirectly from (5), starting from (22)

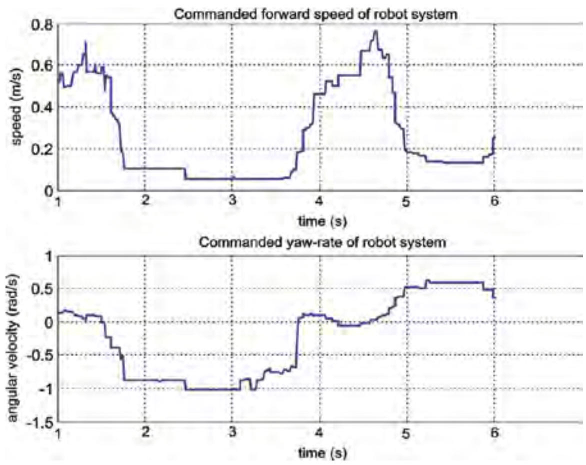


Figure 7. Command (referent) variables V_0 and $\dot{\varepsilon}_0$ obtained as output signals of the FIS-1 block

and tracking the procedure in the reverse direction. As shown in Fig. 8, different tire angular velocities of right and left side tires are generated by the robot controller. The consequence is a differential traction of robot tires due t o the different ground-tire interaction forces. Proposed robot controller is based on combination of a reactive fuzzy navigation and model-based motion control that ensures smooth motion towards target location as described in the paper.

REFERENCES

[1] Siciliano, B. and Khatib, O., Eds., Handbook of Robotics, Morin, P., Samson, C.,"Motion Control of Wheeled Mobile Robots", Minguez, J., Lamiraux, F., Laumond, J-P., "Motion Planning and O bstacle Avoidance", Springer, 2008, ISBN: 978-3-540-23957-4, pp. 800-850.

[2] Lumelsky, V., Stepanov, A., "Path planning strategies for a poi nt mobile automation moving admist unknown obstacles of arbitrary shape", Algorithmica, 2, 403–430 (1987)

[3] Chatila, R., "Path planning and environmental learning in a m obile robot system", Eur. Conf. Artif. In tell. (1982)

[4] Strategiesl, L., "Strategies for solving collision-free trajectories problems for mobile robots and manipulator robots", Int. J. Robot. Res., 3(4), 51–65 (1984)

[5] Chattergy, R., "Some heuristics for the navigation of a robot", Int. J. Robot. Res. 4(1), 59–66 (1985)

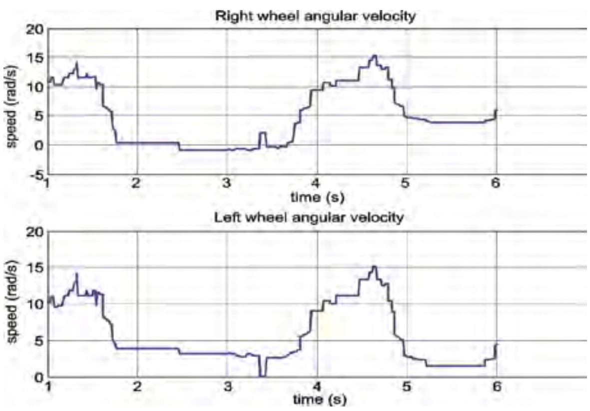


Figure 8. Skid steering – tyre velocities as control variables of wheeled mobile robot

[6] Khatib, O., "Real-time obstacle avoidance for manipulators and mobile robots", Int. J. Robot. Res. 5, 90–98 (1986)

[7] Krogh, B.H., Thorpe, C.E., "Integrated path planning and dynamic steering control for autonomous", IEEE Int. Conf. Robot. Autom. (1986), pp. 1664–1669

[8] Borenstein, J., Koren, Y., "Real-time obstacle avoidance for fast mobile robots", IEEE Trans. Syst. Man Cybern. 19(5), 1179–1187 (1989)

[9] Azarm, K., Schmidt, G., "Integrated mobile robot motion planning and execution in changing indoor environments", IEEE/RSJ Int. Conf. Intell. Robots Syst. (1994), pp. 298–305

[10] Masoud, A., Masoud, S., Bayoumi, M., "Robot navigation using a pr essure generated mechanical stress field", the biharmonical potential approach, Int. Conf. Robot. Autom. (1994) pp. 124–129

[11] Singh, L., Stephanou, H., Wen, J., "Real-time robot motion control with circulatory fields", IEEE Conf. Robot. Autom. (1996) pp. 2737–2742

[12] Borenstein, J., Koren, Y., "The vector field histogram– fast obstacle avoidance for mobile robots", IEEE Trans. Robot. Autom., 7, 278–288 (1991)

[13] Minguez, J., "The obstacle restriction method (ORM): obstacle avoidance in difficult scenarios", IEEE Int. Conf. Intell. Robot Syst. (2005)

[14] Feiten, W., Bauer, R., Lawitzky, G., "Robust obstacle avoidance in unknown and c ramped environments", IEEE Int. Conf. Robot. Autom. (1994) pp. 2412–2417

[15] Fox, D., Burgard, W., Thrun, S., "The dynamic window approach to collision avoidance", IEEE Robot. Autom. Magaz. 4(1), pp. 23–33 (1997)

[16] Fiorini, P., Shiller, Z., "Motion planning in dynamic environments using velocity obstacles", Int. J. Robot. Res. 17(7), 760–772 (1998)

[17] Simmons, R., "The curvature-velocity method for local obstacle avoidance", IEEE Int. Conf. Robot. Autom. (1996) pp. 3375–3382

[18] Minguez, J., Montano, L., "Nearness niagram (ND) navigation: collision avoidance in troublesome scenarios", IEEE Trans. Robot. Autom. 20(1), 45–59

[19] Minguez, J., Osuna, J., Montano, L., "A divide and conquer strategy to achieve reactive collision naviigation systems", IEEE Int. Conf. R obot. Autom. (2004)

[20] Rodić, A., Vukobratović, M., Dynamics, Integrated Control and S tability of Automated Road Vehicles, Ibidem Verlag, Stuttgart, ISBN: 3-89821-203-3, 2002.

[21] Pacejka, H., Bakker, B. „The Magic Formula Tyre Model", Vehicle System Dynamics", Vol. 21, ISBN 90-265-1332-1, 1993

[22] G. Mester, „Intelligent Mobile Robot Controller Design", Proceedings of the Intelligent Engineering Systems", INES 2006, pp. 282-286, London, United Kingdom, 2006.

[23] Rodić, A., Mester, G., "Virtual WRSN – Modeling and Simulation of Wireless Robot-Sensor Networked Systems", Proc. of 8th Int. Symp. on Intelligent Systems and Informatics, September, 2010, Subotica, Serbia

Aleksandar RODIĆ received his B.Sc. in Mechanical Engineering in 1985, and his M. Sc. and P h.D. degree in electrical engineering in 1992 and 1998 f rom the University of Belgrade, Serbia. Since 1987 he was with the Robotics Laboratory of Mihailjo Pupin Inst., where he is currently a Scientific advisor and Head of Robotics department. He is visiting professor at the TU Braunschweig, Germany (2001/2002, 2005), University of Reunion, France (2004, 2006, 2008, 2009 and 2011), and adjunct professor of the University of Szeged (Hungary). From 2010 he is member of the Scientific Council for Technology Development with Ministry of Science of Republic Serbia His research interests includes modelling and simulation of large-scale dynamic systems, identification, adaptive/non-adaptive control, robotics and ar tificial intelligence.

Cooperative Multi Robot Systems for Contemporary Shopping Malls

Katić, M., Duško

Abstract— ***The objective of this paper is to show a new concept of robust and adaptive cooperative-collaborative multi-robot service architecture for contemporary mega stores. The research aim is design and devlopment of methods and technologies for integration of cognitive characteristics of cooperative robots with advanced recognition techniques (vision, RFID). The particular objectives includes the development of methods for cooperative perception and SLAM, the development of task allocation algorithms and navigation strategies, as well as development of cooperative learning. The porposed multi-robot platform enables the different functional scenarios: cooperative distribution stock monitoring, cooperative RFID inventory control, cooperative robot formations, human-robot assistance.***

Index Terms — ***Cognitive Systems, Collaboration, Cooperation, Multi Robot Systems. Servce robotics, Shopping Malls.***

1. INTRODUCTION

Multi-Robot Systems are distributed systems which consist of a m ultitude of networked robots and ot her devices and which, as a whole, are capable of interacting with the environment through the use of perception and action. The main feature of these systems is the use of robust distributed intelligence.

One of these important scenarios for multi robot systems, which are in need of such enabling technology is the retailing industry. Due to increasingly complex market conditions, the retail sector needs to optimize processes, lower costs and es tablish ranges and services tailored to individual customers. The principle of cooperation, theory of collaboration and advanced multi-agent and multi-robot architectures represent the attractive research fields for the solution of diverse problems in service applications such as shopping mega stores.

The motivations and i deas for developing multi-robot system solutions include: 1) the high complexity of the tasks that a single robot cannot accomplish; 2) the tasks in mega stores are inherently distributed; 3) building several restricted robots is easier than having a single powerful robot; 4) multiple service robots can solve control problems faster using parallelism; and 5) with the introduction of multi-robot teams, robustness and fault tolerance are increased based on redundancy. On the other side, robot teams in these applications can replace human repetitive work in everyday tiresome and m onotonous jobs.

Manuscript received December 30, 2010.

Duško Katić. is with the Robotics Laboratory, Mihailo Pupin Institute, University of Belgrade, Belgrade, Serbia (e-mail: dusko.katic@pupin.rs).

2. STATE-OF-THE-ART IN MULTI-ROBOT SYSTEMS FOR SHOPPING MALLS

Network Multi-Robot Systems is a relatively young research field, still quite dynamic and open t o ideas and c oncepts coming from other research areas while initial previous research in the area of *multi-robot systems* has been done on gr oup behaviour, collective behaviour, cooperative behaviour, swarm robotics and m ulti-agent robotic systems [1-3]. Under EU FP6 and FP7, we are aware of a few projects from the area of multi-robotic systems: *I-SWARM*, Intelligent small world autonomous robots for micro-manipulation [4], *URUS*, Ubiquitous Networking Robotics in Urban Settings [5], *GUARDIANS*, Group of Unmanned Assistant Robots Deployed in aggregative navigation supported by scent detection [6], *ROBOSWARM*, Knowledge Environment for Interacting ROBO***t*** SWARMs [7] , *IWARD* , Intelligent **robot** swarm for attendance, recognition, cleaning and delivery [8], *DUSTBOT*, Networked and Cooperating Robots for Urban Hygiene [9], *SWARM-BOTS*, Design **and** implementation **of** self-organizing **and** self-assembling artefacts [10]. These projects can be divided into two groups: one oriented towards swarm robotics and swarm intelligence, and the other oriented towards precise application scenarios for urban and security purposes. Major practical results in the area of multi-robot systems include: (1) mobile robots performing coordinated movement (flocking, foraging, following, etc.); (2) significant fault tolerance in heterogeneous multi-robot teams; (3) heterogeneous and homogeneous multi-robot box pushing; (iv) homogeneous (swarm-type) box pushing; (4) automated construction (wall-building); (5) successful robot soccer teams on multiple types of platforms; (6) self-reconfiguring robot colonies; (7) Milli-bots for cooperative mapping and exploration; (8) robots for data gathering tasks; (9) cooperative air and ground robot cooperation; (10) large-scale heterogeneous teams performing indoor localization, mapping, object detection, and surveillance; (11) swarm behaviours on large number of physical robots.

2.1 Robotic Systems for Modern Retailing

The increasing interest in introducing modern ICT technologies into the contemporary retailing companies and business is the best illustrated through the new projects and Initiatives and new technology review given below. For instance, the METRO Group Future Store Initiative [11] is an attractive testbed for novel research on difficult future problems in retailng industry. Developing innovative technologies is central to these dynamic challenging mega stores, while they offer a more comfortable and customer-friendly shopping experience as well as increasing retailing efficiency. Until today, only individual applications of new technologies and concepts have been employed in trading and retailing; companies are for the first time linking innovative technologies in a complex way. The main technological focus is put on using Radio Frequency Identification (RFID) technology.

RFID Tags (Fig.1). Smart tags or RFID (radio frequency identification) tags are being used to provide more transparency in the distribution systems of major companies.

Fig.1: RFID Tag

The Smart Chip, or transponder, is at the heart of RFID. In retailing, special tags are attached to pallets, boxes or clothes. The chip has a combination of numbers saved on it, the Electronic Product Code (EPC). This makes objects uniquely identifiable. Special RFID readers record the EPC and enter it into a database. This database holds information about manufacturer, price or dispatch date, for example, and this information is then available to authorized users. The most notable example is Wal-Mart in the USA who has invested heavily in the implementation of such systems. The company database systems can keep track of which items are in stock and the exact stock levels. The location of individual products, packing crates or pallets may be recorded and for example, with food products, use–by dates and sell–by dates are easily identified. One recent development is in the field of smart shelves in which the display shelves are equipped with radio-frequency readers that track RFID-tagged products. The readers may then alert employees when items need restocking or if products have been placed on the wrong shelves. Modern retailing in Future Store Initiative currently, includes other future technologies as information terminals, cell phone shopping assistants, self-checkout, musical terminal, etc.

REEM-H2 (Fig.2), the humanoid service robot created by PAL Robotics [12], can be used for shopping malls. Thanks to its autonomous navigation system, its user - friendly touch screen, and its voice and face recognition system, REEM-H2 can find its way in various surroundings and help or entertain people in most public environments.

Besides helping you as a guide or amusing you as an entertainer, REEM-H2 can also transport small packages, and its dynamic information point can be used with a wide variety of multimedia applications: display an interactive map of the surrounding area, call up a variety of information (weather, nearby restaurants, airlines travel time, etc...), offer tele-assistance via video-conferencing.

Fig.2: REEM-H2 Humanoid Robot

In the papers [13-16], it was reported development of a single and multiple communication robots for use in a shopping mall to provide shopping information. They also offer route guidance, and build rapport. In the development, the major difficulties included sensing human behaviours, conversation in a noisy daily environment, and the needs of unexpected miscellaneous knowledge in the conversation. The network - robot system approach was chosen, where a single robot's poor sensing capability and knowledge are supplemented by ubiquitous sensors and a human operator. The developed robot system detects a person with floor sensors to initiate interaction, identifies individuals with radio-frequency identification (RFID) tags, gives shopping information while chatting, and provides route guidance with deictic gestures. The robot was partially teleoperated to avoid the difficulty of speech recognition as well as to furnish a new kind of knowledge that only humans can flexibly provide. The information supplied by a human operator was later used to increase the robot's autonomy.

Fig.3: Robot guiding a customer in shopping mall

As second approach, it was developed a networked robot system that coordinates multiple social robots and sensors to provide efficient service to customers. It directs the tasks of robots based on their positions and people' s walking behavior, manages the paths of robots, and coordinates the conversation-performance between two robots. Laser range finders were distributed in the environment to estimate people' s positions. The system estimates such human walking behaviors as "stopping" or "idle walking" to direct robots to provide appropriate tasks to appropriate people. Each robot interacts with people to provide recommendation information and route information about shops. The system sometimes simultaneously uses two robots to lead people from one place to another. The field trial, which was conducted in a shopping mall where four robots interacted with 414 people, revealed the effectiveness of the network robot system for guiding people around a shopping mall as well as increasing their interest.

Wurman, D'Andrea and Muntz [17-18] applied the concept of distributed intelligence to inventory management. Inspired by air traffic controllers capable of coordinating the arrivals and departures of a big-city hub, they created a system that allows inventory to organize itself, adapting to conditions as they change. D'Andrea and Wurman's powerful software uses a fleet of intelligent robotic drive units to shuttle inventory between storage areas and workers at picking, packing and shipping stations, eliminating the need for workers to go to the inventory themselves.

The result is an incredibly efficient, accurate and flexible distribution chain with no single point of failure. In use by distribution giants like Walgreens and Staples, Kiva's 'intelligent warehouse' [19] is helping distributors fill two to three times as many orders as they could with conventional methods.

Fig.4: Kiva Systems Drive Units

3. *MULTI-ROBOT CONCEPT FOR SHOPPING MALLS*

The main concept of multi-robot systems for shopping malls is based on the representation of a multi-purpose robotic system for service applications with advanced perception and action capabilities [20-23]. The proposed concept is strictly oriented towards the design of multi-robotic systems whose entities have *shared and common* goals. The second important characteristics of the multi-robot system are related to the capability of *awareness of other robot entities.* With the word *aware*, we refer to whether entities reason about the actions and intentions of their teammates. Finally, a single entity's actions *help to advance the goals of other teammates*.

The previously defined characteristics define the multi-robot systems according to the type of interactions between robot entities. The proposed system represents a type of *cooperative interaction,* in which robot entities are aware of other robot entities. They share goals and their actions are beneficial to their teammates. In these systems, robots may at times be working on different parts of the higher level goal, and thus may at times have to ensure that they share the workspace without interfering with each other. However, the majority of the work of the robots is focused on working together to achieve a common goal.

However, the proposed system considers also *collaborative interaction* between robot team mates. When robots have individual goals, they are aware of their teammates, and their actions do help advance the goals of others. Each team member has his/her own goal of performing task, but by working together with others with complementary expertise, each can help other members to better achieve their individual goals. Of course, most of these collaborations are also cooperative, and it is possible to turn a collaborative team into a cooperative team by simply viewing the team goals from a higher perspective.

The general concept of proposed multi-robotic cooperative-collaborative robotic system is based on a *distributed dynamic hybrid architecture* presented in Fig. 5. It consists of individual robot-agents and groups of agents that are networked, capable of acquiring and processing information from their surroundings, communicating between themselves, and sharing knowledge within the team. Robot agents, as individuals, possess advanced cooperative perception, cooperative and communicative action characteristics.

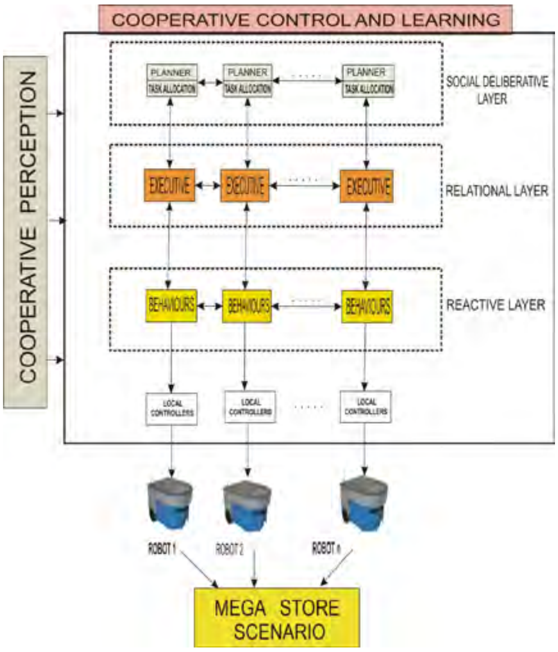


Fig 5: Distributed Hybrid Architecture of an multi-agent system for mega store scenarios

Hybrid control architecture combines local controllers with higher-level control approaches to achieve both robustness and the ability to influence the entire team's actions through global goals, plans, or control. This approach is based on layered architectures, where each robot's control architecture consists of a *planning or social deliberative* layer that decides how to achieve high-level goals; an *executive or relational* layer that synchronizes agents, sequences tasks, and monitors task execution; and a *behavioural* layer that interfaces to the robot's sensors and effectors. Each of these layers interacts with those above and below it. Additionally, robots can interact with each other via direct connections at each of the layers.

On the *social deliberative level*, system behaviour that allows the team to cope with the environmental changes provides a strategy that can be adopted to reorganize the team members' tasks. On the *reactive layer*, every single robot in the team copes with the environmental changes by providing a specific solution to reorganize its own task in order to fulfill the assigned goal. This layer enables realization of single *primitive tasks* or of *composite tasks* (primitive tasks linked by logical conditions on event). For the *relational layer*, it is important to maintain relationships between the robot and its teammates; for the *social deliberative layer*, the main aim is to find an appropriate strategy for task decomposition and task allocation. The proposed system represents a *MR-ST*-IA **taxonomy** according to task allocation classification. MR (multi-robot tasks) denotes that there is more than one robot working on the same task at the same time. *ST* (single-task robots) denotes work of a single robot on only one task, while **IA (**instantaneous task allocation) denotes that tasks are assigned to optimize the instantaneous allocation of tasks.

The collaborative behaviour of multi-robot teams is based on *formal models and techniques that* have been developed to build successful cooperative multi-robot systems and to provide solutions for several types of problems. For the problem of cooperative spatial perception based on distributed sensors, new formal models based on probabilistic (Bayesian) approaches together with qualitative and logic-based representations will be considered. As a second characteristic, formal models for multi-robot plans provide a significant step forward in defining suitable solutions for cooperation and collaboration.

