Sliding Mode Based Behavior Control

Selim Yannier, Asif Sabanovic, Ahmet Onat, Muhammet Bastan

***Abstract*—** In this work, we suggested a new approach for the control of a mobile robot capable of being a building block of an intelligent agent. This approach includes obstacle avoidance and goal tracking implemented as two different sliding mode controllers. A geometry based behavior arbitration is proposed for fusing the two outputs. Proposed structure is tested on simulations and real robot. Results have confirmed the high performance of the method.

***Keywords*—**Autonomous Mobile Robot, Behavior Based Control, Fast Local Obstacle Avoidance, Sliding Mode Control.

I. Introduction

MOST of the works in the field of mobile robotics are based on one of the following assumptions: either the complete knowledge of the environment is a priori known as introduced by the operator (deliberative approach) or robot has no a priory information about the environment (behavior based or reactive approach) [1-3].

In the first case, so-called "model based" methods are used [2]. Requirement of a complete model of the environment is the main difficulty in those systems. Other drawbacks are the high computational power and large memory requirements [1, 2, 4]. Moreover, they do not effectively resolve navigation problems in real-world applications especially in the presence of multiple moving obstacles [5].

The second method is "sensor based". In this case the task is considered as a combination of more elementary tasks called "behaviors" [4, 6]. Programming the execution of a given task then reduces to finding the proper combination of those behaviors to produce the desired task [1, 2].

Many results on behavior based control of mobile robots [1, 4, 6] with variety of obstacle avoidance methods [5, 7] are already published. Moreover, Tunstel used fuzzy logic based control layers in his mapping robot [8]. A similar study is carried by Tsourveloudis also [9]. Luo and Chen used behavior based mobile robot to avoid disturbances of the Internet latency in remote control [10, 11]. Parker [12] has modified behavioral controls for multi-robot cases. Arkin [13] extended behavioral control architecture for multi-robot control. Eustace [14] created "Behavioral Synthesis Model" designed to facilitate cooperation and coordination between multiple robotic devices for execution of complex tasks. Fontan and Mataric demonstrated the application of the distributed behavior based approach to generate a multi robot controller [15].

There are already several implementations of primitive behaviors using variety of methods showing high performance when executed alone. However, once multiple goal realization, such as avoiding obstacle while driving toward a target point, comes into picture then the question "How to choose the most useful action?" comes to the picture. For action selection, Brooks used subsumption architecture; the output of only one layer is executed at a time [4]. Although this configuration works well in less crowded areas, in a real world application results are generally not satisfying. Many researchers suggested and applied fuzzy logic based controllers [8, 16, 17]. The advantage is, potentially conflicting functions can be fused in a natural and smooth way, so that a reasonable decision can be made to serve both functions.

The goal of this work is to propose a basic configuration for the mobile robots, capable of being a building block of an intelligent agent. For such a system, one can identify a number of requirements,

Multigoal support: control of a mobile robot must find the way to select the action that serves a maximum number of goals at the same time.

Robustness: in the case of failures or erroneous readings of the sensors, the robot must still show meaningful behavior within limits.

Platform independence: it should be applicable to mobile robots with different physical size, shape, mechanics and electronics.

II. Plant

The plant consists of two entities: agents and obstacles. Obstacles are entities that are limiting its actions.

A. Kinematics Model of Agents

Sample mobile agent is differential drive type, nonholonomic robot (Fig. 1) defined by [18],

$$\begin{aligned}\dot{x} &= v\cdot\cos\phi \\ \dot{y} &= v\cdot\sin\phi \\ \dot{\phi} &= \omega\end{aligned} \quad , \quad \begin{aligned} v &= (v_R + v_L)/2 \\ \omega &= (v_R - v_L)/L\end{aligned} \tag{1}$$

where $q = (x, y, \phi) \in \Re^3$ is the state of the robot (position and the orientation) in world coordinate frame, L is the length of

Manuscript received November 5, 2004.

S. Yannier is with the Sabanci University, Mechatronics Program, Orhanli Mevkii Tuzla, Istanbul 34956 Turkey. (phone: +90 216 483 9000-2323; fax: +90 216 483 9550; e-mail: selimy@su.sabanciuniv.edu).