The high-level tasks in the social deliberative layer will be achieved using specific paradigms of robust distributed intelligence. The fundamental challenge is to develop an appropriate paradigm for determining how best to achieve global coherence from the interaction of entities at the local level. The first basic paradigm of our architecture is *the behaviourist approach* to autonomous multi-robot control. Rather than decomposing the robot control system based on information processing functions, the behaviourist approach decomposes the high control behaviours into tasks achieving local reactive behaviours, such as obstacle avoidance, exploration, and map building. The result is a series of autonomous robots that can survive in a dynamic world, avoiding obstacles, exploring the environment, following walls, building maps, and so forth.

The *organizational/social paradigms* are additional techniques that have been used in to create similar higher-level, intentional cooperation and/or collaboration in multi-robot teams. ***Organizational** and **social*** paradigms are typically based on an organizational theory derived from human systems. In these approaches, agent/robot interactions are designed by modelling individual and group dynamics as part of an organization. The proposed system will consider three different methods of *organizational/social paradigms* for the realization of collective behaviour of multi-robot system. The first one is the *use of roles*. *Use of Roles* are often used to divide a system into manageable working areas that can each be assigned to a different robot in the team. An easy division of work is achieved by assigning roles according to the skills and capabilities of the individual team members. The second one is using *Market econom**ies** a*s a paradigm for *task allocation*. Multi-robot task allocation is the problem of mapping tasks to robots, such that the most suitable robot is selected to perform the most appropriate task. Market-based approaches to task allocation make use of the theory of market economies to determine how best to allow robots to negotiate responsibilities in the mission. The last method is using *Teamwork models* that allow agents/robots to explicitly reason about coordination and communication. In dynamic environments, the ability to reason about the interactions of agents/robots can enable the team members to reorganize themselves as needed to address new situations that arise. Protocols for establishing team member commitments are determined as part of this general model.

Behaviour of multi-robot systems in different shopping environments and conditions is the subject of permanent, on-line *cooperative perception*, *cooperative planning and action*, *cooperative learning and experience* that robots acquire or get from humans. By being able to communicate, robot-agents collect and share information obtained from their sensor systems, as well as knowledge obtained through learning of cooperative activities. Learning will be especially important in behaviour-based systems to adapt task assignment in robot teams and to deal with individual robot capabilities that change over time. Specific temporal-difference approaches and reinforcement learning are very important in this context.

4. ADVANCED MEGA STORE SCENARIOS

The proposed multi-robot system is oriented toward *generating enabling new technology in realistic environments*, such as advanced mega stores with shopping ***scenario models***. Such environments show different characteristics of collaborative behaviour depending on environment-specific conditions or scenario-specific situations. These scenarios will typically be set in real-world environments with a need for special functionalities such as: exploration, monitoring, robust and adaptive navigation, obstacle avoidance, and human robot interaction.

The proposal for new cognitive methods and technologies in cognitive multi-robots systems will collaborate with megastores, such as the METRO Group Future Mega Store [11] and can build on and benefit from initial results of the Future Store Initiative. So far mega stores have realized simple solutions for simple problems but in order to make progress in collaborative perceptional robotics and monitoring, a research quantum jump is needed in terms of necessary research on vision, navigation, communication and collaboration.

The proposed cognitive architecture aims at integrating the area of multi-robots systems with learning sensor fusion for vision and RFID technology developed and trial this generic technology in the Future Store Initiative. In particular modern real time image recognition techniques that use huge database of 200000 products may become possible with acceptable costs. In this way, our embedded and innovative multi-robot system enabling technology can drive important service application such as retailing for the future, in order to make this a tangible experience and to highlight the benefits for the business community as well as for consumers.

The Main Cooperation Tasks–Scenarios [Fig.6]

1. *<u>Cooperative Vision Distribution Stock Monitoring</u>*. Multiple mobile robots patrol the shopping floor, use visual processing to identify product gaps on shelves, collect information and provide it via a collaborative agent to the store manager at regular intervals so that store operations can optimize the resources to replan all identified gaps. This most important monitoring function could be assisted with a bar code scheme, that is shown on each shelf label indicating the “to-be” product in that area, while a visual check of those products available in that area in comparison with the product picture stored in the merchandise management system, would provide evidence if that product is still available.

2. *<u>Cooperative RFID Inventory Monitoring</u>*. Multiple mobile robots will be equipped

with an RFID-based infrastructure which provides all RFID readings from a shelf area that is marked with a product barcode or RFID tags. RFID can realize an automated inventory count dependent on item tagging. In this way, the autonomous inventory system each day gets the exact picture of products in the store. That means that ordering, product placement and category management can be supported much better towards t heir optimal performance level.

3. *Complementary human robot assistance* is performed by a robot-info/delivery agent. A robot is a guide for a hum an to find desired goods. The subtask is tracking the human-customer agent and assisting him/her to carry the goods. The human customer chooses the articles and puts them into the trolley (small wagon pulled by the robot) while being tracked by the robot.

4. *Cooperative traffic control.* Negotiation and consensus decisions happen when some robot-agents meet each other at a narrow corridor (e.g. warehouse door). Negotiation and consensus algorithms enable agents to "agree" about the order of passing. In this situation, the system priorities determine the way of solving the problem. Cooperative formation is performed by the collective movement of robot team members from warehouse to shopping floor, and vice versa.

5. MAIN OBJECTIVES

The main aim is to provide a new approach for formal modelling, design and ev aluation of cooperative-collaborative multi-robot systems in order to achieve the realization of complex service tasks. The focus of the is on enabling technology for service robotics with special emphasis on scalable dynamic environments like the *shopping mall scenario model*, with multi-agent robotic systems employed as a cooperative teams performing various functional tasks such as visual stock monitoring, RFID inventory keeping, visual inspection, robot formation and human robot assistance tasks. The important aim is the design of new distributed hybrid architecture for a m ulti-robot system integrating mobile robot coordination with advanced intelligent vision-barcode-RFID recognition technology. This hybrid distributive architecture will handle both dynamic and static environments of advanced mega store scenarios.

We have identified several well defined scientific (S), technological (T) and implementation (I) objectives:

S.O.1: Development of new formal sensor fusion models of cooperative 3D perception for shopping environments. New approaches and new formal models will be developed that consider problems of robust real-time information fusion in the cases of time-varying and space-changing sources. The innovation will consist of the development and implementation of cooperative perception tools to integrate the local perception of robots (based on visual cameras, infrared cameras, range sensors, and RFID readers) along with the information provided by the environment sensors.

Fig. 6: Advanced Mega Store Scenarios

S.O.2: Development of advanced mobile robot vision algorithms for monitoring and exploration in modern shopping floors. This task considers detection, recognition and classification of objects of stock, and irregular situations, in real shopping environments together with the interpretation of complex environmental scenes.

S.O.3: Development of specialized algorithms for simultaneous localization and mapping of static and dynamic semi-dynamical shopping environment. Different map representations (topological, grid-based, and geom etrical representations) will be considered that are possible and s uitable for mega store enabling technology in relation to the sensors mounted on the mobile robots. Developed robust algorithms must include a segment-based representation for "static" elements and a point-based representation for "transient" obstacles.

S.O.4: Development of new formal models of cooperation and advanced algorithms for multi-robot task allocation based on behaviourist / organizational/ social paradigms**.** The research for optimal task allocation based on the organizational/social paradigm will try to identify hybrid solutions using different methods as they are used in roles, market economies or teamwork models**.**

S.O.5: Development of efficient navigation strategies for mobile robots. New navigation strategies for robots will involve a multi-objective optimization problem in which the different objectives can be t he length of the paths, the time needed to satisfy requests, the avoidance of conflicts between robots, and so on. Using this approach, developed efficient navigation strategies will send the robots towards locations where, for example, it is more probable to have missing items.

*S.O.6: Development of specialized reactive behaviour control algorithms***.** We will consider the synthesis of simple and complex primitives of reactive robot behaviours, the result will be a number of autonomous robots that can efficiently work in a dy namic environment, avoiding obstacles, exploring the environment, following lines, and building maps.

S.O.7: Development of advanced reinforcement learning techniques for multi-robots in mega stores***.*** New approaches and algorithms for learning specialized cooperative behaviour will be af fected by specific challenges from mega stores scenarios, like multiple goals, noisy perception and actions, and inconsistencies in the internal states and in environment models between the individual robots.

S.O.8: Development of advanced learning techniques performing by examples as well as based on previous experience. Besides developing learning techniques that will allow the robot to learn by experimentation, we will study reinforcement learning from the final feedback. New learning algorithms must include a s olution for several major open problems in multi-robot learning including modelling formal properties of real worlds, dealing with team credit assignment, convergence time of learning algorithms, and coping with dynamical environments including other robots learning.

S.O.9: Definition of a standard benchmark for generic dynamic environments and development of a methodology for performance evaluation and comparison. The goal is to come up with special metrics and benchmark that enables a systematic performance evaluation of the proposed cooperation models and control and l earning methods. The metrics can also be used in cooperative reinforcement learning algorithms for evaluating rewards. The aim is to develop criteria for benchmarking system properties such as robustness, scalability and adapt ivity, depending on the particular scientific issues addressed.

T.O.1. Integration of a versatile, affordable functional multi-robot platform for mega store applications. This technical objective will be based on advanced sensor systems (vision systems, RFID readers, laser scanners, ultrasound sensors) to enable versatile cooperative perception. A wireless communication module enables online communication between members of robot teams and a c entralized operator system in mega stores. Groups of advanced mobile robots will enable high manoeuvrability and autonomy of the robotic system.

T.O.2: Development of control software architecture. The control software architectures integrate a user-interface (command-interface) to enable general system supervision, as well as task allocation and remote control in the case of a semi-autonomous regime of operation in a shopping mall. P lanning and C ontrol strategies and algorithms are based on deliberative, relational and reactive layers intended for cooperative perception and SLAM, cooperative navigation, cooperative planning and cooperative learning. Communication protocols will be established for communication between members of a team and the operational console.

I.O.1. Implementation and Demonstration objectives. These objectives are related to providing and dem onstrating the operation of *scalable mega-store test-**scenarios*** based on our collaborative behaviour paradigm, and a *high-tech functional platform* of a cooperative-collaborative multi-agent system. **Desired**

functionality of the proposed system includes efficient multi-agent collaborative robot teams with advanced cognitive capabilities in advanced shopping environments that perform different cooperative tasks, such as distributed condition monitoring of stock, RFID inventory monitoring, space inspection and human robot assistance. In the scope of the scenario to be experimentally tested, these tasks will form the meaningful benchmark with specific metric for system performance measurement.

6. CONCLUSION

In this paper, a new concept of multi robot architecture for modern shopping malls is presented. This architecture considers integratio of intelligent cognitive characteristics of mobile robot team with contemporary recognition techniques (vision, RFID). The proposed resarch area is a v ery atractive reasearch field in frame of intelligent multi robot systems].

REFERENCES

[1] Butenko, S., Murphey, R., Pardos, P. M., eds. “Cooperative Control: Models, Applications and Algorithms””, *Kluwer Academic Publishers, 2*003.

[2] Lesser, V. R. ”Cooperative multiagent systems: A personal view of the state of the art””, *IEEE Transactions on Knowledge and Data Engineering* “, 2002, pp.133–142.

[3] Izquierdo-Torres, E., “Collective Intelligence in Multi-Agent Robotic”s: Stigmergy””, Self-Organization and Evolution*. University of Sussex*. 2004.

[4] *I-SWARM,* Intelligent small world autonomous robots for micro-manipulation, FP6-507006, *http://www.i-swarm.org*

[5] *URUS,* Ubiquitous Networking Robotics in Urban Settings, FP6-045062, *http: //wwwiri.upc.es/ groups/ urus/*

[6] *GUARDIANS,* Group of unmanned Assistant Robots Deployed in aggregative navigation supported by scent detection, FP6-045269, *http://www.guardians-project.eu/*

[7] *ROBOSWARM,* Knowledge Environment for Interacting ROBOt SWARMs,FP6-04525, *http://roboswarm.eu/*

[8] *IWARD* , Intelligent robot swarm for attendance, recognition, cleaning and del ivery, FP6-045254, *http://www.iward.eu*

[9] *DUSTBOT,* Networked and C ooperating Robots for Urban Hygiene, FP6-045299, *http:// www.dustbot. org*

[10] *SWARM-BOTS***,** Design and implementation of self-organizing and self-assembling artifacts. FP6-IST-FET, *http://www.swarm-bots.org*

[11] Future store initiative, *http://www.metrogroup.de/*

[12] *http://www.pal-robotics.com/robots*

[13] Kanda, T., Shiomi, M., Miyashita, Z., Ishiguro, H., Hagita, N.,” A Communication Robot in a Shopping Mall”, *IEEE Transactions on Robotics, 2010, pp.897- 913.*

[14] Shiomi, M., Kanda, T,, Glas, D. F. , Satake, S., Ishiguro, H., Hagita, N.,” Field Trial of Networked Social Robots in a S hopping Mall”,2009 IEEE/RSJ International Conference on I ntelligent Robots and Systems, 2009, pp. 2846-2853.

[15] Sakamoto, D., Hayashi, K., Kanda, T., Shiomi, M., Koizumi, S.,Ishiguro, H., Ogasawara, T., Hagita, N., “Humanoid Robots as a Broadcasting Communication Medium in Open Public Spaces*”, International Journal of Social Robotics, 2*009, pp. 157-169.

[16] Kanda, T., Shiomi, M., Miyashita, Z., Ishiguro, H., Hagita, N., “An affective guide robot in a s hopping mall”, 4th ACM/IEEE International Conference on Human-Robot Interaction, 2009, pp.173-180.

[17] Wurman, P., D’Andrea, R., Mountz, M.,.”Coordinating hundreds of cooperative, autonomous vehicles in warehouses”. In *Prof.Conference on Innovative Applications of Artificial Intelligence, pages*, 2007, pp. 1752–1757.

[18] Wurman, P., D’Andrea, R., Mountz, M.,”Coordinating hundreds of cooperative, autonomous vehicles in warehouses””. *AI Magazine*, 2008, pp.9–19.

[19] *http://www.kivasystems,com*

[20] Katić, D., Rodić, A., ‘’Intelligent Multi Robot Systems for Contemporary Shopping Malls”, 8th International Symposium on I ntelligent Systems and Informatics SISY’10, 2010, PP.109-113

[21] Mester, G., "Intelligent Mobile Robot Control in Unknown Environments", *Intelligent Engineering Systems and Computational Cybernetics, Part I Intelligent Robotics, Springer*, 2009. pp. 15-26,

[22] Rodić, A., Katić, D., Mešter, G., “Ambient Intelligent Robot-Sensor Networks for Environmental Surveillance and Remote Sensing”, *7th International Symposium on Intelligent Systems and Informatics SISY’09,2009*.

[23] Mester, G., "Sensor Based Control of Autonomous Wheeled Mobile Robots", *The Ipsi BgD Transactions on Internet Research, TIR*, 2010, pp. 29-34.

Duško M. Katić(M’93–SM’99) received the B.Sc. and M Sc. degrees in mechanical engineering from the University of Belgrade, Serbia, in 1982 and 1987, and the Ph.D. degree in electrical engineering from the University of Belgrade, Serbia in 1994. Since 1983, he has been a researcher in the Robotics Laboratory, Mihailo Pupin Institute, Belgrade, Serbia, where he i s currently a scientific director and project manager. Primary research interest of Dr Katić is in areas of intelligent control of industrial, service and humanoid robotics, multi-agent systems and cognitive robotics. He is the author of more than 100 scientific papers in the domestic and international conferences, journals and co-author of 2 books ‘’Intelligent Control of Robotic Systems’’ (Kluwer, 2003) and “Dynamics and Robust Control of Robot-Environment Interaction” (World Scientific, 2009). He a senior member of IEEE and f ounding Chair of Serbia & Montenegro Chapter of IEEE Computational Intelligence Society. He is also a member of Serbian Academy of Engineering Sciences.

Use of Support Vector Machine for Humanoid Robot Motion Synthesis

Borovac Branislav; Raković Mirko and Nikolić Milutin

Abstract— This paper presents the application of a Support Vector Machine (SVM) in humanoid robotics. Two different examples were considered: the adaptation of the recorded human motion (to make it suitable for emulation by humanoid robot), and the generation of robot's arm motion in a reaching task. In the first case, the motion of a human subject (under disturbance) was recorded with the aim to apply it to a humanoid robot subjected to the same disturbance. In view of the fact that the parameters of the human subject and robot differ among themselves, the recorded movement had to be adapted to the robot. The basic requirement imposed on the approximated motion is the preservation of dynamic balance, where the position of the Zero Moment Point has to be constantly within the support area. This was the basic criterion for the approximation of the evaluation. The other task illustrating the application of SVM was to train the system for generating the arm's motion when the robot is to reach an object. It was shown that the SVM can be very effectively trained and used for online generation of driving torques of arm joints, based on the visual feedback about the position of the object to be reached.

Index Terms— ***dynamic balance, humanoid robot, motion approximation, reaching task, SVM regression***

1. *Introduction*

PERFORMANCES of humanoid robots are always compared with those of humans. This is particularly true for motion evaluation. One of the approaches (appropriate for complex movements) is to use the movements recorded for humans and apply them to robots. However, to achieve the same effect, the recorded movements have to be adapted, as the robot's parameters differ from those of the human subject.

This paper presents the use of Support Vector Machine (SVM) in two different tasks. A first one is the prevention of overturn (preservation of dynamic balance) in case of a large disturbance. Such action involves human's entire body and motion of all joints is synchronized so to keep the Zero Moment point (ZMP) within a very narrow support area. To apply it to humanoid, the recorded motion has to be adapted (adjusted to "humanoid's body"), to ensure preservation of dynamic balance[1].

The second task is the realization of reaching. Reaching movement has to be done online, without hesitation and with high motion efficiency (movement should be fast, with a significant slowdown when approaching the object). In other words, despite of the fact that the information about what has to be reached and where the object is 'forwarded' to the control system suddenly (when the visual system has located the object) the movement has to be realized effectively and without hesitation, starting from the instantaneous position of the arm. We present the use of the SVM for the case the humanoid has to be 'trained' how to reach the object efficiently, starting from different positions within the arm's working space.

2. *SVM Regression*

There are a number of algorithms for establishing the unknown interdependence between the input and output data. To determine an unknown interdependence, it is necessary to minimize some of the error functions. The majority of the algorithms minimize the empirical error:

$$R_{\text{emp}} = \frac{1}{l}\sum_{i=1}^{l} L\left(\mathbf{y}_i, f_a\left(\mathbf{x}_\mathbf{i}\right)\right) \tag{1}$$

where l is the number of data serving as the basis for the approximation; $\mathbf{y}_i$ is the desired, while $f_a\left(\mathbf{x}_\mathbf{i}\right)$ is the approximated value at the output; L is the cost function, which may be linear, quadratic, or some other norm. Vapnik [1] has introduced a general type of the error function:

$$\left|y - f_a(\mathbf{x},\mathbf{w})\right|_\varepsilon = \begin{cases} 0, & \left|y - f_a(\mathbf{x},\mathbf{w})\right| \le \varepsilon \\ \left|y - f_a(\mathbf{x},\mathbf{w})\right| - \varepsilon, & \left|y - f_a(\mathbf{x},\mathbf{w})\right| > \varepsilon \end{cases} \tag{2}$$

called the error function with the ε-zone of insensitivity. In (2), the error is equal zero if the difference between the approximate and the given value is smaller than ε. It should be noticed that the error function with the ε-insensitivity zone defines an ε-tube around the output data.

With the function approximation algorithms that minimize only the empirical error, the problem of generalization arises. The problem appears when the training set is small compared to the number of different data that can appear at the input. Structural Risk Minimization (SRM) is a new technique of the statistical learning theory, which we will illustrate on the example of the linear approximation function:

$$f_a(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + \mathbf{b} \tag{3}$$

Apart from minimizing the empirical errors, the SRM also minimizes the generalization errors (by

[1] A necessary and sufficient condition for the dynamic equilibrium of locomotion system is that the ground reaction force is acting within the support surface area. Hereby, at that point the conditions $M_X = 0$, $M_Y = 0$ hold. This point is termed the Zero Moment Point.

minimizing elements of the weight matrix w). Hence, it follows that the structural error will be minimized by minimizing the function of the form:

$$R = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}|\mathbf{y}_i - \mathbf{w}\cdot\mathbf{x}_i - \mathbf{b}|_\varepsilon \quad (4)$$

In (4), the error function with the ε-insensitivity zone is used as an L norm. The parameter C is the penalty parameter which determines the extent to which the empirical error is penalized relatively to the penalization of the large values in the weighting matrix.

The SVM represents the implementation of the minimization of the structural error. The linear SVM regression (minimization of (4)) determines the elements of the weight w and bias b matrices. Since the linear relation between the input and output in real problems rarely exists, it is more often necessary to determine a nonlinear approximation function. Since the SVM easily solves the linear regression, the nonlinear tasks are then mapped to a space of higher dimensionality, in which the task becomes linear. The technique by which this is achieved is the kernel trick [2], which assumes the choice of the kernel function Φ that maps the input vector $\mathbf{x}\in R^n$ to the vector $\mathbf{z}\in R^f$, where the vector $\mathbf{z}$ belongs to a higher-dimensional space compared to the space to which belongs the vector $\mathbf{x}$. Hence $\mathbf{z}=\Phi(\mathbf{x})$, where Φ represents the mapping $R^n \to R^f$. The kernel function Φ is chosen in advance and it remains constant for the problem that is currently solved. It should be pointed out once more that the aim of introducing such mapping is to obtain a problem in the higher-dimensionality space, which can then be solved by linear regression[2]. The most frequently used kernel functions Φ are polynomials, Radial Basis Functions (RBF), sigmoid functions, etc.