A. Sabanoviç is with the Sabanci University, Mechatronics Program, Istanbul, Turkey. (e-mail: asif@sabanciuniv.edu).

A. Onat is with the Sabanci University, Mechatronics Program, Istanbul, Turkey. (e-mail: onat@sabanciuniv.edu).

M. Bastan was with the Sabanci University, Mechatronics Program, Istanbul, Turkey. (e-mail: muhammetb@su.sabanciuniv.edu).

the axis joining driven wheels and v is the velocity of the center of those wheels. Variables that should be controlled are right (v_R) and left (v_L) wheel's linear velocities respectively, which may be translated into the translational and rotational velocity variables $u = (v, \omega) \in \Re^2$ for convenience [18].

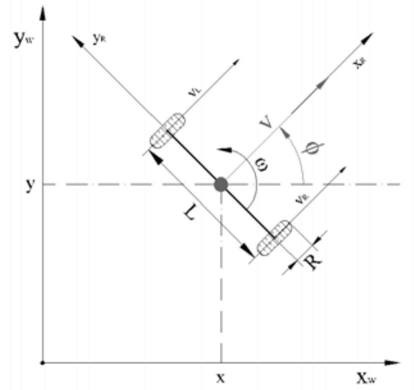

Fig. 1: Wheel set is used as sample physical agent.

1) Low Level Motion Control of Agents

LLMC is the layer where the robot is forced to follow reference velocity v_{ref} and orientation ϕ_{ref}. First, using actual position of the robot (x, y) reference position should be obtained,

$$\begin{aligned} \dot{x}_{ref} &= v_{ref} \cdot \cos\phi_{ref} \\ \dot{y}_{ref} &= v_{ref} \cdot \sin\phi_{ref} \end{aligned} \tag{2}$$

which can be combined as $r_{ref} = \sqrt{x_{ref}^2 + y_{ref}^2}$. Obviously, the control should be selected such that position errors $e_x = x_{ref} - x$ and $e_y = y_{ref} - y$ can be kept under certain threshold. Projection of those two errors on to the velocity and steering direction axis (denoted with subscript r and ϕ respectively) can be calculated.

$$\begin{aligned} e_r &= e_x \cdot \cos\phi + e_y \cdot \sin\phi \\ e_\phi &= -e_x \cdot \sin\phi + e_y \cdot \cos\phi \end{aligned} \tag{3}$$

We can then calculate corrected values for the reference values and corrected errors as

$$\begin{aligned} r_{ref}^{corr} &= r_{ref} + e_r \\ \phi_{ref}^{corr} &= \phi_{ref} + e_\phi \end{aligned} \quad \Rightarrow \quad \begin{aligned} e_r^{corr} &= r_{ref}^{corr} - r = \sigma_r \\ e_\phi^{corr} &= \phi_{ref}^{corr} - \phi = \sigma_\phi \end{aligned} \tag{4}$$

Choosing $u_1 = v_R + v_L$ and $u_2 = v_R - v_L$ as components of the control $\mathbf{u} = [u_1 \quad u_2]$ and using r_{ref}, eq-1 becomes;

$$\begin{aligned} \dot{r} &= u_1/2 \\ \dot{\phi} &= u_2/L \end{aligned} \tag{5}$$

Note the proportionalities; $u_1 \propto v$ ant $u_2 \propto \dot{\phi} = \omega$.

In above calculations the sliding manifold $\boldsymbol{\sigma} = [\sigma_r \quad \sigma_\phi]^T$ is chosen to be $\boldsymbol{\sigma} = [\sigma_r \quad \sigma_\phi]^T = [e_r^{corr} \quad e_\phi^{corr}]^T = \mathbf{0}$ since the aim of this control is to force those errors to decay to zero.