3. ADJUSTMENT OF THE MOTION RECORDED FROM HUMAN FOR A HUMANOID

The primary task of any humanoid robot is to maintain dynamic balance [3]. Disturbances are always present. Compensation of small disturbances can be done by conventional techniques [4], but in case of larger disturbances (e.g. stumbling upon an obstacle, shoving aside, etc.), maintaining of dynamic balance becomes more complicated. Compensatory actions in humans mostly represent a coordinated, vigorous, and synchronized movement [4, 5]. After a vigorous motion aimed at preserving the dynamic balance, a human uses slow movement to restore the state from which he/she continues to perform the motion disrupted by the disturbance. In order for such motion to be performed by a humanoid robot (there exist differences in kinematic and dynamic parameters between human subject and humanoid robot) it has to be appropriately modified in such a way that the effects remain intact (the dynamic balance maintenance).

The location of the Zero Moment Point (ZMP) [6, 7] will be used as an indicator of the quality of maintenance of robot's dynamic balance. Besides maintaining the dynamic balance, the recorded data must also be modified so as to emulate the form of human motion for the same type of disturbance, and thus produce the same effect.

3.1 Data recording

In cooperation with the researches of the Holodeck Gait Laboratory, which is part of the Laboratory for Computer Science and Artificial intelligence at MIT (Massachusetts Institute of Technology), the data were recorded using the VICON 512 system which operates at 120 fps. For recording, 33 markers were employed, whose location was recorded with ~1 mm accuracy. Three adults participated in the experiment. Each person was asked to stand on his/her left foot while leaning against an obstacle with the left shoulder. The leaning force was measured and once it reached the limit of 20N the obstacle was abruptly removed, while the movement made in attempt to prevent the fall was recorded. The 20N limit was chosen so that the projection of person’s center of gravity falls outside of support surface. In this case, in order to maintain dynamic balance, each subject had to perform an energetic movement which brings the projection of the center of gravity back under the foot while the ZMP constantly remains within the support surface.

During every movement, force plate (Advanced Mechanical Technology Inc., Watertown, MA) was used to measure and record the location of ground reaction force, which in this case coincided with ZMP. The accuracy of the ZMP location measurement was approximately 2 mm. Each subject repeated the movement 10 times, so that a total of 30 movements were recorded [5]. In this paper, only the data of one recorded movement were used and processed.

3.2 Model of a humanoid robot

Kinematic structure of the model of a humanoid robot used in this experiment consists of four kinematic chains as shown in Fig. 1. The first kinematic chain represents the legs, the second forms the body and the right arm, the third represents left arm, while the fourth kinematic chain represents the neck and head. The joints with multiple DoFs (Degrees of Freedom) were modeled as a set of virtual links (links with zero mass and negligible length) connected by 1-DoF joints. For instance, the hip joint, which is a spherical joint with 3 DoFs, was modeled as a set of three 1-DoF joints whose axes of rotation are mutually orthogonal [8].

3.3 Motion approximation

The recorded motion and the measured ZMP location are shown in Fig. 2, where marker loca-

[2] Solution to the hypersurface $f=\mathbf{w}^T\Phi(\mathbf{x})+b$, which is linear in the space R^f, leads to a nonlinear hypersurface in the starting space R^n, to which belongs the input vector $\mathbf{x}$.

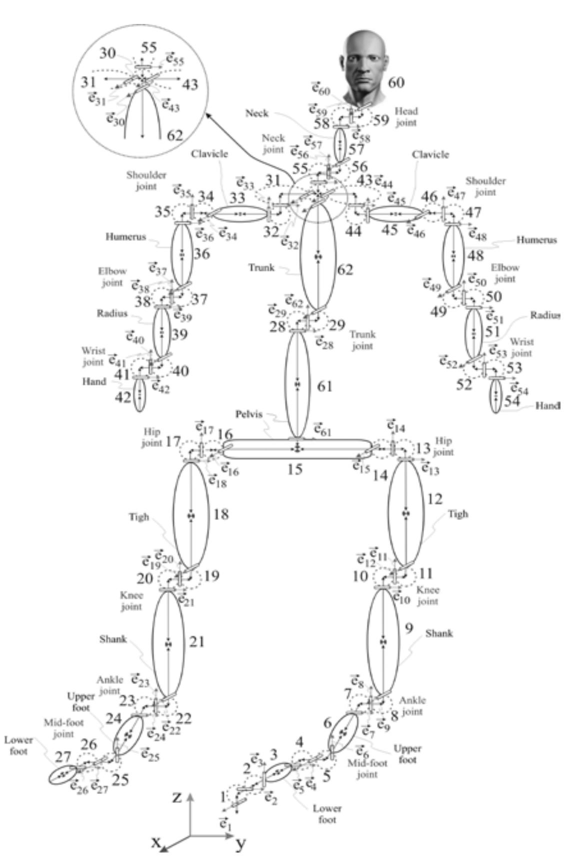


Fig.1 Mechanical structure of the model of a humanoid robot with 62 DOFs

tions are represented by small circles. It should be noted that the figure depicts a visualization of the recorded movement and illustrates the measured location of ZMP. It is obvious that the man very skillfully maintains the ZMP within the support surface.

If the recorded movement is applied on a humanoid robot, the ZMP location will significantly deviate compared with that of human subject movement, and the movement has to be adapted to suit the humanoid mechanism's parameters. Thus, the values of internal coordinate to be applied to humanoid should be approximated by smooth functions in such a way to preserve movement character while maintaining system's dynamic balance. This demands the values of velocities and accelerations in joints to be modified, because they have a direct impact on ZMP location [6, 7].

Two methods were used for the adaptation of the recorded motion (i.e. its approximation): the cubic spline approximation and SVM regression. The approximation was performed without taking into account the ZMP location (it was calculated just as a consequence of the motion). Fig. 3 shows stick diagram of humanoid robot motion obtained by cubic spline approximation with a smoothing parameter of 0.9 and corresponding ZMP trajectory. High smoothing parameter indicates the approximation which is very close to the recorded movement.

The motion approximation by SVM regression is illustrated in the following examples. Gaussian function of normal distribution was adopted for the kernel function. The simulation was performed for eight different cases, varying the values of ε-insensitivity zones and penalty parameter. Table 1 systematizes the results so that each combination of ε-insensitivity zone and penalty coefficient is paired with maximum deviation in the ZMP trajectory and maximum deviation of approximated values of internal coordinates from the recorded ones. It should be noted that the form of movement depends strongly on the value of ε-insensitivity zone. For higher values of ε, the movement loses its form, which was a basic reason for adopting just two values of ε-insensitivity zone: 0 and 0.01. It is obvious from the table that the increase of penalty coefficient lowers the maximum deviations of internal co-ordinates, but increases the ZMP deviations. Obviously, a trade-off must be established between SVM training parameters, i.e. between the deviations of ZMP and internal coordinates. This is illustrated in Figs. 4 and 5.

Let us first compare the cases shown in Fig. 4 and Fig. 5. In both cases the ε-insensitivity zone was 0.01. In the example shown in Fig. 4 the penalty coefficient was 1000, while in Fig. 5 it was 10, which reveals its influence on the ZMP trajectory.

The following illustrates how the increase of ε-insensitivity zone impacts the motion approximation. The case illustrated in Fig. 6 was additionally simulated. The approximation was also performed by SVM regression. The value of ε-insensitivity zone was 0.1, while the penalty coefficient was 10, just as in Fig. 5. A comparison of the cases shown in Figs. 5 and 6 reveals the loss of

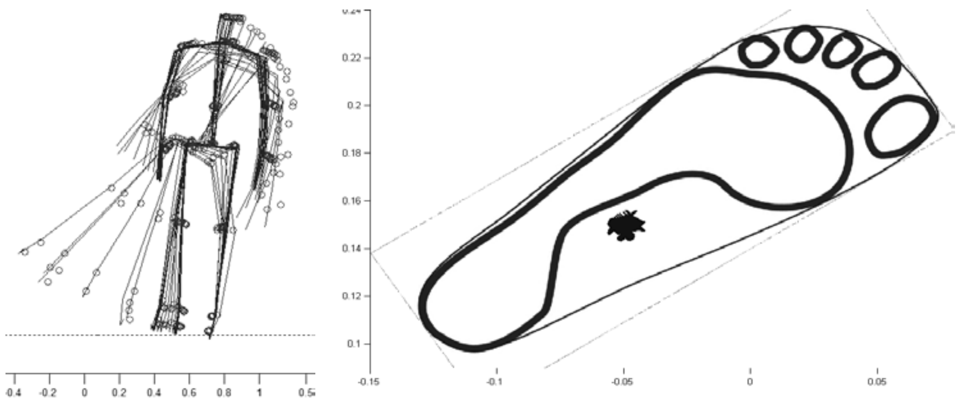


Fig. 2 Stick diagram and marker locations on a human (left); ZMP locations obtained by measurement (right);

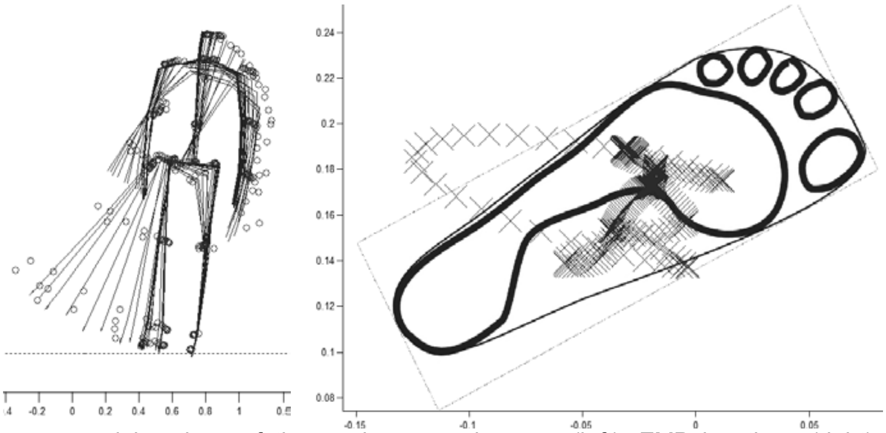


Fig. 3 Stick diagram and locations of the markers on a hum an (left); ZMP locations (right); the data were approximated by cubic splines with a smoothing parameter of 0.9

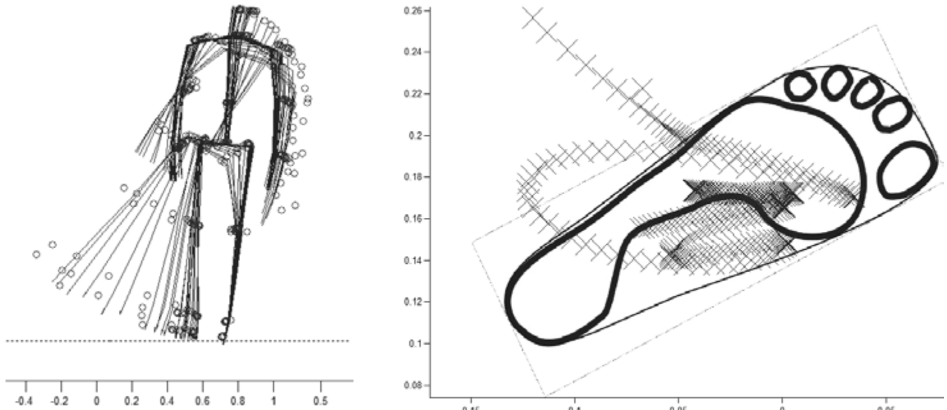


Fig. 4 Stick diagram and marker locations on a human (left); ZMP location (right); the data were approximated by SVM regression (ε-insensitivity zone equals 0.01, penalty coefficient is 1000)

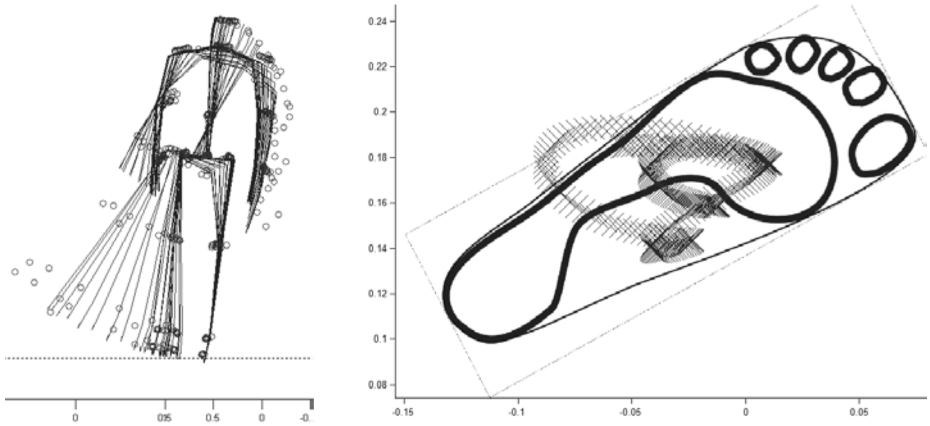


Fig. 5 Stick diagram and marker locations on a human (left); ZMP location (right); the data were approximated by SVM regression (ε-insensitivity zone equals 0.01, penalty coefficient is 10)

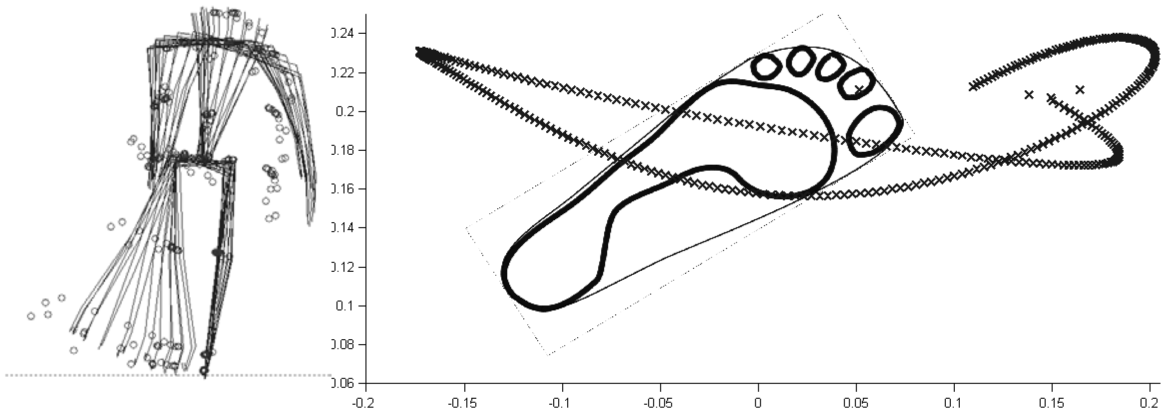


Fig. 6 Stick diagram and marker locations on a human (left); ZMP location (right); the data were approximated by SVM regression (ε-insensitivity zone equals 0.1, penalty coefficient is 10)

TABLE 4.1 INFLUENCE OF E –INSENSITIVITY ZONE AND PENALTY COEFFICIENT ON DEVIATION OF INTERNAL COORDINATES AND ZMP

No.	ε–insensitivity zone	Penalty coefficient	Maximum deviation of internal coordinates [rad]	$\lvert ZMP_{\max} - ZMP_{\min} \rvert$ [m]
1	0.01	1	0.3175	0.0838
2	0	1	0.3162	0.0843
3	0.01	10	0.2934	0.1137
4	0	10	0.2950	0.1156
5	0.01	100	0.2588	0.1336
6	0	100	0.2609	0.1504
7	0.01	1000	0.2390	0.2255
8	0	1000	0.2425	0.2885

the required form of movement. Consequently, the ZMP trajectory also completely changed as compared to the previous cases.

Thus, the selection of ε-insensitivity zone requires a careful consideration. A wider ε results in smoother motion approximations with lower accelerations at the joints. However, ε must not be too large because the distortion of the desired form of movement can occur.

4. *REACHING TASK*

The second example is the reaching task. If we consider the way how the reaching task is realized by humans we can notice several characteristics [9, 10]. As first, the human's hand moves approximately linearly provided there are no obstacles in the way. Also, in the beginning of the movement, the hand performs a sudden fast motion, to slow down significantly afterwards. However, the main difference in the realization of the reaching task between the robot and the human is in the dynamics of the motion, since the human's arm motion is very agile, whereas the robot's arm moves, as a rule, significantly slower.

Since we also would like the robot's hand move fast in the beginning, when it is still far from the object, and slow down when it comes close to it, the robot's hand motion is modeled as a mass *m* connected to the target point by a spring and a damper (Fig. 7).

If it is supposed that the position of the target point is fixed, the motion of the system can be described by the following second-order differential equation:

$$m\ddot{\mathbf{r}} = -K(\mathbf{r}-\mathbf{r}_{\mathbf{d}}) - C\dot{\mathbf{r}} \ . \qquad (5)$$

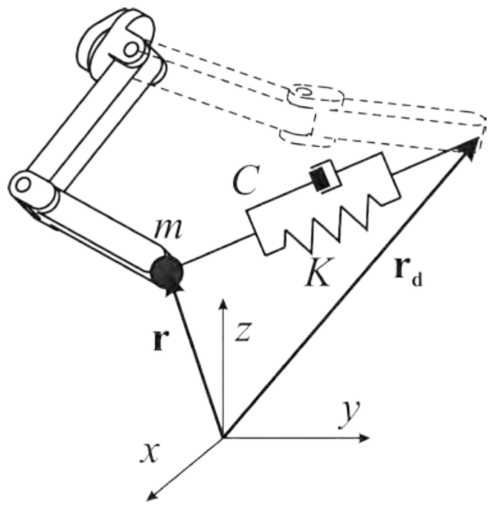


Fig. 7 Robot's hand having three DOFs

Now we will describe the procedure of the SVM training for the task of the robot's arm reaching the object. The robots' arm simulations were carried out on the same humanoid robot model described in section 3. Finally, the verification of the SVM control for the robot's arm (and consequently hand) motion is realized by assigning random points in the workspace to be reached by the robot's hand and comparing it with the system's motion shown in Fig. 7.

The motion of a multibody system can be described by the following differential equation:

$$\mathbf{H}\ddot{\mathbf{q}} + \mathbf{h}_{\mathbf{0}} = \boldsymbol{\tau} \ , \qquad (6)$$

where **H** is the inertia matrix of the system; $\mathbf{h}_0$ is the vector that includes the velocity and gravitation effects; $\ddot{\mathbf{q}}$ is the vector of generalized (internal) accelerations, and $\boldsymbol{\tau}$ is the vector of driving torques.

Based on the relations of direct kinematics, it is possible to find the relationship between the robot's tip and internal coordinates:

$$\mathbf{r} = \mathrm{f}(\mathbf{q}) \ . \qquad (7)$$

By differentiating equation (7) one obtains the relation between the generalized velocities of the robot's arm joints, as well as the relation between the generalized accelerations and linear acceleration of the robot's hand:

$$\dot{\mathbf{r}} = \mathbf{J}\dot{\mathbf{q}} \ , \qquad (8)$$

$$\ddot{\mathbf{r}} = \mathbf{J}\ddot{\mathbf{q}} + \mathbf{a} \ , \qquad (9)$$

where the matrix $\mathbf{J} = \partial\mathbf{r}/\partial\mathbf{q}$ represents the system's Jacobian and the vector $\mathbf{a} = (d\mathbf{J}/dt)\dot{\mathbf{q}}$ is the adjoint vector – column. The combination of equations (5)-(9) gives an expression that allows calculation of the corresponding driving torques at the joints:

$$\boldsymbol{\tau} = -\mathbf{H}\left(m\mathbf{J}\right)^{-1}\left(K\left(\mathbf{r}-\mathbf{r}_{\mathbf{d}}\right) + C\mathbf{J}\dot{\mathbf{q}} + m\mathbf{a}\right) + \mathbf{h}_{\mathbf{0}} \ . \qquad (10)$$

Thus, it is important to point out that the driving torques can be calculated only if **J** is a square matrix (i.e. when the system has no redundancy), which is fulfilled in the case considered. However, if the system is redundant, the Jacobi matrix is not square, so the inverse matrix $\mathbf{J}^{-1}$ does not exist. Such case is not considered here, but it is necessary to remind that the addition of certain conditions can allow the determination of a pseudo-inverse Jacobian that could be used in (10).