The control should be chosen such that components of the positive definite Lyapunov function candidate $\boldsymbol{\gamma} = \boldsymbol{\sigma}^T\boldsymbol{\sigma}/2 \geq \mathbf{0}$ satisfy Lyapunov stability criteria. Requesting its derivative to have the form $\dot{\boldsymbol{\gamma}} = -\mathbf{D} \cdot \boldsymbol{\sigma}^2 \leq \mathbf{0}$ we obtain;

$$\dot{\boldsymbol{\gamma}} = \boldsymbol{\sigma} \cdot \dot{\boldsymbol{\sigma}} = -\mathbf{D} \cdot \boldsymbol{\sigma}^2 \Rightarrow \boldsymbol{\sigma} \cdot (\dot{\boldsymbol{\sigma}} + \mathbf{D} \cdot \boldsymbol{\sigma}) = \mathbf{0} \tag{6}$$

for some constant $\mathbf{D} > \mathbf{0}$. In the above equation, either $\boldsymbol{\sigma} = \mathbf{0}$ or $(\dot{\boldsymbol{\sigma}} + \mathbf{D} \cdot \boldsymbol{\sigma}) = \mathbf{0}$ is zero. The former case do not provide any information for the behavior of the $\boldsymbol{\sigma}$, however in the later case $\boldsymbol{\sigma}$ will tend to zero, so the errors too, for $t \to \infty$. By selecting different condition for $\dot{\boldsymbol{\gamma}}$ finite time reaching to zero can be assured [18].

Solving above equation for discrete time systems where small computational delays are neglected we obtain [18];

$$\mathbf{u}^k = \mathbf{u}^{k-1} + (1/dt) \cdot ((1 + dt \cdot \mathbf{D}) \cdot \boldsymbol{\sigma}^k - \boldsymbol{\sigma}^{k-1}) \tag{7}$$

where dt is discrete time interval, k denotes the k^{th} time interval. Clearly, u^k belongs to the current time interval while u^{k-1} represents the past value.

Finally, actual references for the right and left wheel velocities for wheel velocity controllers are found as,

$$\begin{aligned} v_R^{ref} &= (u_1 + u_2)/2 \\ v_L^{ref} &= (u_2 - u_1)/2 \end{aligned} \tag{8}$$

2) Sensors of the Agent

Agents are equipped with sensors, which are required for feedback control. Sensors include, but are not limited to encoders (to determine the agent's location) and ultrasonic distance measurement sensors (to measure the distance to obstacles in the environment).

III. Proposed Solution: System Layer Design

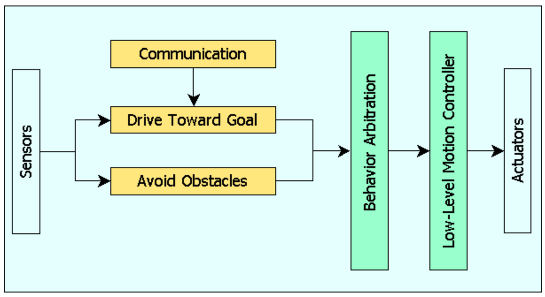

Fig. 2: Structure of the proposed solution.

Proposed control is a layered structure formed out of parallel and serial layers as shown in Fig. 2. Parallel layers are Obstacle Avoidance (OA) and Drive Toward Goal (DTG) behaviors that are performing independently and producing an output in the form of "desired change in the velocity and orientation". Serial layers on the other hand are connections of those parallel layers to the hardware.

For both OA and DTG behaviors, a force based method is proposed. The basic concept is to represent the sensor data of the robot as a repulsive force from the obstacles and to an attractive force toward the goal point. Then based on those two vectoral quantities, the agent can decide about its direction and velocity.

Used repulsive forces in this work has the form,

$$\vec{F}_{obs} = -A \cdot \sum_{i=1}^{n} \frac{1}{d_i^2} \cdot \hat{r}_i \qquad (9)$$

where the sum runs over all detected obstacles, A is a scaling factor, d_i is the distance to obstacle i, and $\hat{r}_i$ is the direction from the agent to the obstacle i. The inverse proportionality ensures significant increase in magnitude when the agent is too close to an obstacle, causing stronger reaction to avoid collision.

Used attractive force toward the goal point has the form,

$$\vec{F}_{atr} = B \cdot d^2 \cdot \hat{r} \qquad (10)$$

where B is the scaling factor, d is the distance and $\hat{r}$ is the direction from the agent to that point.

Note that the agent does only have local information on distances to obstacles and to goal point. In some applications, those two forces are summed and the resultant force is used to navigate the robot [19, 20]. However this total force mostly point away from the obstacle and as the agent gets closer to an obstacle gets dominant in the sum. Resulting the agent to move away from the obstacle, which also means moving away from the goal point, since that goal point must be somewhere behind that obstacle. Moreover, at many points the summed force may reach the balance having magnitude zero, which is actually the main reason why robots are stuck close to the passages like door openings.