As is evident from (10), the driving torques $\boldsymbol{\tau}$ depend on the generalized coordinates $\mathbf{q}$, velocities $\dot{\mathbf{q}}$, and distance between the current and target positions, $\mathbf{r}-\mathbf{r}_d$. If we want to change the movement duration (the rate of hand motion), it suffices to change only the coefficient C. Our aim is to use the presented model to form an appropriate training set. Hence, we will adopt the angles $\mathbf{q}$ and angular velocities $\dot{\mathbf{q}}$ at all arm joints, position vector of the hand from the target position ($\mathbf{r}-\mathbf{r}_d$), as well as the damping C as input, and the driving torques at the joints corresponding to such motion as the output. The procedure of determining the input and output quantities is as follows:

1. An arbitrary point in the robot's workspace is selected as the starting position of the robot's hand.
2. For the selected point, the inverse kinematics problem is solved.
3. An arbitrary point is then selected in the robot's workspace to serve as the desired (target) position.
4. The damping coefficient is also arbitrarily selected from the predefined range.
5. The system's motion is simulated, and the driving torques in each iteration are calculated from (10).
6. When the distance between the current and target positions is sufficiently small (the hand is close enough to allow grasping), the simulation is stopped and the steps 3-6 are repeated. The last state serves as the starting one for the subsequent simulation.
7. After generating a sufficient number of movements, the procedure is stopped.

In this way, the hand's motion is simulated from the point at which the previous movement terminated to the next (arbitrarily chosen) target point. For each time instant, the values of all input and output quantities are obtained. The input vector thus formed for the training set $[\mathbf{q}^T \;\; \dot{\mathbf{q}}^T \;\; (\mathbf{r}-\mathbf{r}_d)^T \;\; C]^T$ is of dimension 10, whereas the dimension of the output vector $\boldsymbol{\tau}$ is 3.

Let us remind that the inputs to the SVM training set are the generalized coordinates $\mathbf{q}$, velocities $\dot{\mathbf{q}}$, the remaining distance to the target position $\mathbf{r}-\mathbf{r}_d$, and the damping coefficient C, by which the hand motion velocity is indirectly assigned, whereas the outputs are the joint driving torques $\boldsymbol{\tau}$. The hand's target position in the workspace is randomly assigned under the constraint that the x-coordinate of the given position is positive. In this way, the boundary of the workspace is determined by the hemisphere of a radius of 0.8 m.

In order to obtain an SVM with good generalization properties it is necessary to prepare a sufficiently dense training set, i.e. to generate a sufficient number of target positions. If the number of assigned target positions is small, the training set is sparse and the underfitting arises. On the other hand, if the number of assigned points is too large, it can yield overfitting, that is a bad generalization. For example, in this paper the overall number of target positions is ~1300, and the overall number of obtained input and output data pairs is ~770.000. The sampling period is 2.5 ms, which means that the hand's departure from the current position to the newly assigned position needs in average 592 iterations, i.e. 1.4 s. The value of the damping coefficient C is randomly chosen in the span from 200 to 400 Ns/m. All these data were obtained experimentally.

Of the possible 770.000 data, 2500 were randomly selected to train the SVM. It is evident from (10) that the relationship between the input and output quantities is nonlinear. Hence it is necessary to solve the nonlinear SVM regression using the kernel trick. The kernel function selected in this work is an RBF function, while the L norm for the empirical error is a function with the ε-insensitivity zone, where $\varepsilon = 10^{-15}$.

After completing the SVM training, the testing was performed by simulating the robot's arm motion. Fig. 8 shows the stick diagram of an upright humanoid robot whose task is to reach the target point by its right hand, the current and target positions being selected arbitrarily. To validate the control of the robot's hand motion using the trained SVM model it was necessary to assign the unseen input data. The simulation was carried out in the following way.

In the beginning, the starting position of the robot's hand is assigned in the world coordinate frame. Based on the relations of the inverse kinematics, the values of generalized coordinates q corresponding to this position were calculated. Since we assumed that the hand (and arm) moves from the rest, the values of generalized velocities in the beginning of the simulation are 0. Then, the target position in the workspace is

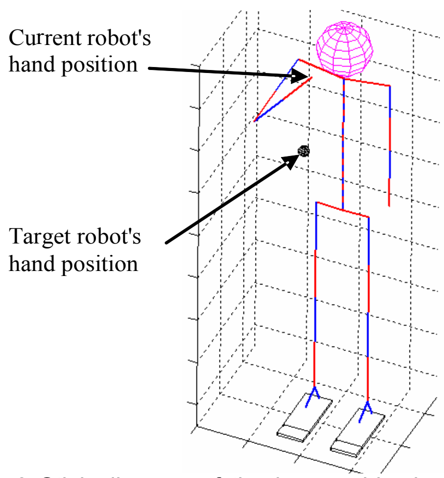


Fig 8 Stick diagram of the humanoid robot model and illustration of the assignment of the hand's target position

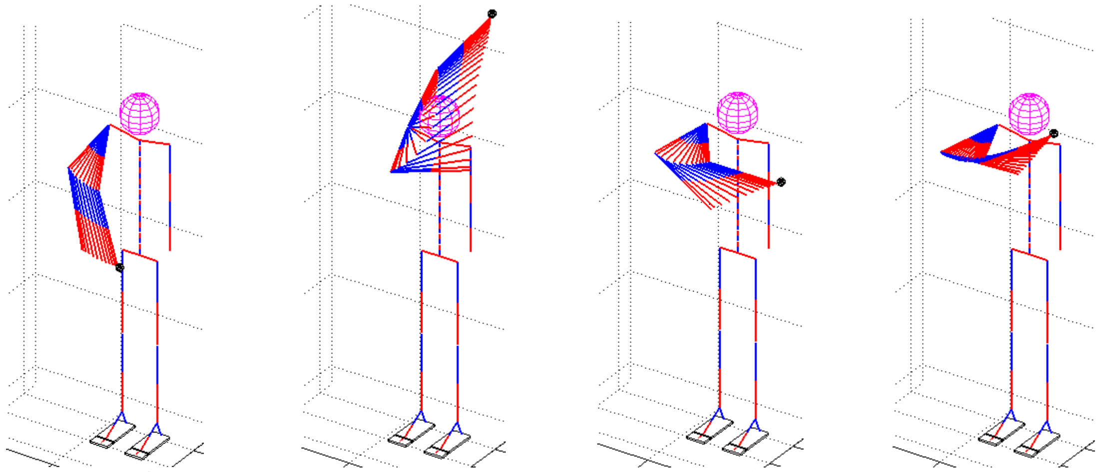


Fig. 9 Stick diagrams showing the hand's and arm's motion for the four different starting and target positions of the hand and for four different damping coefficients C. The distances between the starting and target positions and the corresponding damping coefficients are: a) $|\mathbf{r}-\mathbf{r}_d|$=0.198 m, C=231 Ns/m (); b) $|\mathbf{r}-\mathbf{r}_d|$=0.7518 m, C=282 Ns/m (case 2); c) $|\mathbf{r}-\mathbf{r}_d|$=0.3875 m, C=300 Ns/m (case 3): d) $|\mathbf{r}-\mathbf{r}_d|$=0.3113 m, C=363 Ns/m (case 4).

selected in a random way, as well as the damping coefficient C, which remains constant during the realization of one movement.

Based on these data, using the trained SVM, three driving torques at the hand joints were determined and then applied in the simulation of the motion. The action of the joint driving torques causes the hand motion and changes the values of the generalized coordinates, generalized velocities, as well as the distance to the target position. In each sampling period, using the information about the reached (the new current position of the hand) position, as well as the remaining distance to the target, the trained SVM determines new driving torques to be applied at the joints, to produce a further hand movement. The process is repeated until the hand reaches a distance of <2 cm from the target position. The hand then stops, assuming that the target has been reached, a new target point is randomly selected, and the process is continued.

Fig. 9 shows four different simulated movements that differ in respect of the damping coefficient, as well as in respect of the starting and target positions of the hand. It is evident that in all four cases the hand's trajectory almost coincides with the linear path, and that there are no overshoots on the way to the target position.

Now we shall analyze in detail the cases presented in Fig. 9. In the case 1, the distance between the hand's starting and target positions was $|\mathbf{r}-\mathbf{r}_d|$=0.198 m, and the damping coefficient was 231 Ns/m. The time needed for the hand to reach the position at a 2-cm distance from the target position was 0.555 s, and the average speed during this movement was 0.3568 m/s. In the case 2 where the distance $|\mathbf{r}-\mathbf{r}_d|$ = 0.7518 m and the damping coefficient was 282 Ns/m, the hand reached the target position in 1.3125 s, which means that the average speed of its motion was 0.5728 m/s. The average speed in the case 2 was higher since the path the hand had to traverse was by almost four times longer than in the case 1, so the hand had more time to accelerate.

The effect of the damping coefficient C on the velocity of hand motion is better seen from a comparison of the cases in which the distances $|\mathbf{r}-\mathbf{r}_d|$ were approximately the same. Because of that we will consider the cases 3 and 4, shown in Fig. 9. The distances between the starting and target positions in the two cases were 0.3875 m and 0.3113 m, respectively, and the corresponding damping coefficients were C=300 Ns/m and C=363 Ns/m. In case 3, the time needed for the hand to reach the distance of 2 cm was 0.72 s, and in the case 4 it was 1.0875 s, the average speeds being 0.5382 m/s and 0.2863 m/s respectively. It is evident that although the path in the case 3 was longer by 7 cm, the average speed was significantly higher than in the case 4.

We will now analyze the simulated movement which consists of successive reaching two target positions as shown in Fig. 10. In this example, the arm's hand in the first part of the movement is in the position 1, and the target position 2 is randomly assigned. The damping coefficient during the hand's motion to the position 2 is also randomly chosen, and it amounts to 250 Ns/m.

When the hand's distance from the target point becomes smaller than 2 cm, the first part of the movement is terminated. Then, a new target position 3 is randomly generated. In the course of the second part of motion, from the position of ending the previous movement, the hand moves towards the new target position 3. The damping coefficient during the second part of the movement was also randomly chosen to be 301 Ns/m.

Fig. 10 shows the analysis of the motion along the straight lines between the positions 1 and 2 and positions 2 and 3. It can be noticed that the hand's trajectory deviates from the expected straight-line path compared to the expected rectilinear path between 1 and 2. In the second part of the movement, the departure from the straight

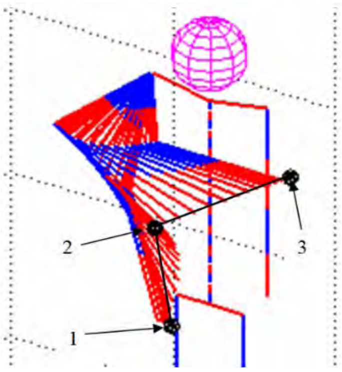


Fig. 10 Stick diagram of the humanoid robot model and illustration of the assignment of the hand's target position

line almost does not exist. The deviation from the straight-line path may arise as a consequence of the fact that the generated training set consists of a finite number of starting and target positions while its region that is close to the set limits is not sufficiently dense. In that case, when the assigned target position is not sufficiently close to a position from the SVM training set, the predicted values of driving torques can not be sufficiently well approximated, causing thus deviations.

However, it should be also noticed that the requirement for an exact tracking of a rectilinear path is not strict, and that neither human in reaching an object can strictly follow a straight-line path during the movement realization. It is much more important that in the human's performing reaching task has no overshoots, which has been fulfilled in all the presented examples of the simulated motion of the robotic hand.

5. Conclusion

The work considers the application of SVM in humanoid robotics on two essentially different examples: modification of a complex compensating movement recorded from a human to be applied to the humanoid robot and for generating arm movement to reach an object. The system's training to realize the reaching task (the online generation of the joint torques) from an arbitrary (instantaneous) position of the arm within the workspace was realized with the aid of SVM. It was demonstrated that the SVM is suitable for both applications.

References

[1] Vapnik V., “The nature of statistical learning theory, 2nd. ”, *Springer*, 2000.

[2] Kecman V., “Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models”, *The MIT Press*, 2001.

[3] Vukobratović M., Borovac B., Potkonjak V., Jovanović M., “Dynamic Balance of Humanoid Systems in Regular and Irregular Gaits: an Expanded Interpretation”, *World Scientific*, Int. Jour. of Humanoid Robotics, Vol. 6, No. 1, 2009, pp. 117-145.

[4] Vukobratović M., Borovac B., Raković M., Potkonjak V., Milinović M., “On Some Aspects of Humanoid Rrobots Gait Synthesis and Control at Small Disturbances”, *World Scientific,* Int. Jour. of Humanoid Robotics, Vol. 5. No. 1., 2008, pp 119-156.

[5] Vukobratović M., Herr H., Borovac B., Raković M., Popovic M., Hofmann A., Jovanović M., Potkonjak V. “Biological Principles of Control Selection for a Humanoid Robot’s Dynamic Balance”, *World Scientific,* Int. Jour. Of Humanoid Robotics, Vol. 5, No. 3, 2008, pp. 639-678.

[6] Vukobratović M., Borovac B., “Zero-Moment Point-Thirty Five Years of its Life”, *World Scientific*, Int. Jour. of Humanoid Robotics, Vol. 1, No.1, 2004, pp. 157-173.

[7] Vukobratović M., Borovac B., Potkonjak V., “Towards a Unified Understanding of Basic Notions and Terms in Humanoid Robotics”, *Cambridge University Press*, Robotica Vol. 25, 2007, pp. 87-101.

[8] Potkonjak V., Vukobratović M., Babković K., Borovac B., “General Model of Dynamics of Human and Humanoid Motion: Feasibility, Potentials and Verification”, *World Scientific,* Int. Jour. of Humanoid Robotics, Vol. 3, No. 2, 2006, pp. 21-4.

[9] Konczak J., Dichgans J., “The development toward stereotypic arm kinematics during reaching in the first 3 years of life”, *Springer,* Exp Brain Res, vol. 117, 1997, pp. 346–354,.

[10] Konczak J., Borutta M., Dichgans J., “The development of goal-directed reaching in infants, II. Learning to produce task-adequate patterns of joint torque”, *Springer,* Exp Brain Res, vol. 113, 1997, pp. 465–474.

[11] Mester, G., Rodic, A., "Autonomous Locomotion of Humanoid Robots in Presence of Mobile and Immobile Obstacles", Studies in Computational Intelligence, Towards Intelligent Engineering and Information Technology, Volume 243/2009, Springer, 2009, pp. 279-293.

[12] Mester, G., Rodic, A., "Contribution to the Simulation of Humanoid Kondo Robot", Annals of Faculty Engineering Hunedoara – International Journal Of Engineering, Tome IX, Fascicule 1, 2011, pp. 73-78.

[13] Mester, G., "Modelling of the Humanoid Robot Motion", Ipsi Journal, Transactions on Advanced Research, Volume 7, Number 1, 2011, pp. 21-25.

Branislav A. Borovac is professor of Robotics, at the Faculty of Tech. Sciences, University of Novi Sad, Serbia, and Head of the the Chair of Mechatronics, Robotics and Automatization. He is coauthor of two research monographs and a number of papers published in international journals and in proceedings of conferences and congresses. His research interest includes biped locomotion, humanoids, robot modeling and control.

Mirko M. Raković was born in 1982. He graduated from the Faculty of Technical Sciences (FTN), Novi Sad, Serbia. He is currently a Ph.D. student at the Chair of Mechatronics, Robotics and Automatization at the FTN. His research interest includes robotics, bipedal gait, mechatronics, computer electronics and programming.

Milutin N. Nikolić was born in 1984. He received Master degree in Mechatronics at the Faculty of Technical Sciences (FTN), Novi Sad, Serbia. He is currently a Ph.D. student at the Chair of Mechatronics, Robotics and Automatization at the FTN. His field of research includes robotics, bipedal gait, mechatronics and computer vision.

Fuzzy-Logic Sensor-Based Navigation of Autonomous Wheeled Mobile Robots in the Greenhouse Environments

Mester, Gyula

Abstract—*This paper presents the sensor-based fuzzy logic navigation of autonomous wheeled mobile robots in the greenhouse environments. The paper deals with the fuzzy control of autonomous mobile robot motion in unknown environment with obstacles and gives the wireless sensor-based remote control of autonomous mobile robot motion in greenhouse environments using the Sun SPOT technology. The simulation results show the effectiveness and the validity of the obstacle avoidance behavior in unknown environment and velocity control of a wheeled mobile robot motion of the proposed fuzzy control strategy. The proposed remote method has been implemented on the autonomous wheeled mobile robot Khepera that is equipped with sensors and the free range Spot from the Sun Spot technology. Finally, the effectiveness and efficiency of the proposed sensor-based remote control strategy are demonstrated by experimental studies and good experimental results of the obstacle avoidance behavior in the greenhouse environments.*

Key words — *autonomous mobile robot,* control, *Fuzzy Logic, greenhouse environments, obstacle avoidance.*

1. INTRODUCTION

CURRENTLY many researches in robotics are dealing with different problems of motion control of autonomous wheeled mobile robots in greenhouse environments. In the recent years, autonomous wheeled mobile robots have been required to navigate in more complex domains, where the environment is unknown. This paper deals with the fuzzy control of autonomous mobile robot motion in an unknown environment with obstacles and gives the wireless sensor-based remote control of autonomous mobile robots motion in the greenhouse environment using the Sun SPOT technology.

A greenhouse is defined as a house of glass, polycarbonate or fiberglass construction used for propagation, growth and care of plants. The function of a greenhouse is to create the optimal growing conditions for the full life of the plants [1].

Fuzzy logic approaches to mobile robot navigation and obstacle avoidance have been investigated by several researchers.

Many application works of fuzzy logic in the mobile robot field have given promising results.

Fuzzy reactive control of a mobile robot incorporating a real/virtual target switching strategy has been made in [2]. Navigation control of the robot is realized through fuzzy coordination of all the rules.

Real-time fuzzy reactive control is investigated for automatic navigation of an intelligent mobile robot in unknown and changing environments. The reactive rule base governing the robot behavior is synthesized corresponding to the various situations defined by instant mobile robot motion, environment and target information.

Paper [3] presents a strategy for autonomous navigation of field mobile robots on hazardous natural terrain using a fuzzy logic approach and a novel measure of terrain traversability. The navigation strategy is comprised of three simple, independent behaviors: seek-goal, traverse-terrain, and avoid obstacles.

The sensor-based navigation of a mobile robot in indoor environment is very well presented in [4]. The paper deals with the problem of the navigation of a mobile robot either in unknown indoor environment. Fuzzy controllers are created for the navigation of the mobile robot. The good results obtained illustrate the robustness of a fuzzy logic approach with regard to sensor imperfections.

Design, stability analysis and implementation of new intelligent fuzzy control systems for perception and navigation of nonholonomic autonomous mobile robots have been made in [5]. Reactive, planned and teleoperated techniques are considered.

Paper [6] proposed the use of a single side reflex for autonomous navigation of mobile robots in unknown environments. In this work, fuzzy logic based implementation of the single-sided reflex is considered. Simulation and experimental results are presented to show the effectiveness of the proposed strategy in typical obstacle situations.

Conventionally, mobile robots are equipped by ultrasonic sensors. It is supposed the autonomous mobile robot has groups of

Manuscript received June 15th, 2011. Gyula Mester is with the Robotics Laboratory, Institute of Informatics, Department of Technical Informatics, University of Szeged, Hungary (e-mail: gmester@inf.u-szeged.hu).

ultrasonic sensors to detect obstacles in the front, to the right and to the left of the vehicle. The model of the wheeled mobile robot has two driving wheels and the angular velocities of the two wheels are independently controlled.

The fuzzy control of a w heeled mobile robot motion in unknown environments with obstacles is proposed. Outputs of the fuzzy controller are the angular speed difference between the left and right wheels of the vehicle and the vehicle velocity.

Wireless sensor-based remote control of mobile robots motion in greenhouse environments using the Sun SPOT technology is proposed. The proposed method has been implemented on the mobile robot Khepera that is equipped with sensors and t he free range Spot from the Sun Spot technology. The autonomous mobile robot equipped with sensors is capable of driving to the end and back along crop rows inside the greenhouse.

Finally, the effectiveness and ef ficiency of the proposed sensor-based remote control strategy are demonstrated by experimental studies and good experimental results of the obstacle avoidance behavior in greenhouse environments.

The paper is organized as follows:

Section 1: Introduction. Model of the autonomous wheeled mobile robot is given in Section 2. In Section 3 environment perception is illustrated. In Section 4 s trategy of autonomous wheeled mobile robot motion control in unknown environments is proposed. In Section 5 w ireless sensor network (WSN) is illustrated. In Section 6 Sun-SPOT-based remote control of wheeled mobile robots in the greenhouse environment is proposed. Conclusions are given in Section 7.

2. *MODEL OF THE AUTONOMOUS WHEELED MOBILE ROBOT*

In this paper the model of the autonomous wheeled mobile robot has two driving wheels and the angular velocities: ω_l, ω_r of the two wheels are independently controlled (Fig. 1).

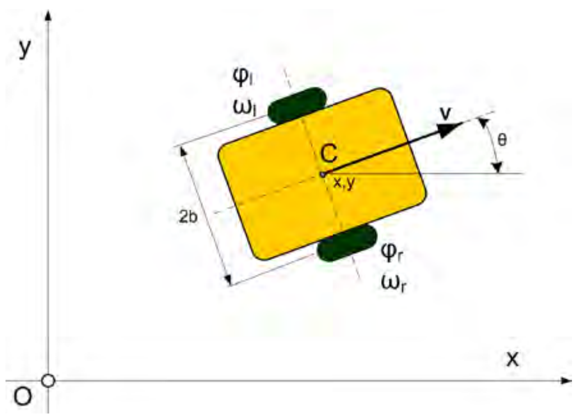


Figure 1. Model of the differentially driven mobile robot in the two-dimensional work space

The contact between the wheel of the mobile robots and t he horizontal plane supposes both the conditions of pure rolling and non-slipping during the motion. The velocity of the contact point between each wheel and the horizontal plane is equal to zero.