A preferable approach is to make robot follow the obstacle boundary so that it can go around it to reach other side where goal point is located. This is generally done with special algorithms and requires considerable computational resource. In this work, the choice of separate treatment of goal and obstacle forces aims to implicitly realize circumnavigation behavior with minimum possible effort.

A. Obstacle Avoidance (OA)

This layer aims to orient the robot such that the total repulsive force vector from the obstacle is oriented with the axis joining the two wheels, since this simple condition ensures the agent to circumnavigate the closest obstacle. For this purpose, a sliding mode controller (SMC) is used.

First, the net repulsive force $\vec{F}_{obs}$ is calculated using sensor measurements. Then decomposed to two components: one along velocity direction of the agent $\vec{F}_r$ and other in the direction perpendicular to it, $\vec{F}_\phi$.

$$\begin{aligned} F_r &= \left\|\vec{F}_{obs}\right\| \cdot \cos\theta \\ F_\phi &= \left\|\vec{F}_{obs}\right\| \cdot \sin\theta \end{aligned} \quad , \quad \begin{aligned} &\theta = \phi - \theta_{obs} \\ &-\pi \le \theta \le \pi \end{aligned} \qquad (11)$$

where θ_{obs} is the orientation of $-\vec{F}_{obs}$ (from robot to the obstacle) in world coordinate frame.

The rate of change of those components is,

$$\begin{aligned} \dot{F}_r &= -\left\|\vec{F}_{obs}\right\| \cdot \dot{\theta} \cdot \sin\theta = -F_{obs} \cdot \left(\dot{\phi} - \dot{\theta}_{obs}\right) \cdot \sin\theta \\ \dot{F}_\phi &= \left\|\vec{F}_{obs}\right\| \cdot \dot{\theta} \cdot \cos\theta = F_{obs} \cdot \left(\dot{\phi} - \dot{\theta}_{obs}\right) \cdot \cos\theta \end{aligned} \qquad (12)$$

From here, one can conclude that, by changing orientation of the robot, control of both F_r and F_ϕ is feasible. By representing OA loop as two dimensional system,

$$\begin{aligned} \dot{F}_r &= u_r^{OA} \\ \dot{F}_\phi &= u_\phi^{OA} \end{aligned} \qquad (13)$$

one can design a controller. For a safe travel, the agent must be reoriented to keep $\vec{F}_r$, minimum, generally zero, while maximizing F_ϕ. Defining errors to be minimized as,

$$\begin{aligned} e_r^{OA} &= F_r^{ref} - F_r \\ e_\phi^{OA} &= F_\phi^{ref} - F_\phi \end{aligned} \qquad (14)$$

and the sliding surface as $\boldsymbol{\sigma}_{OA} = \begin{bmatrix} e_r^{OA} & e_\phi^{OA} \end{bmatrix}^T$, then using Lyapunov function candidate $\boldsymbol{\sigma}_{OA}^T \boldsymbol{\sigma}_{OA} / 2 \ge \mathbf{0}$ and procedure described in section II.A.1 we obtain

$$\begin{aligned} u_r^{OA,k} &= u_r^{OA,k-1} + (1/dt) \cdot \left(\left(1 + dt \cdot D_r^{OA}\right) \cdot e_r^{OA,k} - e_r^{OA,k-1}\right) \\ u_\phi^{OA,k} &= u_\phi^{OA,k-1} + (1/dt) \cdot \left(\left(1 + dt \cdot D_\phi^{OA}\right) \cdot e_\phi^{OA,k} - e_\phi^{OA,k-1}\right) \end{aligned} \qquad (15)$$

Using eq-13 and eq-15 together,

$$\frac{u_r^{OA}}{u_\phi^{OA}} = \frac{-\sin\theta^{OA}}{\cos\theta^{OA}} \Rightarrow \theta^{OA} = \tan^{-1}\left(-\frac{u_r^{OA}}{u_\phi^{OA}}\right) \qquad (16)$$

and $\Delta\phi^{OA} = \phi - \theta^{OA}$ where θ^{OA} is the reference orientation for a collision free path and $\Delta\phi^{OA}$ is "the desired change in the current orientation".