The rotation angle of the wheel about its horizontal axle is denoted by $\varphi(t)$ and the radius of the wheel by R. The position of the wheel is characterized by constants:

2b-distance between wheels, R-wheel radius and its motion by a time-varying angle:

$\varphi_r(t)$ – the rotation angle of right wheel and

$\varphi_l(t)$ – the rotation angle of left wheel.

The configuration of the mobile robot can be described by five generalized coordinates such as:

$$q = [x,\ \ y,\ \ \theta,\ \ \varphi_r,\ \ \varphi_l]^T \qquad (1)$$

Where: x and y are the two coordinates of the center of mass C – robot position, θ is the orientation angle of the mobile robot (robot orientation).

Kinematic model of the vehicle velocity v and the angular velocity of the mobile robot are given by the equation:

$$\begin{bmatrix} v \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} R/2 & R/2 \\ R/2b & -R/2b \end{bmatrix} \begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} \qquad (2)$$

3. *ENVIRONMENT PERCEPTION*

The autonomous wheeled mobile robot must be capable of sensing its environment. Every autonomous wheeled mobile robot needs some sensing devices to get first a per ception of its environment and t hen to move in this environment [5]. It is really important to have fast distance measurement from the mobile robot to the surrounding obstacles [6].

Conventionally, mobile robots are equipped by ultrasonic sensors. It is supposed the autonomous mobile robot has groups of ultrasonic sensors to detect obstacles in the front, to the right and to the left of the mobile robot. The imprecise perception of ultrasonic sensors is a result of the fact that these sensors provide a relatively accurate measurement of the distance to an object, bur poor information about its exact location due to the angular resolution.

Another source of uncertainty is a consequence of specular reflection and well-known problems such as cross-talking and noise. Several procedures have been dev eloped to overcome the disadvantages of ultrasonic sensors [5].

4. *STRATEGY OF AUTONOMOUS WHEELED MOBILE ROBOT MOTION CONTROL IN UNKNOWN ENVIRONMENTS*

When the autonomous mobile robot is moving towards the target and the sensors detect an obstacle, an avoiding strategy is necessary. While the mobile robot is moving it is important to compromise between avoiding the obstacles and moving towards the target position.

With obstacles present in the unknown environment, the mobile robot reacts based on both the sensed information of the obstacles and the relative position of the target [2].

In moving towards the target and avoiding obstacles, the mobile robot changes its orientation and velocity. When the obstacle in an unknown environment is very close, the mobile robot slows down and r apidly changes its orientation.

Fuzzy-logic-based control is applied to the navigation of the autonomous wheeled mobile robot in unknown environments with obstacles [7], [8], [9], [10], [11], [12]. The intelligent mobile robot reactive behavior is formulated in fuzzy rules.

Inputs to the fuzzy controller are:

- the obstacle distances p,
- the obstacle orientation θ_1,
- the target distances l,
- the target orientation θ_2 .

Outputs of the fuzzy controller are:

- the angular speed difference between the left and right wheels (wheel angular speed correction) of the vehicle: $\Delta\omega = \omega_r - \omega_l$ and the vehicle velocity V.

The obstacle orientation θ_1 and the target orientation θ_2 are determined by the obstacle/target position and the robot position in a world coordinate system. The block diagram of the fuzzy inference system is presented in Fig. 2.

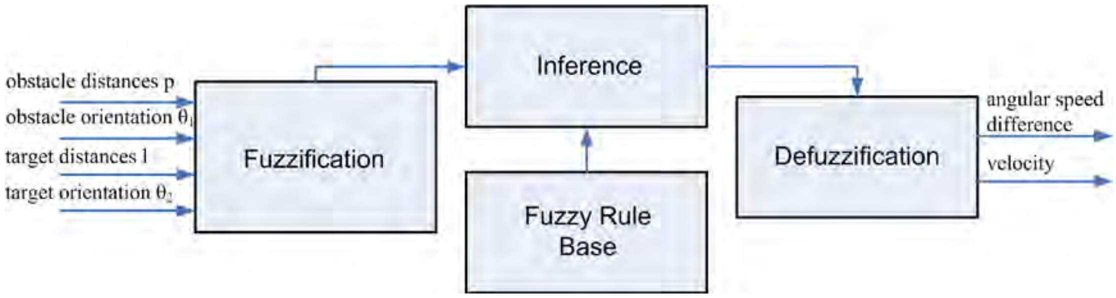


Figure 2. The block diagram of the fuzzy inference system

For the proposed fuzzy controller the input variables for the obstacle distances p are expressed using two linguistic labels: Gaussian membership functions near and far (p Є [0, 3 m]).

The input variables for the obstacle orientation θ_1 are expressed using two linguistic labels: Gaussian membership functions left and right (θ_1 Є [-π, π rad]). The input variables for the target distances l are simply expressed using two linguistic labels: Gaussian membership functions near and far (l Є [0, 3 m]). The input variables for the target orientation θ_2 are simply expressed using three linguistic labels:Gaussian membership functions left, target-direction and right (θ2Є [-3.14, 3.14 rad]). The fuzzy sets of the output variables the wheel angular speed correction Δω= ωr - ωl (turn-right, zero and turn-left) of the mobile robot are shown in Fig. 3.

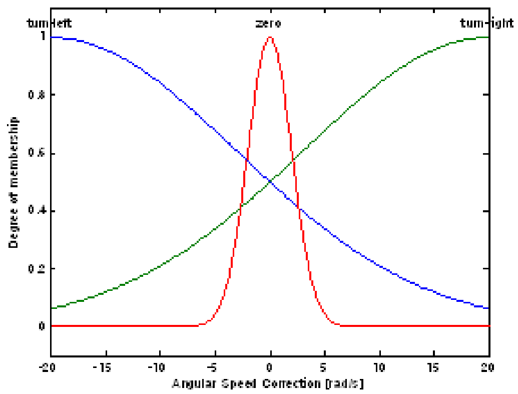


Figure 3. Membership functions of the angular speed difference Δω

The output variables are normalized between: Δω Є [-20, 20 rad/s].

The other output variable of the fuzzy controller is vehicle velocity: Velocity Є [-10, 20 m/s]. The fuzzy sets for the output variables Velocity (low, high) are shown in Fig. 4.

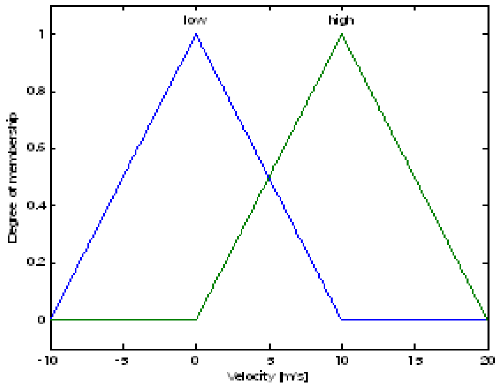


Figure 4. Membership functions of velocity of the autonomous mobile robot

The rule-base for mobile robot fuzzy control is:

- If θ_2 is right then Δω is turn-right
- If θ_2 is left then Δω is turn-left
- If p is near and l is far and θ_1 is left then Δω is turn-right
- If p is near and l is far and θ_1 is right then Δω is turn-left
- If θ_2 is targetdirection then Δω is zero
- If p is far and θ_2 is targetdirection then Δω is zero
- If p is near and l is far then velocity is low

- If p is far and l is far then velocity is high
- If p is far and l is near then velocity is low.

In the present implementation of the fuzzy controller the Center of Area method of defuzzification is used.

Simulation experiments are commonly used for the initial system analysis and c ontrol design while the experimental scalable tested system has to be used in the final phase of system evaluation and control verification.

The obtained results and c ontrol architecture can be afterwards adapted to the different application of mobile robots. Based on t his, the important task in system development is accurate and valuable modeling of the observed system.

Now, the author applied the proposed fuzzy controller to the mobile robot moving in an unknown environment with obstacles.

A simulation example of a wheeled mobile robot is presented in Fig. 6. Corresponding fuzzy control is implemented to perform tasks of obstacle and collision avoidance.

The results of the simulation are shown in Fig. 6. regarding the goal seeking and the obstacle avoidance mobile robot paths [9].

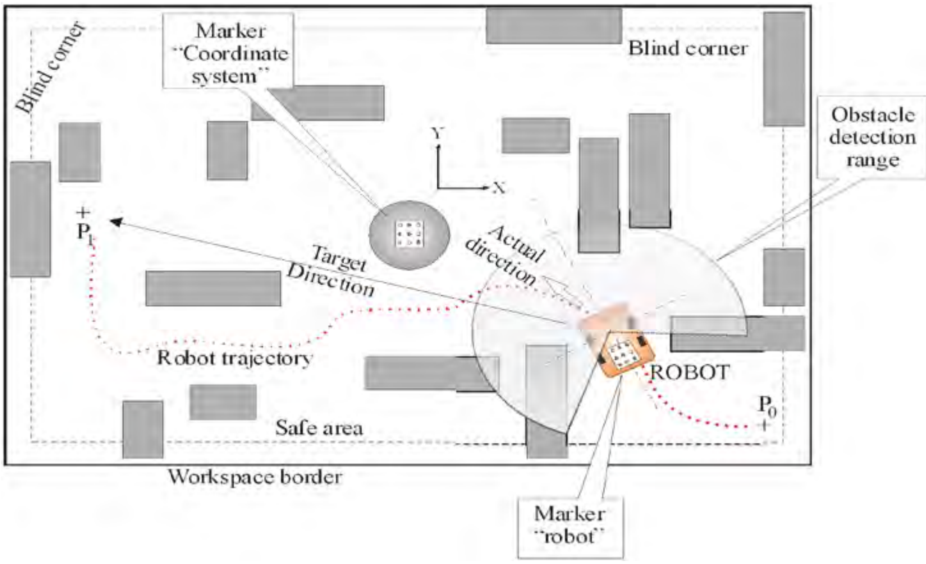


Figure 6. Obstacle avoidance trajectory of mobile robot

5. WIRELESS SENSOR NETWORK

A wireless sensor network (WSN) is a computer network consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations [13], [14], [15]. The development of wireless sensor networks was originally motivated by military applications such as battlefield surveillance.

Figure 7, presents the sensor node architecture. However, wireless sensor networks are now used in many civilian application areas, including environment and habitat monitoring, healthcare applications, home automation, and traffic control.

In addition to one or more sensors, each node in a sensor network is typically equipped with a radio transceiver or other wireless communications device, a s mall microcontroller, and an energy source, usually a battery.

Figure 8, shows the typical wireless sensor network. The size a single sensor node can vary from shoebox-sized nodes down to devices the size of grain of dust. The cost of sensor nodes is similarly variable, depending on t he size of the sensor network and t he complexity required of individual sensor nodes.

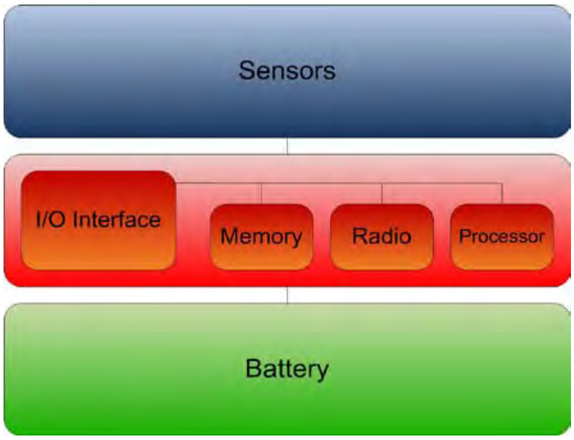


Figure 7. Sensor Node Architecture

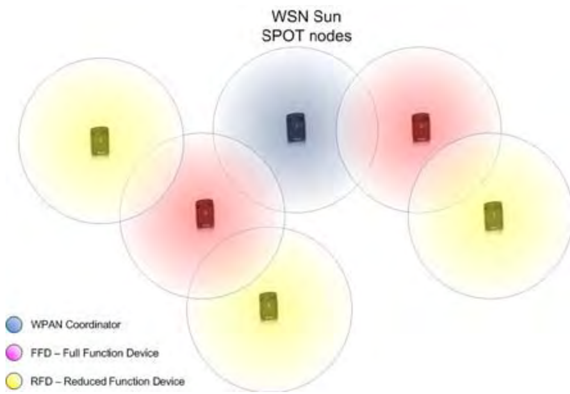


Figure 8. Typical wireless sensor network (WSN)

Wireless Robot-Sensor Networked systems (WR-SN) [9] refer to multiple robots operating together in coordination or cooperatively with:

sensors,

embedded computers, and

human users.

Cooperation entails more than one entity working toward a common goal while coordination implies a relationship between entities that ensures efficiency or harmony.

Communication between entities is fundamental to both cooperation and coordination and hence the central role of the networked system. Embedded computers and sensors are now ubiquitous in homes and factories, and increasingly wireless ad-hoc networks or plug-and-play wired networks are becoming commonplace.

Robots are functioning in environments while performing tasks that require them to coordinate with other robots, cooperate with humans, and act on information derived from multiple sensors. In many cases, these human users, robots and sensors are not collocated, and the coordination and communication happens through a network. Networked robots allow multiple robots and auxiliary entities to perform tasks that are well beyond the abilities of a single robot [9].

Robots can automatically couple to perform locomotion and manipulation tasks that either a single robot cannot perform, or would require a special-purpose larger robot to perform. They can also coordinate to perform search and reconnaissance tasks exploiting the efficiency that is inherent in parallelism. Further they can perform independent tasks that need to be coordinated.

Another advantage of networked robots is improved efficiency. Tasks like searching or mapping, in principle, are performed faster with an increase in the number of robots. A speed-up in manufacturing operations can be achieved by deploying multiple robots performing operations in parallel, but in a coordinated fashion. Perhaps the greatest advantage of using the network to connect robots is the ability to connect and harness physically-removed assets.

Mobile robots can react to information sensed by other mobile robots in the next room. Human users can use machines that are remotely located via the network.

The ability to network robots also enables fault-tolerance in design. If robots can in fact dynamically reconfigure themselves using the network, they are more tolerant to robot failures.

Finally, networked robots have the potential to provide great synergy by bringing together components with complementary benefits and making the whole greater than the sum of the parts.

6. *SUN SPOT BASED REMOTE CONTROL OF AUTONOMOUS WHEELED MOBILE ROBOTS IN THE GREENHOUSE ENVIRONMENT*

In this paper Sun SPOT-s (Small Programmable Object Technology) have been used to create remote control over a autonomous mobile robot [13], [14], [15], [16]. Typical applications of Sun SPOT-s include monitoring, tracking, and controlling.

The autonomous mobile robot equipped with sensors is capable of driving to the end and back along crop rows inside the greenhouse.

Sun SPOT is a small electronic device made by Sun Microsystems. The Sun SPOT is designed to be a flexible development platform, capable of hosting widely differing application modules. The Sun SPOT connection strategy is presented in [18], [19]. For this task 2 SunSPOT-s have been used from the development kit (Sun Microsystems, Inc. 2007).

Sun SPOTs are programmed in a Java programming language, with the Java VM run on the hardware itself. It has quite a powerful main processor running the Java VM “Squawk” and which serves as an IEEE 802.15.4 wireless network node.

The Sun SPOT base station has been used to read the data from the free range SPOT and send its contents to the PC.

Sun SPOTs wireless protocol is Zigbee-based protocol [13], [14], [15], [16], [17], [18], [19]. The PC with the Bluetooth connection sends the control signal to the autonomous mobile robot. Communication between the system components using SunSPOT is a very effective method.

In the Robotics Laboratory, Department of Informatics, University of Szeged it is possible to use the sensor-based remote control system [18], [19].

The user can start control experiment of mobile robots in Sun SPOT environment (Fig. 9, Fig.10), [17].

Figure 9. Experiment of remote control system

Figure 10. The greenhouse environments

7. CONCLUSION

The paper proposed the wireless sensor-based remote control of autonomous mobile robots motion in the greenhouse environments and a fuzzy reactive navigation strategy of collision-free motion and velocity control in unknown environments with obstacles.

Fuzzy-logic-based control is applied to the navigation of the autonomous wheeled mobile robot in unknown environments with obstacles. The intelligent mobile robot reactive behavior is formulated in fuzzy rules.

The proposed method has been implemented on the autonomous mobile robot Khepera.

The wireless robot-sensor networked systems are illustrated.

The simulation results show the effectiveness and the validity of the obstacle avoidance behavior in unknown environments and velocity control of a wheeled mobile robot motion of the proposed fuzzy control strategy.

Finally, the effectiveness and efficiency of the proposed sensor-based remote control strategy are demonstrated by experimental studies and good experimental results.

ACKNOWLEDGMENT

This research was partially supported by the TÁMOP-4.2.2/08/1/2008-0008 program of the Hungarian National Development Agency.

REFERENCES

[1] Badgery-Parker, J., Agnote DPI/249. Edition 1, 1999. pp. 1–2.
[2] W.L. Xu, S.K. Tso, Y.H. Fung: “Fuzzy reactive control of a mobile robot incorporating a real/ virtual target switching strategy”, Robotics and Autonomous Systems, Vol. 23, 1998, pp. 171-186.
[3] Homayoun Seraji, Ayanna Howard, Edward Tunstel, “Terrain-Based Navigation of Planetary Rovers: A Fuzzy Logic Approach”, Proceedings of the 6th International Symposium on Artifical Intelligence and Robotics & Automation in Space: i-SAIRAS 2001, Quebec, Canada, 2001, pp. 1-6.
[4] H. Maaref, C. Barret, ”Sensor Based Navigation of a Mobile Robot in an Indoor Environment”, Robotics and Autonomous Systems, Vol. 38, 2002, pp. 1–18.
[5] Federico Cuesta, Anibal Ollero, “Intelligent Mobile Robot Navigation”, ISBN 3-540-23956-1, Springer, 2006.
[6] Ranajit Chatterjee, Fumitoshi Matsuno, “Use of single side reflex for autonomous navigation of mobile robots in unknown environments”, Robotics and Autonomous Systems, Vol. 35, 2001, pp. 77–96.
[7] Gyula Mester, ”Intelligent Mobile Robot Controller Design”, Proceedings of the Intelligent Engineering Systems”, INES 2006, London, United Kingdom, 2006, pp. 282-286,.
[8] Gyula Mester, “Intelligent Mobil Robot Control in Unknown Environments”, Intelligent Engineering Systems and Computational Cybernetics, ISBN 978-1-4020-8677-9, DOI 10.1007/978-1-4020-8678-6_2, Springer, 2009, pp. 15-26.
[9] Aleksandar Rodic, Gyula Mester, Virtual WRSN – Modeling and Simulation of Wireless Robot-Sensor Networked Systems. Proceedings of the 8th IEEE International Symposium on Intelligent Systems and Informatics, SISY’10, pp. 115-120, ISBN: 978-1-4244-7395-3, Subotica, Serbia, 2010, pp. 115-120.
[10] Gyula Mester, “Sensor Based Control of Autonomous Wheeled Mobile Robots“, Ipsi Journal, TIR, Volume 6, Number 2, ISSN 1820-4503, 2010, pp. 29-34.
[11] Dragan Saletic, Uros Popovic, “Fuzzy expert system for automatic movement control of a platform on a ground with obstacles”, Proceedings of the YuINFO 2006, , Kopaonik, Serbia, 2006, pp. 1-6.
[12] J. Botzheim, Yuichiro Toda, and Naoyuki Kubota, “Path planning for mobile robots by bacterial memetic algorithm”. Proceedings of the 2011 IEEE Workshop on Robotic Intelligence in Informationally Structured Space, RIISS 2011, Symposium Series on Computational Intelligence, Paris, France, 2011, pp. 107–112.
[13] Matijevics Istvan, Simon János, "Control of the Greenhouse's Microclimatic Condition using Wireless Sensor Network" The IPSI BgD TIR, Vol. 6, Number 2, Belgrade, Serbia, 2010, pp. 35-38.
[14] Simon János, Matijevics István, "Implementation of Potential Field Method for Mobile Robot Navigation in Greenhouse Environment with WSN Support", Proceedings of the Conference, SISY 2010, Subotica, Serbia, 2010, pp. 319-323.
[15] Simon János, Goran Martinović, Matijevics István, "WSN Implementation in the Greenhouse Environment Using Mobile Measuring Station" International Journal of Electrical and Computer Engineering Systems, Osijek, Croatia, 2010, pp. 37-44.
[16] Sun™ Small Programmable Object Technology (Sun SPOT) Owner’s Manual Release 3.0, Sun Microsystems, Inc. 2007.
[17] http://www.inf.u-szeged.hu/robotics
[18] Tamás Szépe, “Sensor Based Control of an Autonomous Wheeled Mobile Robot“, Proceedings from PROSENSE 3rd Seminar Presentations, , Ljubljana, Slovenia, 2010, pp. 34-37.
[19] Tamás Szépe, “Robotirányítás támogatása távoli érzékelőrendszerrel“, VMTT Konferenciakiadvány,. Újvidék, Szerbia, 2010, pp. 527-532.
[20] Várkonyi-Kóczy, A.R., “A Universal Autonomous Robot Navigation Method,” International Journal of Advanced Computational Intelligence and Intelligent Informatics (JACIII), Vol. 12, No. 2, 2008, pp. 206-22.

Gyula Mester received his D. Sc.degree in Engineering from the University of Novi Sad in 1977. Currently, he is a Professor at the University of Szeged, Department of Informatics, Hungary. He is the author of 206 research papers. His professional activities include R/D in different fields of robotics engineering: Intelligent Mobile Robots, Humanoid Robotics, Sensor-Based Remote Control. He is an invited reviewer of more scientific journals and the author of several books. He is the coordinator of the Robotics Laboratory from the University of Szeged, in European Robotics Research Network. His CV has been published in the Marquis “Who’s Who in the World 1997”.

A Complexity Reduced Hybrid Autonomous Navigation Method for In-Door Robots

Annamária R. Várkonyi-Kóczy

Abstract—*Autonomous robot navigation is an important research field because these robots can solve problems where the human presence is impossible, dangerous, expensive, or uncomfortable. In this paper, a new hybrid autonomous navigation method is introduced. The algorithm is composed of visibility graph based global navigation and simple potential field based local navigation parts. It applies a new automated graph generation method which may become necessary if, because of the observed new obstacles, a new path should be generated. The quasi optimal route is found by applying the well known A* algorithm on the graph. The presented technique offers a quasi optimal universal navigation technique which can successfully be used in all, known, unknown, and dynamically changing environment.*

Index Terms— *mobile robots, indoor robots, autonomous navigation, local navigation, global navigation, hybrid navigation, vision-based obstacle detection, road map algorithms, potential field based guiding, A* algorithm, obstacle memory*

1. Introduction

NOWADAYS the topic of autonomous navigation has become popular among researchers all over the world (see e.g. [1]). This is because autonomous navigation can provide a solution to several problems which cannot be solved without an autonomous robot. There are situations where human interference is not an option and there is no other solution but applying autonomous robots to tackle these problems. Autonomous robots can advantageously be used in such cases as well where the human presence is unhealthy, monotonous, or very expensive. To mention some examples, there are situations that deal with problem solving which comprise even the carrying of the radioactive by-products of nuclear power plants or lifesaving in case of disasters by taking medicines to the injured if they cannot be approached only by means of little autonomous machines. On the other hand, examples can be enumerated, which concentrate on information gathering in order to support certain scientific activities (see e.g. [2]). However, the aforementioned are extreme cases, but problems can be found in all walks of life. Autonomous robots could substitute people where the task is too monotonic, such as the collection of litter in a building or looking out for them at night.

The navigation system of an autonomous robot is one of its components which is responsible for the decision where to move in the next moment. The word 'autonomous' stands for a concept, which considers that the robot does not communicate with the outside world under normal circumstances. This system requires pieces of information in order to reach its goal apart from what its current position is. The navigation system can have a priori, pre-supplied data about its environment, and on the other hand sensory data is the one which carries information on the momentary state of the immediate surrounding of the robot. To perform the navigation task suitably, the navigation system must be capable to use as much information about the environment as possible.

As far as the types of obstacles are concerned, there can be static and dynamic ones. The former refers to objects the positions of which are pre-supplied into the robot and the latter to those ones which are detected by means of sensors.

The alignment of navigation systems fall into three categories: global, local, and hybrid. In case of global navigation (see e.g. [3], [4]), only pre-supplied data are accessible while in case of local navigation (see e.g. [5], [6], [7]), only the sensory measurements are taken into account.

Hybrid method merges the advantages of both aforementioned systems. This concept which aims to combine the intelligence of global navigation algorithms with the reactive dynamic behavior of local ones was introduced by the author of this paper in 2003 (see [8] and also [9]). This is reached by introducing a hybrid navigation system consisting of two modules, one of which uses the a priori information and determines the main steps of the optimal route towards the goal, whereas the other carries out the navigation itself using a local approach (see Fig. 1).

In this paper, a new hybrid navigation algorithm is presented which can be used both indoors and outdoors. The main advantages of the method compared to previous techniques are: (1) It

Manuscript received October 20, 2010. This work was supported by the Hungarian Fund for Scientific Research (OTKA T T78576) and by the Structural Fund for Supporting Innovation in Knowledge and Technology Intensive Micro- and Spin-off Enterprises (GVOP-3.3.1.-05/1.2005-05-0160/3.0).

A.R. Várkonyi-Kóczy is with the Institute of Mechatronics and Vehicle Engineering, Óbuda University. (e-mail: varkonyi-koczy@uni-obuda.hu).

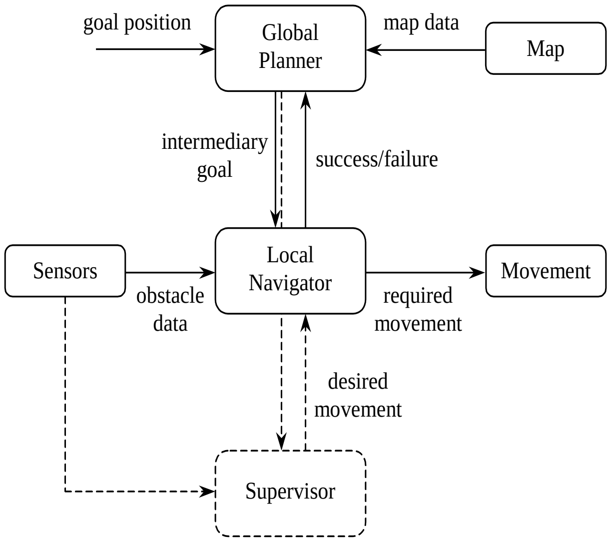


Figure 1. General block scheme of a hybrid navigation system. Under normal circumstances, the supervisor block (below, dashed line) is not present, but it can be used in a possible teaching phase.

ensures quasi optimality in the performance; (2) handles the local minima problem systematically; and (3) applies a new complexity optimized automated graph generation for the path finding in maps.

The paper is organized as follows: Section II briefly summarizes the main ideas of the most characteristic global, local, and hybrid navigation approaches. The new hybrid navigation system and path generation method are introduced in Section III. Section IV shows simple examples to illustrate the effectiveness of the method and, finally, Section V gives the conclusions.

2. Local, Global, and Hybrid Navigation

Navigation algorithms can be categorized based on the type of information available for the robot during the navigation.

Global algorithms use a priori information available about the whole area of the robot's operation and usually apply some kind of trajectory planning method that is based on previously existing optimization algorithms, thus are able to find an optimal trajectory according to various optimality criteria. The information is typically given in the form of a map containing static obstacles.

Local algorithms use only sensory information about the robot's immediate surrounding. This can be given in many forms, e.g. values representing readings from distance sensors or 3D models built of camera-images. Local methods respond well to dynamic challenges, thus they may be used in unknown environment however the optimality of the trajectory can not be guaranteed.

Hybrid techniques apply both global and local elements.

2.1 Global navigation

As these algorithms use information about the whole area, they usually have to handle a large amount of data and are complex. This asks for doing as many calculations offline as possible. This suits the supplied information, since the robot's sensors can only provide up-to-date information about the robot's immediate surrounding.

Roadmap algorithms: Global navigation algorithms can easily be realized by first constructing a graph from the supplied map and then searching for the shortest path on this graph [10], [4]. The problem

$$\{M, p_{start}, p_{end}\} \mapsto P \tag{1}$$

of global navigation can be divided into two subproblems

$$M \mapsto G \tag{2}$$

and

$$\{G, p_{start}, p_{end}\} \mapsto P \tag{3}$$

where M, G, and P are the sets of maps, graphs, and paths, respectively, while p_{start} and p_{end} are the start and end points. The nodes of the graph correspond to free points in real space and edges to paths between these points, which the robot can safely go along. The graph generation can be done as a precalculation step if the map only contains static obstacles. Searching for the shortest path can be done using the A* [11], D* [12], or other well-known algorithms. For the graph generation, several different solutions exist, each having its own advantages and disadvantages [11].

One of the biggest drawbacks of roadmap algorithms is their low performance in dynamic environments, as current approaches simply replan from scratch when changes are observed. This can be improved by continuously replanning in an anytime fashion as more and more information becomes available. The method can be further enhanced by extrapolating obstacle trajectories based on previous motion and adding a time dimension to the search space [13].

Visibility graphs: Visibility graph is a graph generation method for polygonal obstacles [14], [4]. Each node corresponds to a vertex of an obstacle and two nodes are connected if the segment joining them does not intersect any obstacles (i.e. they are visible from each other, hence the name). If we add a start and end node to the graph corresponding to the start and end points in real space, then the shortest path between the end and start node on the graph ensures the shortest path in real space (hence the other name).

Consider the set of possible maps *M* to comprise the sets of obstacles which are represented as polygons. Let *P* be the set of polygons from which topological maps are built up. The procedure to generate a graph by these pieces of information is to extrude the vertices of each polygon one by one to a desired distance from the original positions. By this step we ensure that the resultant graph of the generation will decrease the risk of a possible bump of the robot with obstacle. In this way the agent will meet the extruded vertices while progressing to the goal. Let *V'* be the set that includes these new vertices, by means of which the $h: V' \mapsto G$ can be defined. This is considered to be the graph generator function.

Automated graph generation: Let *M* be the set of possible maps and *G* be the set of graphs. We are looking for a function $f: M \mapsto G$, so that for every $m \in M$ map the shortest path *p* on the generated $g \in G$ graph is not significantly longer than the possible shortest path in real world space and, furthermore, *f* can be fully automated.

In case of local minima the solution is the regeneration of the graph by means of the $f^*: M' \mapsto G$ function, where $M' = M \cup M_o$ and M_o denotes the set of observed obstacles. Thus, *M'* contains both the static elements (all of them are part of set M) and the observed obstacles (M_o) that were supplied by the external sensors of the robot.

2.2 Local navigation

Since these algorithm use only information of the robot's surrounding, they usually do not have to handle as much data as global algorithms. This allows them to do all the calculations online, which again suits the information as the robot's immediate surrounding is usually known precisely at the very moment.

Potential field based navigation: One of the possible local navigation systems is the so called Potential Field Based (PFB) guiding [5] which is our preferred one, as well. The core idea of this concept is that the environment affects with repulsive forces to the agent according to the following expression:

$$y = -\sum p(d_i) \cdot e_i \qquad (4)$$

where d_i is the distance between the *i*th obstacle and the robot, $\mathbf{e}_i$ is the unit vector pointing towards the *i*th obstacle from the robot, and *p* is a potential function. This potential function can be based on Coulomb's law or some kind of variation of it or trained using neural networks. Vector **y** is the obstacle avoidance vector. In some cases *p* can have negative values, which means attractive force. We are going to use this feature in case of obstacle avoidance. We are going to use this feature in case of the goal.

The main advantage of PFB guiding lies in its simplicity. The algorithm needs very little processing power and can easily be trained. Furthermore, the inputs of a PFB system can be directly mapped to distance sensors of a robot. Nevertheless, a major disadvantage of this approach is that the robot can get trapped in local minima other than the goal. This problem can be dealt with, but this usually makes the algorithm much more complex, losing its main advantage [15].

The Vector Field Based (VFB) navigation [16] is an improved version of the PFB guiding. The main differences are that the algorithm takes into account the repulsive vectors themselves instead of the sum of the vectors as in case of PFB and the direction of the repulsive force of an obstacle can be other then parallel with the direction in which the obstacle is detected depending on it in a more complex way. This allows us to teach the robot different kinds of navigation styles according to our needs.

Bug algorithm family: There exists a family of algorithms which only use local knowledge, but are complete in the sense that they always find a safe path [17]. This family consists of Bug 1, Bug 2, and a number of their derivatives. Their name originates from the similarity between the robot model used by these algorithms and an insect. The robot has two simple behaviors: follow a wall (left or right) or move towards the goal in a straight line.

Bug 1 algorithm can be described with a set of simple rules: 1, head toward the goal by default; 2, if an obstacle is reached, then circumnavigate it and remember the point that is closest to the goal; 3, return to that point and continue. It is trivial that this algorithm is complete (considering that the number of obstacles is finite), since the robot gets closer to the goal with each obstacle circumnavigated and every obstacle is circumnavigated at most once. Bug 2 and their derivates work basically the same.

At first glance these algorithms seem to be a perfect solution as they are complete and only use local knowledge which is always available. However, there are certain drawbacks, which make them inferior to hybrid navigation systems in many ways. First of all, realizing the wall following behavior using only distance sensors is not an easy task. There are solutions which address these problems, but they can not eliminate it completely. But their biggest drawback lies in the fact that for typical real world situations, the path given by Bug algorithms is far from optimal. The upper bound for the length of the path is proportional to the sum of every obstacles' perimeter, which makes bug algorithms unsuitable for many applications.

2.3 Hybrid navigation

In recent years, attention has been directed to hybrid navigation [8], [18], [9], [19]. These systems combine the advantages of local and global methods, while their complexities and running times are kept as low as possible.

3. HYRON, THE NEW HYBRID NAVIGATION SYSTEM

The new hybrid navigation system 'HYRON' aims to incorporate the advantages of hybrid systems and also to improve the effectiveness by applying well-known and quick local navigation (PFB) and optimal global planning (visibility graph, A*) parts. Furthermore, a new graph generation method is also introduced which ensures the quasi optimality of the generated path.

A hybrid navigation system comprises a local and a global navigation system and accessory functionalities that help the previous two to work together. The global and local parts can be any kind of global and local algorithms. The accessory functionalities have to include an algorithm for detecting if the local system has encountered a situation it can not solve. Other functionalities can be added to improve the system, but this depends on the global and local parts chosen.

Fig. 2 shows how the new hybrid navigation system is built up. The global part is responsible for designing a quasi optimal (optimal if only pre-known, static obstacles are present) path consisting of temporary goal-points and leading to the final goal and for the avoidance of static obstacles, while the local part makes the robot move to the next (temporary) goal. However, it is not enough to purely merge the two subsystems, because their functionality has to be enhanced.

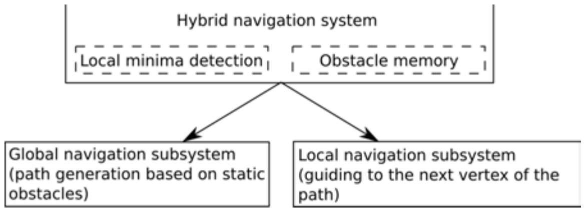


Figure 2. Block diagram of the new hybrid navigation system

This improvement includes the local minima detection and the obstacle memory itself. The new concept of obstacle memory covers dealing with the temporary storage of dynamic obstacles that have just been recognized. By merging the content of this memory and the set of static obstacles we get a wider look of the environment of the robot. After that the visibility graph algorithm must be applied on this union and a new path is searched on it to reach the goal. Of course the capacity of this temporary storage is limited, so only recent obstacles are taken into account. The recent expression refers to a specific surrounding of the agent, as if a square (or circle) shaped net were moving simultaneously with the robot which is positioned at its center. If an obstacle is nearby then it supposedly intersects with one or more segments of this grid and if so, it is considered to be a dynamic obstacle and stored in this temporary memory. (Note that this grid is solved by use of sensors that are constructed to the robot.) It is needless to say that if there is no outer interference, the agent keeps trying to follow the selected path as much as it can.

Fig. 3 demonstrates the data and control flow diagram of the new system. As it is visible, the graph generation requires information from the sensors, whether a new obstacle and/or local minimum is encountered and also the map, which contains the known obstacles, that is supplied previously to the agent. If all the required pieces of information are ensured, the graph generation is performed by the concept of visibility graph. After that only a path finding algorithm, such as the A* or D* ([11]) has to be applied on it in order to get the optimal route to the goal. Last but not least, the local navigation is going to use this path and the data from the sensors to span the distance between to nodes of the generated graph.

Consider the previously given graph generator

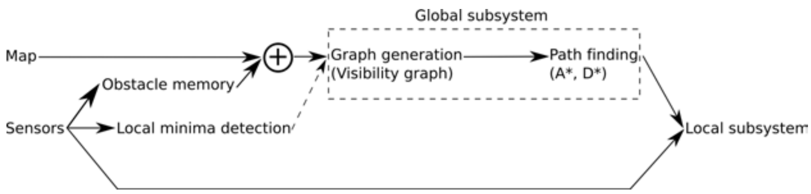


Figure 3. Data and control flow diagram of the HYRON system

function $h : V' \mapsto G$. The output of this function is a graph that is generated by the visibility graph algorithm. After we have a graph on which an optimal path finding is easy to be implemented, the navigation system is going to use it, if there is no unexpected obstacle en route. In this case, we can simply rely on our global plan and touching the corresponding vertices as progressing to the goal on the edges. The navigation itself is realized by the local part between two vertices. It has simple reasons: we have to take the strict fact into consideration that the environment is diverse, where the surrounding of the agent can change in every moment. In details it may mean that people are going to roam all around in a building or we can recognize new (static or dynamic) obstacles. It is evident that these situations can be handled only by local navigation. In the following, local and global navigation, and graph generation are discussed in details.

3.1 Global navigation

The global navigation system has to give a path for any start and end point pair on a map it has previously learned (if the map exists). A path is defined as a sequence of points. The start and end points can be any points on the map that are not occupied by an obstacle. The fact that this system can learn the map before any task is given to it, carries high importance to its performance, since this allows it to make calculations which are not feasible during the actual navigation. For most roadmap algorithms this means that the graph can be generated without time constraints, allowing us to use more complex algorithms.

Let us recall that M and G are the sets of maps and graphs respectively while p_{start} and p_{end} are the start and end points, p_i is the ith point of the path, $i = 1...n$, and n is the number of points on the path. Using offline calculations the

$$g : \{M, p_{start}, p_{end}\} \mapsto (p_1, p_2, ..., p_n) \qquad (5)$$

global navigation system can be divided into

$$g_{offline} : M \mapsto G \qquad (6)$$

and

$$g_{online} : \{G, p_{start}, p_{end}\} \mapsto (p_1, p_2, ..., p_n) \qquad (7)$$

parts where $g_{offline}$ is a more complex function which can not be executed real-time, while g_{online} is a simpler one and can be executed real time.

For the global navigation system we can make the following requirements:

- It has to be complete. That is, for every possible map and start and end point pair, it has to be able to give a path that the robot can safely go along.
- It has to give such a sequence of points, so that between any point and the next one a local navigation system would be able to navigate through. For this requirement we defined that for any p_i and p_{i+1} segment p_ip_{i+1} should not intersect any obstacles.
- It has to give a path that is near optimal.

Many roadmap algorithms fulfill the above requirements; however, not all of them give a good quality solution. We have compared different graph generating algorithms based on their performance during modifying a graph with newly observed obstacles and on the optimality of the given path and found that the visibility graph method has adequate performance while giving an optimal path.

3.2 Local navigation

Between the points of the path given by the global algorithm, a local navigation system has to guide the robot. Since we defined that the segment $p_i p_{i+1}$ can not intersect any obstacles, in a static environment the simplest algorithm, which guides the robot to the next point on a straight line, could be used. Unfortunately, we can not expect the environment to be static, so we need more complex solutions.

As a consequence of the fact that the environment is to be changing as time goes on, we have to take certain things into consideration. Firstly, different creatures (people, animals, robots) can pass the territory where the robot moves on. Secondly, the surrounding of the robot can vary on a wide scale, e.g. a door can be opened in one moment, and closed in the other. Given these facts and the desire that the algorithm be fast enough were an inspiration to implement the PFB local navigation algorithm. Due to its properties, it can handle easily the quick changes of the environment however it is our task to enhance it.

As we know, the potential field based guiding uses the resultant of the vectors of repulsive powers and the one directing toward the goal. In our case these conditions are given but we enhanced it in order to satisfy our expectations.

The robot is about to follow the path that was designed by the global navigation system, while it is feasible and there are no unexpected obstacles en route to the goal. Given the segment $p_i p_{i+1}$, which intersects one or more $o \in O$ (member of the set of obstacles). In this case our agent can not pass through these obstacles, so there is no option but switching to the local navigation system, while it reaches the end vertex of the segment. Nevertheless, this subsystem knows nothing about the global goal and if so, the agent could not avoid the obstacle. That is the reason why a temporary target must be chosen to the local subsystem which attracts the agent so the PFB navigation will work fine. The best solution is to choose the end vertex of the segment $p_i p_{i+1}$.

3.3 Graph regeneration

As the local navigation of the new hybrid system is not a complete one, the robot can get trapped in a local minimum. In this case the global navigation system has to merge the contents of the obstacle memory and the set of static obstacles, regenerate the graph, and find a new path on it. The components of the hybrid system we have discussed so far can all be run in real time, even on devices of low performance (e.g. microcontrollers), however, the regeneration of the map can take much time for the visibility roadmap algorithm.