B. Drive Toward Goal Point (DTG)

This layer aims to orient the robot toward the goal point by orienting the velocity direction of the robot with the attractive force. For this purpose, a SMC is used.

Using any method such as dead reckoning the position of the robot is calculated. Then the attractive force $\vec{F}_{atr}$ and its components: one along velocity direction of the agent $\vec{G}_r$ and other in the direction perpendicular to it $\vec{G}_\phi$ are calculated.

$$\begin{aligned} G_r &= \left\|\vec{F}_{atr}\right\| \cdot \cos\theta' \\ G_\phi &= \left\|\vec{F}_{atr}\right\| \cdot \sin\theta' \end{aligned} \quad , \quad \begin{aligned} &\theta' = \phi - \theta_{atr} \\ &-\pi \le \theta' \le \pi \end{aligned} \qquad (17)$$

To obtain an orientation toward the goal point the force along the heading direction should be maximized $G_r^{ref} = \left\|\vec{F}_{atr}\right\|$, while the other component must be minimized, $G_\phi^{ref} = 0$. By following the same reasoning in III.A, we can find the rate of change of the goal forces as,

$$\begin{aligned} \dot{G}_r &= -\left\|\vec{F}_{atr}\right\| \cdot \dot{\theta} \cdot \sin\theta' = u_r^{DTG} \\ \dot{G}_\phi &= \left\|\vec{F}_{atr}\right\| \cdot \dot{\theta} \cdot \cos\theta' = u_\phi^{DTG} \end{aligned} \qquad (18)$$

we obtain our control variables u_r^{DTG} and u_ϕ^{DTG}. The errors are,

$$\begin{aligned} e_r^{DTG} &= G_r^{ref} - G_r \\ e_\phi^{DTG} &= G_\phi^{ref} - G_\phi \end{aligned} \qquad (19)$$

Defining the sliding surface as $\boldsymbol{\sigma}_{DTG} = \begin{bmatrix} e_r^{DTG} & e_\phi^{DTG} \end{bmatrix}^T$, using Lyapunov function candidate $\boldsymbol{\sigma}_{DTG}^T \boldsymbol{\sigma}_{DTG} / 2 \ge \mathbf{0}$ and procedure described in section II.A.1 we obtain

$$\begin{aligned} u_r^{DTG,k} &= u_r^{DTG,k-1} + (1/dt)\cdot\left(\left(1+dt\cdot D_r^{DTG}\right)\cdot e_r^{DTG,k} - e_r^{DTG,k-1}\right) \\ u_\phi^{DTG,k} &= u_\phi^{DTG,k-1} + (1/dt)\cdot\left(\left(1+dt\cdot D_\phi^{DTG}\right)\cdot e_\phi^{DTG,k} - e_\phi^{DTG,k-1}\right) \end{aligned} \quad (20)$$

Using eq-18 and eq-20 together,

$$\frac{u_r^{DTG}}{u_\phi^{DTG}} = \frac{-\sin\theta^{DTG}}{\cos\theta^{DTG}} \Rightarrow \theta^{DTG} = \tan^{-1}\left(-\frac{u_r^{DTG}}{u_\phi^{DTG}}\right) \quad (21)$$

and $\Delta\phi^{DTG} = \phi - \theta^{DTG}$ where θ^{DTG} is the reference orientation to move the robot toward the goal point and $\Delta\phi^{DTG}$ is the "the desired change in the current orientation of the robot.

C. *Behavior Arbitration*

While the robot navigates DTG produces $\Delta\phi^{DTG}$ and whenever an obstacle is sensed OA produces $\Delta\phi^{OA}$. Since the robot sensed the obstacle while moving toward the goal point, those two commands will be in conflict.

To be able to avoid obstacles while driving toward the goal point, $\Delta\phi^{DTG}$ and $\Delta\phi^{OA}$ must be combined such that both request are partially fulfilled. For this purpose, serial behavior arbitration layer calculating the weighted sum of $\Delta\phi^{DTG}$ and $\Delta\phi^{OA}$ to transmit to the low-level motion controller is proposed. Used non-constant weights are calculated from geometry.