The most trivial improvement is to only modify the graph and not start from scratch. This can be done as the following:

1) Find all original edges that intersect any new obstacle and remove these edges.
2) For all vertices of new obstacles, connect them with all new or old vertices that are visible from them.

Step one guarantees that every possible path on the graph is also possible path in real space. Step two guarantees, that every optimal path in real space is also a possible optimal path on the graph. Unfortunately, this method is still too complex to be used in real-time because the running time is proportional to the square of the number of obstacles.

Our solution is to generate a graph that is simpler than the one defined by the visibility algorithm. This can be done by connecting new vertices only to vertices that are nearer to them than a certain distance. Of course the generated graph is not necessarily connected. In this case, the distance has to be increased and the connecting of new vertices has to be redone. The whole algorithm is the following:

1) Find all original edges that intersect any new obstacle and remove these edges.
2) For all vertices of new obstacles, connect them with all new or old vertices that are visible from them and closer than a certain distance.
3) If the graph is not connected, then increase the distance and continue with step 2.

So far we have not dealt with the complexity of finding intersecting obstacles for edges. This can be implemented using binary space partitioning techniques or hash maps. Our algorithm uses binary space partitioning.

Binary space partitioning (BSP) techniques [20] recursively subdivide a space into convex sets using hyperplanes. The resulting tree representation of space, called a BSP tree, was first used to accelerate polygon rendering, but can also be applied to collision detection. If the tree is balanced, then the time needed for finding a number of sets in the tree is proportional to the logarithm of the number of elements in the tree. In our case, the tree is generated for the whole map and we store in each leaf the obstacles that the convex set contains. Then, we use this tree to find areas that a certain edge intersects. As the tree can be built so, that each leaf contains at maximum a constant number of obstacles, we can find obstacles that an edge intersects in time proportional to the logarithm of the number of obstacles.

The BSP tree can be built using the following steps:

1) Sort all obstacles on the map using one of their coordinates.
2) Divide the obstacles into two groups, based on whether their coordinate is greater or less than a constant. The constant should be chosen to make the two groups equal in size.
3) If the resulting groups contain more obstacles than a constant parameter, then continue recursively on them, using another coordinate.

The above steps build a balanced BSP tree, where each leaf holds at maximum a constant number of obstacles. This can be done offline for the map. New obstacles can be easily added online, by finding the leaf that should contain it, in time proportional to the logarithm of the number of obstacles, and adding the obstacle to this leaf. Then, finding intersection obstacles for an edge can also be done in running time proportional to the logarithm of the number of obstacles. Testing the connectivity of the graph is trivial.

The resulting graph is not an optimal one, but is usually close to optimal, thus the path found on it can be accepted as quasi optimal. The running time of the graph regeneration is roughly proportional to the number of new obstacles and the logarithm of the size of the map.

4. Examples

For illustrating the performance of the introduced navigation technique, in this section simple examples are shown.

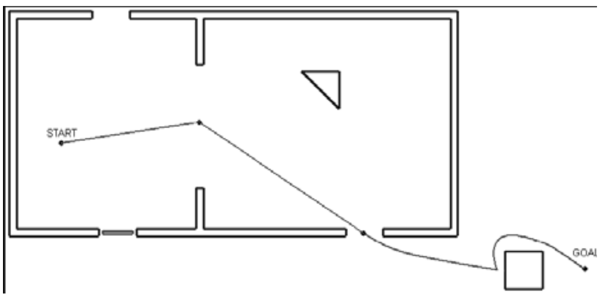


Figure 4. Path generated by the hybrid method presented in [8]

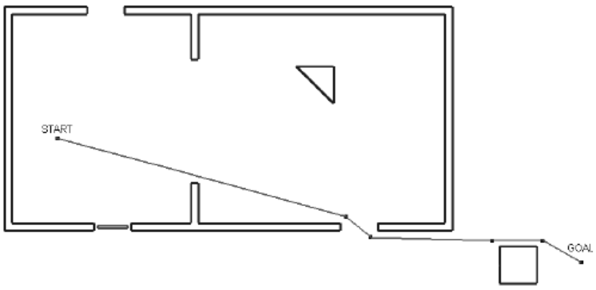


Figure 5. Path generated by HYRON

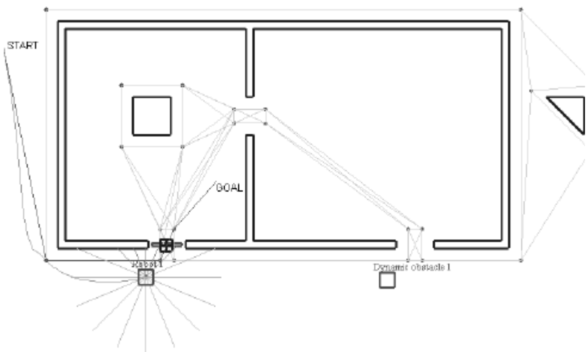


Figure 6. Graph regeneration, part 1

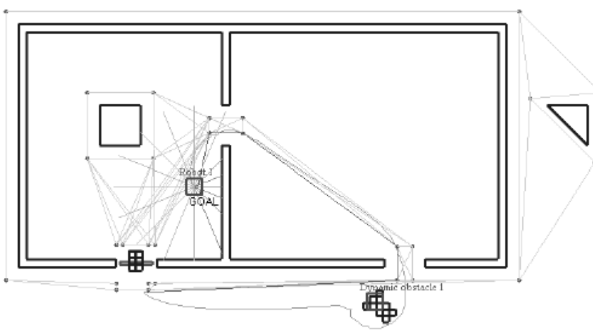


Figure 7. Graph regeneration, part 2

The first example makes a comparison between the hybrid algorithm described in [8] and the new method in case of a simple room arrangement. Fig. 4 shows the path generated by the method presented in [8]. (The route is far from optimal and usually not all static obstacles are considered.) Fig. 5 shows the near optimal path of the new hybrid navigation.

Fig. 6 demonstrates the case when the agent does not know whether a door is closed or not. In this case, the visibility path graph is generated by not taking the door into consideration (it can be opened) and calculating the path through it. However, when the robot concludes that it is closed it cannot move ahead. This is one case of local minima, because there is no progression to the goal. By generating a new graph, which includes the door as an obstacle, the problem is solved and another route can be taken, however, it is more expensive compared to the original one. This situation is represented by Fig. 7. This figure shows also the path of the agent to the goal, including the avoidance of an unexpected obstacle: Dynamic Obstacle 1.

5. CONCLUSION

In this paper, a new hybrid robot navigation system has been reported. The method combines the so called potential field based guiding (PFB), which is its local part, a roadmap algorithm which is its global part, and a new concept, the graph regeneration based on the obstacle memory. Simulations proved that it can successfully be applied in those situations too where obstacle and local minimal avoidance is the case. As a consequence, we offer a new approach by means of which many unexpected conditions can be tackled. The new method is universal and can successfully be applied in real situations in all walks of life.

REFERENCES

[1] Mester, Gy., Rodic, A., “Autonomous Locomotion of Humanoid Robots in Presence of Mobile and Immobile Obstacles,” Studies in Computational Intelligence, Towards Intelligent Engineering and Information Technology, Vol. 243/2009, pp. 279-293.

[2] Mars Pathfinder, http://marsprogram.jpl.nasa.gov/MPF/, 1997.

[3] Blåsvær, H., Pirjanian, P., and Christensen, H.I., “AMOR - An Autonomous Mobile Robot Navigation System,” Proc. of the IEEE Int. Conference on Systems, Man, and Cybernetics, Vol. 3, 1994, pp. 2266-2271.

[4] Latombe, J., “Robot Motion Planning,” *Kluwer Academic Publishers*, Boston, MA, USA, 1991.

[5] Koren, Y. and Borenstein, J., “Potential Field Methods and their Inherent Limitations for Mobile Robot Navigation,” Proc. of the 1991 IEEE Conference on Robotics and Automation, Apr. 1991, pp. 1398–1404.

[6] Wettergreen, D., Gaskett, C., and Zelinsky, A., “Development of a Visually-Guided Autonomous Underwater Vehicle”. Proc. of the IEEE Conference on OCEANS98, 1998.

[7] Mester, Gy. “Sensor Based Control of Autonomous Wheeled Mobile Robots,” The Ipsi BgD Transactions on Internet Research, TIR, Vol. 6, No. 2, 2010, pp. 29-34.

[8] Kiss, L., Várkonyi-Kóczy, A.R., and Ruano, A., “A Hybrid Autonomous Robot Navigation Method Based on Artificial Intelligence and Soft Computing Techniques,” Proc. of the IFAC International Conference on Intelligent Control Systems and Signal Processing, ICONS 2003, Faro, Portugal, April 8-11, 2003, pp. 251-256.

[9] Várkonyi-Kóczy, A.R., “A Universal Autonomous Robot Navigation Method,” International Journal of Advanced Computational Intelligence and Intelligent Informatics (JACIII), Vol. 12, No. 2, March 2008, pp. 206-223.

[10] Canny, J.F., “The Complexity of Robot Motion Planning,” *MIT Press*, Cambridge, MA, 1988.

[11] Russell, S.J. and Norvig, P., “Artificial Intelligence - A Modern Approach,” *Prentice-Hall*, 1995.

[12] Stentz, A., “Optimal and Efficient Path Planning for Partially-Known Environments,” Proc. of the IEEE International Conference on Robotics and Automation, San Diego, CA, Vol. 4, May 1994, pp. 3310-3317.

[13] Van der Berg, J., Ferguson, D., and Kuffner, J., “Anytime path planning and replanning in dynamic environments,” Proc. of the 2006 IEEE International Conference on Robotics and Automation, pp. 2366-2371.

[14] Lozano-Pérez, T. and Wesley, M.A., “An algorithm for planning collision-free paths among polyhedral obstacles,” Communications of the ACM, 1979.

[15] Rimon, E. and Koditschek, D.E., “Exact Robot Navigation Using Artificial Potential Functions,” IEEE Transactions on Robotics and Automation, Vol. 8, 1991, pp. 501-518.

[16] Visontai, M., Kiss, L., Baranyi, P., and Várkonyi-Kóczy, A.R., “3-Dimensional Potential Based Guiding,” Proc. of the IEEE Conference on Intelligent Engineering Systems, INES2000, Portorož, Slovenia, 2000, pp. 305-308.

[17] Lumelsky, V.J. and Stepanov, A.A., “Path-planning Strategies for a Point Mobile Automation Moving Amidst Unknown Obstacles of Arbitrary Shape,” Algorithmica, 1987.

[18] Tsai, C., Hu, S., Huang, H., and Hsieh, S., “Fuzzy hybrid navigation of an active mobile robotic assistant: A multisensory fusion approach,” IEEE Workshop on Advanced Robotics and Its Social Impacts ARSO 2007, Hsinchu, Dec. 9-11, 2007, pp. 1-6.

[19] Chen, C., Li, H. and Dong, D., “Hybrid Control for Robot Navigation A Hierarchical Q-Learning Algorithm,” IEEE Robotics and Automation Magazine, Vol. 15, No. 2, June, 2008, pp. 37-47.

[20] Berg, M., Kreveld, M., Overmars, M., and Schwarzkopf, O., “Computational Geometry,” *Springer-Verlag*., 2000, pp. 251–265.

Potential field-based approach for obstacle avoidance trajectories

Claudiu Pozna, Radu-Emil Precup, Laszlo T. Koczy, Aron Ballagi

Abstract: This paper proposes a new mathematical description of the potential field used in obstacle avoidance trajectory design. The main benefits of this description are the quickness of minimum computation and the compensation for the main drawbacks of potential field method. After the presentation of the potential field definition and its minimum computation this concept is included into an obstacle avoidance trajectory design method expressed under the form of an obstacle avoidance trajectory algorithm. A state-space controller is designed in order to control the car on the obstacle avoidance trajectory. Digital simulations performed for a complete dynamic model of a car validate the method.

Index: Obstacle avoidance, potential field, trajectory design, control, simulation.

1. INTRODUCTION

In the field of obstacle avoidance trajectory, several methods are known. These methods can be classified in the following ways. From the avoidance manoeuvre concept point of view the classification includes the potential field method, the vector histogram method, the curvature-velocity method and also methods based on artificial intelligence (AI) tools including the fuzzy logic approach. From the point of view of the road map decomposition these methods can be divided into global and local ones.

The potential field method is based on the artificial elastic mesh construction according to the results presented in [3,4,5,13,14,15]. This mesh incorporates obstacles and car trajectory. More precisely, the obstacles generate repulsive forces on the car and the trajectory generates attractive forces. The equilibrium between reactive and attractive forces deforms the desired trajectory and transforms it into an obstacles avoidance trajectory. The method was developed for static or even mobile obstacles. From this concept the elastic band concept is derived [11] which considers that the avoidance trajectory is the deformed shape of an elastic band (shape deformed by the obstacles). Another use of potential field is the velocity obstacle approach, where the potential collision is computed employing an obstacle motion prediction [8]. The histogram approach suggested in [2] computes both the motion direction and the velocity from the transformation of the occupancy into a histogram description.

The curvature-velocity method analyzed in [7, 9] is based on the assumption that the vehicle moves in circular or linear paths. The motion commands are searched directly in the space of possible linear and circular velocity. The concept of possible velocities is laid with the concept of dynamic window. In the end, to find the appropriate combination of linear and rotational velocities (included in the dynamic window) a minimization problem is solved. The method is developed by wave propagation techniques in the dynamic window [16].

Fuzzy logic methods have been developed and discussed in [6, 19] because the obstacle avoidance means navigation in uncertain environments, and these methods can deal with vague, imprecise and uncertain information. The well-known robustness of these methods is another reason that motivates these trails.

If the road map decomposition point of view is accepted, the global methods assume that a complete model of the environment is available; this means that a complete trajectory from the starting point to the target can be wither computed online or stored in memory. Local approaches use a small fraction of the world model in order to generate the control signals.

Because the method to be proposed here belongs to the category of potential field methods, it possesses the well-acknowledged drawbacks of these methods [12, 28]: trap situations due to local minima; no passage between nearness obstacles; oscillations in the presence of obstacles; oscillations in narrow passages.

A different variant of these methods is to divide the avoidance manoeuvre in two steps, planning the avoidance trajectories and next control the vehicle on these trajectories. Such an algorithm is proposed in [3], and it is based on the computation of the trajectories from the sum of repulsive and attractive force; a PID controller is

Manuscript received October 30, 2010. This work was supported in part by the Project PNII Idei 842

Claudiu Pozna. Author is with the Informatics Department, University Szechenyi Istvan Gyor Hungary and Product Design and Robotics, University Transilvania of Brasov Romania (e-mail: pozna@sze.hu).

Raud–Emil Precup is with the Department of Automation and Applied Informatics, “Politehnica” University of Timisoara, Romania (e-mail: radu.precup@aut.upt.ro).

Laszlo T. Koczy. Author is with Inst. of Mechanical Engineering and Information Technology Szechenyi Istvan University Gyor and Departament of Telecommunications and Media Informatics Budapest (koczy@tmit.bme.hu)

Aron Ballagi is with Department of Automation Szechenyi Istvan University Gyor (ballagi@sze.hu)

tuned such that to control the vehicle on the avoidance trajectory.

From the authors' point of view this method has the following shortcomings: the equilibrium position computation is a time consuming process; in real-world applications the shape and the dimensions of the obstacles are discovered during the avoidance manoeuvre (so it is impossible to offer a global solution), and just a PID controller does not seem to be enough robust to cope with the dynamic model of the car.

This paper extends the previous theoretical results reported in [28]. Another application is included and described shortly.

The paper is organized as follows. The next section starts with presenting the main concepts in obstacle avoidance. Then, the minimum potential field concept is profoundly studied from the problem definition up to the mathematical model involved. Section 3 is focused on an original obstacle avoidance trajectory algorithm based on the previous potential field definition. Simulations for a car example are presented in Section 4, and Sections 5 ends the paper with conclusions.

2. THE MINIMUM POTENTIAL FIELD

The definition of the task to be accomplished should be presented firstly. The robot is following a trajectory, which is a priori defined in the map and in a certain moment, discovers an obstacle that makes the initial trajectory impossible to pursuit. The car control system must react and fulfil the following online tasks: discover the shape and position of the obstacle; compute the avoidance trajectory; control the car on the trajectory; return to the initial trajectory.

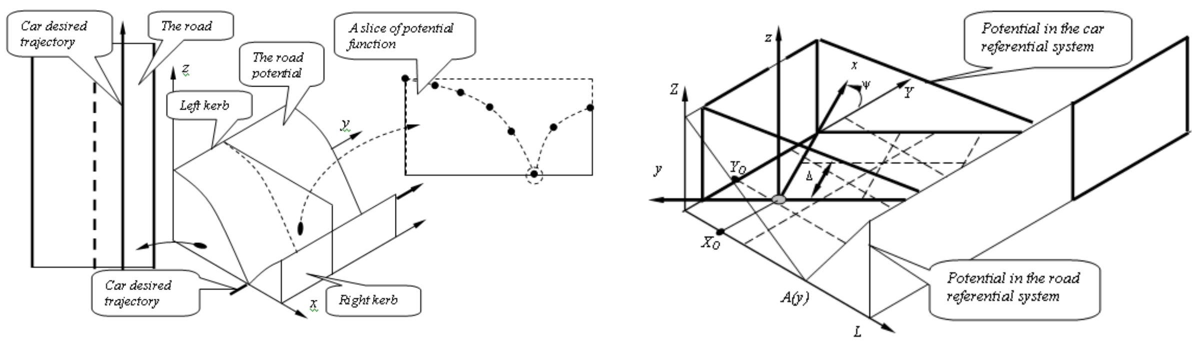
Since the obstacle is discovered online, during the avoidance manoeuvre, the avoidance trajectory will be composed by several parts; this means that, on line, the control system design must make use of a local approach as a loop composed by: discovering the obstacle and the road (the universe), designing a part of the avoidance trajectory and controlling the car on this trajectory part. Because this is an online process, any time consuming computations must be avoided. For this reason, it is needed to link together from a mathematical point of view the knowledge about the universe with the creation of the trajectory. This means that it is important to consider the sensor behaviours in the design of the obstacle avoidance trajectory algorithm final stage.

The analysis of some actual approaches concerning the potential field method [3,4,5,13,14,15] leads to the conclusion that the avoidance trajectory represents the static deformation of a hypothetical elastic network associated with the robot, road and obstacle. It is also known that the equilibrium of these kinds of structures can be obtained in terms of imposing a minimum potential energy condition.

The new approach starts with the idea stating that it is not necessary to construct such a complicated potential field and compute the numerical solution of equilibrium. In fact, it is more suitable to construct a simple potential field, which can be used (minimized) relatively quickly.

The construction of the potential field is based on finding a mathematical function having the following properties:

- in the absence of the obstacle the minimum of this function is the initial trajectory, the presence of the obstacle will generate a new minimum to avoid the obstacle,
- after the obstacle avoidance the trajectory must converge to the a priori one,
- the function definition can be connected to the relative speed between the obstacle and the car,
- the function can be constructed easily employing information from sensors data and initial trajectory data,
- the computation of the minimum must be less time-consuming process.



a) b)

Figure 1.

In order to present the mathematical function of the potential field, the potential of the road in absence of obstacles is presented. First a mesh $(x_{1...n}, y_{1...m})$ is defined on the road (in the O*xy* plane) and for each grid a potential P_R is defined according to the mathematical function (1):

$$P_R = z(x, y),\ (x, y) \in (x_{1...n}, y_{1...m}). \qquad (1)$$

The graphical representation of this function is illustrated in Figure 1a. The function has two maximums linked to the road margins and a minimum to the desired trajectory of the car. Because the potential function is discrete, several *slices* $P_R = z(x_{1\ldots n}, y_i),\ i = \overline{1,m}$ can be considered. The potential minimum of the slice is a point on the desired a priori trajectory.

In the end, the potential function minimum is a collection of *m* points. Because the obstacles are discovered in the car referential system, the detailed expression of the function (1) must be defined here and it must preserve the a priori trajectory. Figure 1b illustrates the link between the road and the car referential systems.

The mathematical expression of the function in (1) expressed in the car referential system is (2):

$$z(x,y) = \frac{|\,y - a(x,y)\,|}{2}[m_2(\mathrm{sgn}(y - a(x,y)) + 1) - m_1(\mathrm{sgn}(y - a(x,y)) - 1)] \quad (2)$$

where:

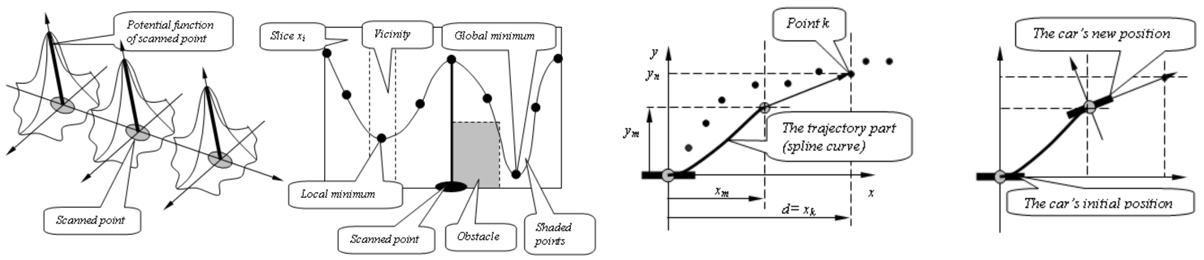
$$a(x,y) = \frac{X_o - A(y) - x\sin\psi}{\cos\psi}, \quad (3)$$

$A(y)$ is the desired position of the car in the road referential frame, $(X_O\ , Y_O)$ is the car position on the road referential frame, ψ is the car orientation, $m_1 = 1/[L - A(y)], m_2 = 1/A(y)$ and L is the width of the road.