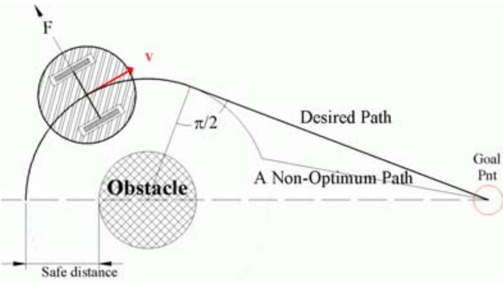

Fig. 3: Optimum and non-optimum path example for an agent while avoiding an obstacle.

Observing the agent moving toward the goal point, while avoiding an obstacle (Fig. 3), we can see that when the angle between $\vec{F}_{obs}$ and $\vec{v}$ is close to π, OA must gain importance while this angle is greater or equal to $\pi/2$ the collision has low probability and the DTG must gain importance. This can be done by defining two complimentary weights as,

$$A = 1 - B = \begin{cases} 1\ (\max) & for \quad \theta = \pi \\ \vdots & \\ 0\ (\min) & for \quad \theta \le \pi/2 \end{cases} \quad (22)$$

where θ is the angle between velocity $\vec{v}$ of the robot and repulsive force $\vec{F}_{obs}$. Then ϕ^{ref}, the reference orientation for the agent can be defined as,

$$\phi^{ref} = \phi + A^2\cdot\Delta\phi^{OA} + B^2\cdot\Delta\phi^{DTG} \quad (23)$$

Both A and B are used as square to increase the smoothness.

1) *Velocity Reference*

The control layers described above does not involve any velocity generation. Let $v^{ref}(t)$ be the scalar reference velocity that should be increased using maximal acceleration a starting from the initial time $t = t_i$, until the maximum velocity $v_{\max}$ is reached. On the other hand when the agent gets closer to the goal point this velocity should decrease. A suitable choice to generate this reference velocity is

$$v^{ref}(t) = \min\left(a\cdot t, v_{\max}\cdot\sqrt{2\cdot a\cdot d}\right) \quad (24)$$

where d is the distance to the goal point. [21].

The output of the BA layer is the reference velocity v^{ref} and orientation ϕ^{ref} that are sent to the low-level motion controller where the motor velocities are calculated. Although in this application, velocity reference is not affected in OA layer, a deceleration when an obstacle is detected and acceleration when the path is free could also be added.

D. *Communication*

Communication is the link between robot and user. This link can be used for the transmission of commands to the robot and transmission of collected data by the robot to the user. Moreover, communication can safely be used in multi-robot collaboration where small time delays due to the transmission time are not important.

IV. Simulations, Experiments and Results

Proposed control for mobile robots is tested on the developed simulation tool written in C++ language. Moreover, real experiments are conducted on "*SoccerBot*" designed and distributed by Thomas Bräunl et al., The University of Western Australia. In the real experiments, an overhead camera is used both to estimate the robots position and distance o the obstacles.

A. *Experiment: Stationary Obstacles*

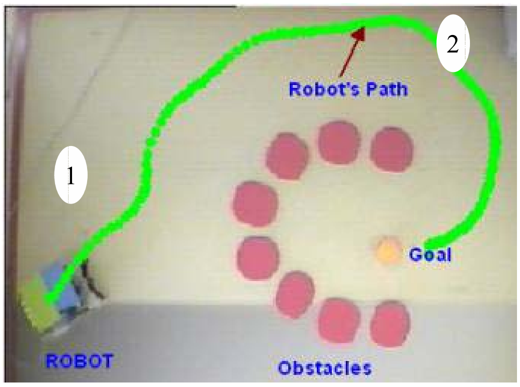

Fig. 4: Example experimental output record, mobile robot's path while avoiding obstacles (red) is shown (green).

Stationary obstacles are placed in the environment together with one agent (Fig. 4). We can observe smooth and safe navigation through obstacles. Moreover, the figure shows clearly the effect of the BA: first, the agent was moving toward the obstacle, then the OA layer influenced the robot in such a way that, the robot reoriented itself to circumnavigate the obstacle (point 1). At a later point (2), where obstacle is not between the agent and the goal point anymore, the behavior arbitration inhibited the output of the OA layer and the agent starts to move toward its goal point.

B. Experiment: Stationary Obstacles and Door Openings

Below (Fig. 5) is shown another configuration where the mobile robot smoothly travels among obstacles.

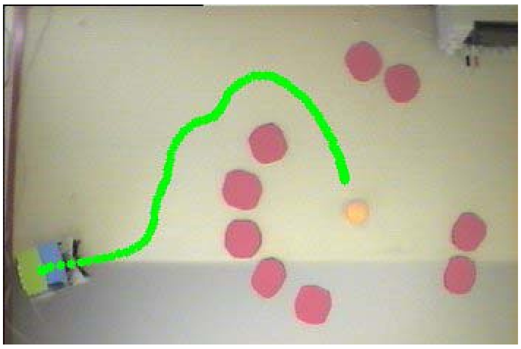

Fig. 5: Avoidance of stationary obstacles and door openings.

C. Experiment: Stationary Obstacles

Below (Fig. 6) is shown another configuration where the mobile robot smoothly travels. Remark that many mobile robot controllers stuck with those kinds of obstacles.

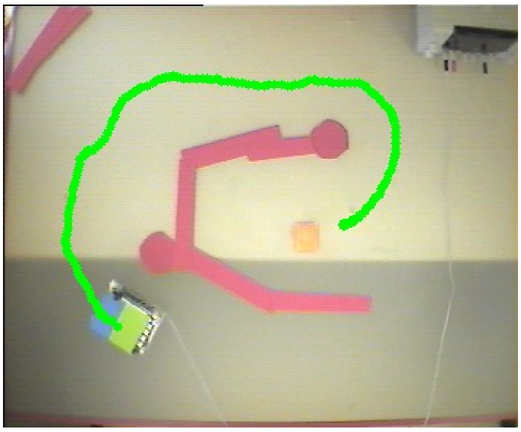

Fig. 6: Avoidance of stationary obstacles.

D. Experiment: Passages

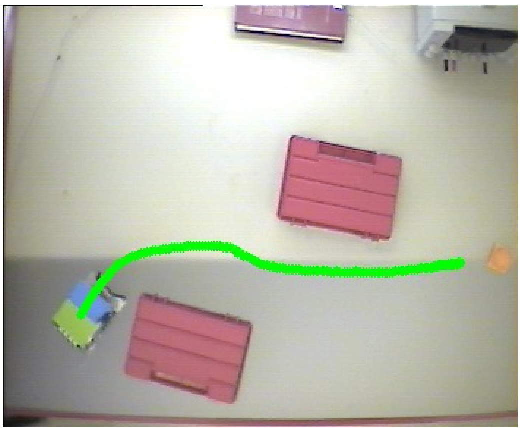

Fig. 7: Motion in a passage.

In this experiment, we tested the navigation of the agent in a relatively narrow passage. Passages and door opening are difficult places for mobile robots since they cause local minimas in the environment. In Fig. 7 we observe successful motion of the agent.

E. Simulation: Moving Obstacles

In this experiment, we tested the reaction of the agent to the moving obstacles (MO1, MO2, MO3 and MO4). As shown in Fig. 8, an agent is told to move from point S to point T.

First confrontation happened with MO1 (circled area 1). The agent reacted (moved toward bottom) quickly to avoid the obstacle. When the path was clear, it reoriented it-self toward T until next confrontation. Similar behavior is observed for other confrontations. We see clearly that the agent moves naturally and safely in the area where it encounters moving obstacles.

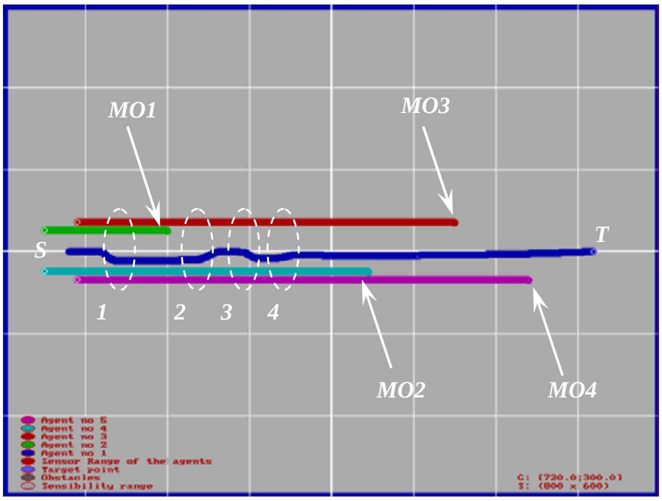

Fig. 8: Avoidance of moving obstacles.