Since the use of a laser scanner is intended, the obstacles are recognized like collections of points in the Oxy plane. More precisely, such a point is defined by the distance and angle between the car (scanner) referential and the scanned point belonging to the object. The obstacles potential is constructed for the same mesh grid $(x_{1\ldots n}, y_{1\ldots m})$ already defined for the road potential function. The idea is to generate a potential function for each scanned point, so in the end the obstacle potential function will be a sum of potential functions. The mathematical model of the obstacle potential function P_O is:

$$P_O = \sum_{i=1}^{k} e^{-\frac{c_1(y-y_i)^2 + c_2(x-x_i)^2}{2\sigma}}, \quad (4)$$

where: c_1, c_2 and σ are the parameters linked to the steering motor performance and the relative speed between the car and the obstacle, and (x_i,y_i) is the position of the *i*-th scanned point with index $i = \overline{1,k}$, $(x,y) \in (x_{1\ldots n}, y_{1\ldots m})$. The object identification is not generated and the scanned points are used directly. Figure 2a illustrates such a potential function for three scanned points ($k = 3$).



a), b)

Figure 2

If the two potential functions defined here will be added, the *universe* (road and obstacles) potential function P_U will be obtained according to (5):

$$P_U = P_R + P_O\,. \quad (5)$$

The avoidance trajectory at the first step will be defined like a collection of points (6):

$$\Gamma = \{(x_i, y_i^{\min}) \mid i = \overline{1,n}\}\,. \quad (6)$$

This collection of points is computed with the minimum of the *universe* potential function. In order to obtain this minimum, the optimization problem (7) must be solved:

$$y_i^{\min} = \arg\min_y P_U(x_i, y), \quad \text{subject to } y \in \{y_{1\ldots m} \mid y_{i-1}^{\min} - I < y_{1\ldots m} < y_{i-1}^{\min} + I\} \quad (7)$$

where $I = \mathrm{const}$ represents the vicinity radius. Several numerical methods can be employed to solve this problem [22, 27].

The computational aspects related to (7) are not time-consuming because the potential field is divided into slices and for each slice the computation of the minimum consists in finding the smallest element of a vector. The search area is constrained to $y_i \in (y_{i-1}^{\min} - I, y_{i-1}^{\min} + I)$ in order to avoid the jump of the minimum point from a local minimum, which is in the vicinity of the previous minimum, to a global minimum. The reasons of this constrained are:

- there are cases when the global minimum is a *shaded point* (in the obstacle back), so choosing this point means passing through the obstacle (see Figure 2a),
- even if the global minimum point is *not* a shaded point, these jumps of minimum point outside the vicinity $y_i \in \left(y_{i-1}^{\min} - I, y_{i-1}^{\min} + I\right)$ will generate a rugged trajectory.

.

Two strategies can be imagined characterized by either using (directly) the minimum point in the command decision or split the problem in two steps, first designing the avoidance trajectory and second, controlling the car on this trajectory. The second version is chosen here but the first one is also very attractive to be developed in a future work. With (7) a collection of minimum potential points is obtained. Based on this result a smooth curve is designed, and this curve will be the avoidance trajectory. Some comments with this regard are:

- a C_2 class curve must be designed because the car can follow (without slippage) linear, circular or clotoidal trajectories,
- because the obstacle is discovered during the avoidance manoeuvre, only a part of these points and will be used to define a part of the avoidance trajectory,
- the avoidance trajectory will be composed by several parts.

This idea is illustrated in Figure 2b.

In order to define the mentioned part of the trajectory the following steps are necessary:

- from the minimum points, select the first *k* and compute the middle point (x_{mid}, y_{mid}),
- define the direction *t* using the middle point and point *k*,
- corroborate the initial position, direction (tangent to O*x*) and curvature (0) with the final position (x_{mid}, y_{mid}), direction (*t*), curvature (0) and define a spline trajectory.

For each trajectory part a polynomial function will be obtained:

$$y(x) = a_5 x^5 + a_4 x^4 + \ldots + a_0 , \tag{8}$$

where $a_{0 \ldots 5}$ are found by imposing the boundary conditions (9):

$$\begin{aligned} &y(0) = 0, \dot{y}(0) = 0, \ddot{y}(0) = 0, \\ &y(x_{mid}) = y_{mid}, \dot{y}(x_{mid}) = t, \ddot{y}(x_{mid}) = 0 \end{aligned} . \tag{9}$$

The following comments are emphasized in this context:

- the number of points (*k*) is related to a certain length *d* (Figure 2b), choosing the first *k* points means this prediction is trusted enough, so that the trajectory will be followed upon the distance $d = x_{mid}$,
- the final trajectory is composed by several parts,
- since each part is a C_2 curve and the boundary conditions are considered, the final trajectory will be also a C_2 curve.

This algorithm includes the avoidance trajectory definition and also decisions that must be taken by the control system during this maneuver. The algorithm consists of the following steps:

1. Set the desired trajectory in accordance with the global map of the locomotion. This trajectory is not related to the unknown obstacle.
2. Scan the road permanently and, if no obstacles are found, control the car on the desired trajectory.
3. If obstacles are scanned, compute the avoidance trajectory as follows:
    1. Adjust the coefficients $c_{1,2}$ in accordance with the relative speed between the car and the scanned points. Compute the potential function in (5).
    2. Compute the minimum of the potential function using (6) and (7).
    3. Choose the first *k* points for a predicted trust distance *d* and compute the part of the avoidance trajectory using (8).
    4. Control the car on the avoidance trajectory.
    5. If during the locomotion the sensors accept this trajectory from the beginning to the end, go to step 2. Else, continue with 3.6.
    6. If sensors will denial the trajectory, adjust the predicted trust distance *d* and go to 3.1. Else, stop.

This algorithm needs sensor data fusion. With this sensor system several variables must be computed including the position of the scanned point and the relative velocity between the robot and scanned point. Moreover, this system must enable the decision making on either accepting or rejecting of the current trajectory part.

In order to simulate the avoidance manoeuvre, a set of Matlab [20] programs has been developed. The scanner behaviour, which discovers this *universe*, can be simulated and the potential function and its minimum can be computed. All these actions are made for each part of the avoidance trajectory. In the end, the entire image of *universe* and avoidance trajectory is presented. We note that running these programs, the robot discovers the original unknown *universe* step by step, and takes decision using local knowledge.

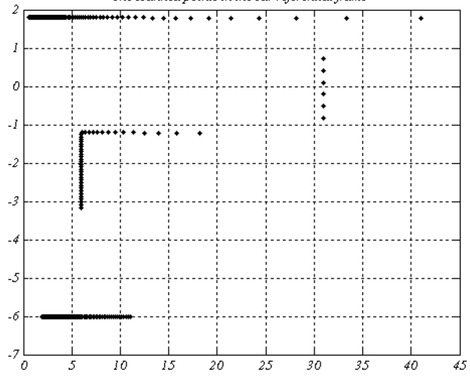


Figure 3

A sample of the simulation results is presented in Figure 3 and Figure 4. Since the avoidance trajectory (in this case) is composed by several parts (here 20) only the results corresponding to step 10 are presented here, and the final result has been inserted. The entire avoidance trajectory is composed by all 20 parts in this case. The plots presented in Figure 3 show the scanned points in the car referential frame and Figure 4 show the potential of road and obstacle.

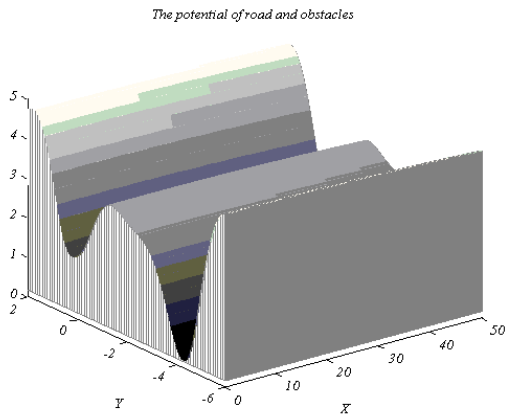


Figure 4

Figure 5 illustrates several avoidance trajectories for different positions of the obstacle. More precisely the obstacle was translated on the road width and the simulations were run for different situations.

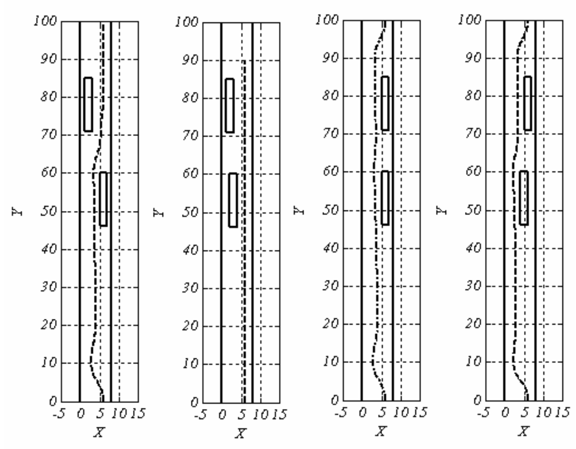


Figure 5.

3. *Conclusion*

The proposed obstacles avoidance trajectory methods belong to the potential field methods. The idea was to replace the nonlinear potential functions, which require numerical methods in order to compute its minimum with discrete potential functions subject to very quickly minimization. We consider that the present work improves the potential field methods due to its mentioned mathematical construction and because it integrates directly the sensor data in the mathematical model and it avoids the mentioned oscillations narrow obstacles and passages.

The suggested method has bee integrated to an obstacle avoidance algorithm. According to this algorithm, the vehicle discovers *on line* the *universe* and takes decision in order to reach its goal. The goal is an a priori trajectory that must be transformed because of the (initial unknown) obstacles. These transformations are done in two steps, first the obstacle avoidance trajectory is designed next the control design on this trajectory is performed. The controller design is based on a state-space method.

Several parts compose the obstacle avoidance trajectory. Each part is related to the knowledge about the local regions. Locomotion on such trajectory part involves a certain trust distance, which is subject to online adjustments.

The algorithm was validated by digital simulation of both the algorithm and the control system behaviour. Future research will be focused on:

- experiments,
- the derivation of a mathematical model based on the car and steering system dynamic in order to compute the coefficients $c_{1,2}$, σ,
- the derivation of a mathematical model which will predict the part n of the avoidance trajectory before the car finishes the part n–1 of the avoidance trajectory,
- the incorporation of advanced control strategies to improve the control system performance indices, accompanied by several analyses [1, 24, 25, 26].

Acknowledgment

The paper is supported by the Romanian Ministry of Education, Research and Innovation through the PNII Idei project 842/2008.

The support of HUNOROB project (HU0045, 0045/NA/2006-2/ÖP-9), a grant from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Hungarian National Development Agency, a Széchenyi István University Main Research Direction Grant, and National Scientific Research Fund Grant OTKA K75711 is also acknowledged

References

[1] Benadero, L., Giral, R., A. El Aroudi and J. Calvente, Stability analysis of a single inductor dual switching dc–dc converter, Mathematics and Computers in Simulation, Volume 71, Issues 4-6, June 2006, 256-269.
[2] Borenstein J., Koren, Y., The Vector Field Histogram – Fast Obstacle Avoidance for Mobile Robots, IEEE Transactions on Robotics and Automation, 7(3), June (1991), 278-288.
[3] Brandt, T., Sattel, T., Wallaschek, J., On automatic Collision Avoidance Systems. SAE International (2005), 316-326.
[4] Brock,O., Generating robot motion The integration of planning and execution, PhD thesis, Stanford Univ. (1999).
[5] Brock, O., Khatib, O., Viji,S., Task – Consistent Obstacle Avoidance and Motion Behaviour for Mobile Manipulation, In Preceding of the 2002 IEEE

International Conference on R obotics & Automation Washington, DC May 2002, 388-393.
[6] Fayad, Webb, Fuzzy Logic Based Collision Avoidance Algorithm for a M obile Robot, Workshop on Recent Advances in Soft Computing 1999, (July 1-2 1999), De Montfort University, Leicester.
[7] Feiten,W., Bauer, R., Lawitzky, G., Robust Obstacle Avoidance In Unknown and Cramped Environments, in IEEE Intl. Conf. on Robotics and Automation ICRA'94, San-Diego, May (1994), 2412-2417.
[8] Fiorini, P., Shiller, Z., Motion Planning in Dynamic Environments using Velocity Obstacles, Intl. Journal on Robotics Research, 17 (7), , (1998), 711-727.
[9] Fox, D., Burgard, W., Thrun, S., The Dynamic Window Approach to Collision Avoidance, IEEE Robotics and Automation Magazine, 4(1), , March (1997), 23-33.Franklin, G. Feedback Control of Dynamic Systems. 1999.
[10] Hrich, K., Optimization of Emergency Trajectories of Autonomous Vehicles with Respect to Linear Vehicles Dynamic. IEEE/ASME 2005 International Conference on Advance Intelligent Mechatronic California SUA (2005), 245-252.
[11] Isermann, R.,: Diagnosis Methods for Electronic Controlled Vehicles. Vehicle System Dynamics 36, No.2-3 (2001) 77-117.
[12] Khatib, O., Real-time Obstacle Avoidance for Manipulators and Mobile Robots, The Intl. Journal of Robotics Research, 5(1) (1986) 90-98.
[13] Khatib, M., Chatila, R., An Extended Potential Field Approach for Mobile Robot Sensor-based Motions, In Intl. Conf. On Intelligent Autonomous Systems IAS'4, (1995) 135-143.
[14] Ko, N.Y., Simmons, R.G., Seo, D.J., Trajectory modification Using Elastic Force for Collision Avoidance of Mobile Manipulator, In Springer-Verlag Berlin Heidelberg (2006), 190-199.
[15] Latombe, J.-C., Robot Motion Planning, Kluwer Academic Publishers, Boston, (1991).
[16] Lee W. S., Rao, M. V. C, Modelling and des ign of tape transport mechanism, Mathematics and Computers in Simulation, Volume 72, Issue 1, July 2006, 26-37.
[17] Maamri, N., Bachelier , O., Mehdi, D., Pole placement in a uni on of regions with prespecified subregion allocation, Mathematics and Computers in Simulation, Volume 72, Issue 1 , July 2006, 38-46.
[18] Maeda, N., Collision Avoidance Control Among Moving Obstacles for a Mobile Robot on the Fuzzy Reasoning," in Proceedings of the Symposium on Theory and Practice of Robots and Manipulators, (1990), 135-142.
[19] Mathworks: Matlab User's Guide, Mathworks Inc., Natick, MA, 2000.
[20] Ogata, K., Modern Control Engineering. Prentince Hall, 1997.
[21] Pérez, L.V., Bossio, G.R., Moitre, D., García,G.O., Optimization of power management in an hybrid electric vehicle using dynamic programming, Mathematics and Computers in Simulation, Volume 73, Issues 1-4, Nov. 2006, 244-254.
[22] C. Pozna and F . Troester, Human Behavior Model Based Control Program for ACC Mobile Robot, Acta Polytehnica Hungarica, Volume 3, (2006), 59-70.
[23] Precup, R-E., S. Preitl, S., Optimisation criteria in development of fuzzy controllers with dynamics, Engineering Applications of Artificial Intelligence, vol. 17, Sept. 2004, 661–674.
[24] Precup, R-E., S. Preitl, S., Development method for low cost fuzzy controlled servosystems, Proceedings of 2006 International Symposium on I ntelligent Control, Munich, Germany, 2006, 2707–2712.
[25] Pryymak, B., Moreno-Eguilaz, J. M., Peracaula, J., Neural network flux optimization using a model of losses in induction motor drives, Mathematics and Computers in Simulation, Volume 71, Issues 4-6, June 2006, 290-298.
[26] Tlustý, J., Santarius, P., Valouch, V., J. Škramlík, J., Optimal control of shunt active power filters in multibus industrial power systems for harmonic voltage mitigation, Mathematics and Computers in Simulation, Volume 71, Issues 4-6, June 2006, 369-376.
[27] Triandaf, I., Schwartz, B., A collective motion algorithm for tracking time-dependent boundaries, Mathematics and Computers in Simulation, Volume 70, Issue 4, Dec. 2005, 187-202.
[28] Pozna, C., Troester,F., Precup, R-E., Tar, J.K., Preitl, S., On the Design of an Obstacle Avoiding Trajectory: Method and Simulation, Mathematics and Computers in Simulation, Volume 79, Issue 7, March 2009, 2211-2226.

Claudiu Pozna's degrees earned: engineer at Transilvania University of Brasov (1990): Ph.D. in Robotics (1998). Visiting professor at the following Universities: IUT Metz (2001); Herriot-Watt University Edinburgh (2002); Hochschule Heilbronn (2004-2005). Research interests: Artificial Intelligence; Robotics; Control Engineering. . He is currently with the.Szecheny Istvan University of Gyor and w ith the Transilvania University of Brasov. He is also editor at Journal of Advanced Computational Intelligence and Intelligent Informatics, Fuji Technology Press

Radu-Emil Precup (M'03–SM'07) received the Dipl.Ing. (Hons.) degree in Automation and Computers from the "Traian Vuia" Polytechnic Institute of Timisoara, Romania, in 1987, the Diploma in Mathematics from the West University of Timisoara, Timisoara, in 1993, and t he Ph.D. degree in Automatic Systems from the "Politehnica" University of Timisoara (PUT), Timisoara, in 1996. From 1987 to 1991, he was with Infoservice S.A., Timisoara. He is currently with the PUT, where he became a Professor in the Department of Automation and Applied Informatics, in 2000, and is currently a Doctoral Supervisor of Automation and Systems Engineering. He is also an H onorary Professor with the Óbuda University, Budapest, Hungary (previously named Budapest Tech Polytechnical Institution) and a member of the Doctoral School of Applied Informatics with the Óbuda University.

Laszlo T. Kóczy's degrees earned: Technical University of Budapest: M. Sc. Electrical Engineering (1975); M. Phil. Control Eng. (1976); Ph.D. (1977), Dr. habil. 1998, (a postdoctoral degree in Hungary the prerequisite of full professorship), Doctor of the Hungarian Academy of Science, (1998, the highest earnable postdoctoral degree in Hungary)Visiting positions: Various long-term and s hort-term visiting professorships since 1990 till current. Lecturing, Ph.D. supervision and r esearch projects (Australian National University; Murdoch University (Perth, Australia), University of New South Wales (Sydney, Australia); J. Kepler Universität Linz (Austria); University of Trento (Italy); Tokyo Institute of Technology (Yokohama, Japan; Chair Professor in 1993/94); Pohang Institute of Science and Technology (Korea).
Summer University lecturing: Helsinki University of Economics, University of Helsinki (Finland), University of Minas Gerais (Belo Horizonte, Brazil), Dalian Maritime University (China). Research interests: Telecommunication systems, Intelligent models and s ystems, Very large and complex systems and networks. Professional Societies: International Fuzzy Systems Association: President 2001-current; President Elect, 1999-2001; Vice President, 1995-99.IEEE: Senior Member, 1999-; NNS Regional Activities Chair 2001-2002; Chapters Chair 2002-.EURO WG on Fuzzy Sets (currently EUSFLAT): Founding Member, 1975-. Hungarian Fuzzy Society: Founding President, 1990-1999. Fuzzy Initiative Nordrheinwestfalen Advisory Board member, 1992-. ANZIIS Advisory Board member, 1994-. Hongkong University Grants Committee member, 1994- .

Áron Ballagi's degrees earned: University of Miskolc: M.Sc. in Mechanical Engineering, Branch of Machine-tool Constructor and Information Technology (1994), M.Sc. in Metallurgical Engineering, Branch of Automation (1996), B.Sc. in Economy Management (1997). Visiting positions: long-term and s hort-term visits and s cholarships as Ph.D. student at the following Universities: Heriot-Watt University, Edinburgh, Scotland; Technical University of Kosice, Kosice, Slovakia; University of Duisburg, Duisburg, Germany; Czech Technical University, Prague, Czech Republic; Technical University of Vienna, Vienna, Austria; Brno University of Technology, Brno, Czech Republic. Research interests: computational intelligence, fuzzy control, fuzzy signatures, fuzzy communication, autonomous robotics. Membership in scientific societies: John von Neumann Computer Society Robotics Section, Hungarian Fuzzy Association.

Welcome to IPSI BgD Conferences and Journals!

http://www.internetconferences.net

http://www.internetjournals.net

CIP – Katalogizacija u publikaciji
Narodna biblioteka Srbije, Beograd
ISSN 1820 – 4503 =
The IPSI BGD Transactions on Internet Research
COBISS.SR - ID 119128844

ISSN 1820-4511