F. Simulation: Complex Environments

Developed algorithm is also tested on relatively complex environments. One example where a single robot is trying to move toward the inner point of spirally shaped room (Fig. 9).

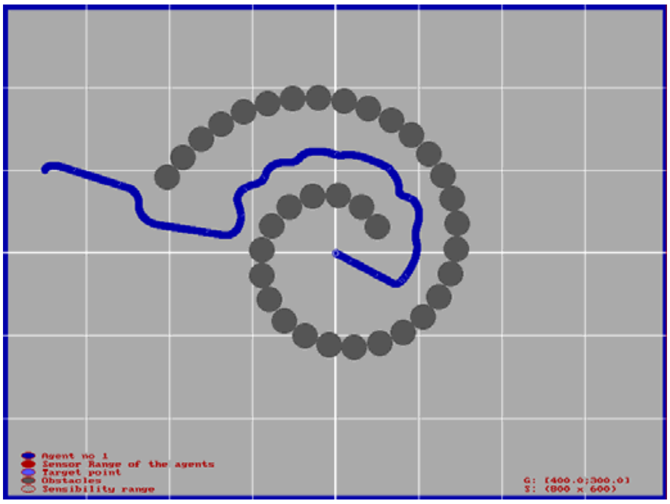

Fig. 9: A robot is trying to move toward the inner point of spiral.

V. Conclusion

In this work, we suggested a solution for the basic tasks of a mobile robot capable of being a building block of an intelligent agent. This solution includes obstacle avoidance and goal tracking implemented as two parallel controllers. A geometry based behavior arbitration is proposed for fusing the outputs. Reaching to a specific point while avoiding obstacles is a simple multi goal example for a mobile robot. Those two basic goals are already in the control, and working in harmony. Further control layers can be added to the control and this addition will augment richness of the behaviors observed.

Proposed structure is tested both on simulations and on real robot with different scenarios. Especially, problematic cases to many other approaches are investigated. Results confirmed the high performance of the method. We can conclude that the proposed control is a potential alternative for mobile robots operating in dynamic and unstructured environments and/or as an agent in multiagent system.

References

[1] R. C. Arkin, *Behavior-based robotics*. Cambridge, Mass.: MIT Press, 1998.

[2] J. Ferber, *Multi-agent systems: an introduction to distributed artificial intelligence*. Harlow, Eng.: Addison-Wesley, 1999.

[3] W. L. Xu and S. K. Tso, "Sensor-Based Fuzzy Reactive Navigation of a Mobile Robot Through Local Target Switching," *IEEE Trans. on Systems, Man and Cybernetics, Part C*, vol. 29, pp. 451-459, 1999.

[4] R. A. Brooks, "A robust layered control system for a mobile robot," MIT Artificial Intelligence Laboratory, Massachusetts A.I. Memo 864, Sep. 1985.

[5] K.-T. Song and C. C. Chang, "Reactive navigation in dynamic environment using a multisensor predictor," *IEEE Trans. on Systems, Man and Cybernetics, Part B*, vol. 29, pp. 870-880, 1999.

[6] R. A. Brooks, *Cambrian intelligence: the early history of the new A.I.* Cambridge, Mass.: MIT Press, 1999.

[7] A. Steinhage and R. Schoner, "The dynamic approach to autonomous robot navigation," presented at IEEE Int. Symp. on Industrial Electronics, ISIE' 97, 1997.

[8] E. Tunstel and M. Jamshidi, "Fuzzy logic and behavior control strategy for autonomous mobile robot mapping," presented at IEEE World Congress on Computational Intelligence, Third IEEE Conf. on Fuzzy Systems, 1994.

[9] N. C. Tsourveloudis, K. P. Valavanis, and T. Hebert, "Autonomous vehicle navigation utilizing electrostatic potential fields and fuzzy logic," *IEEE Trans. on Robotics and Automation*, vol. 17, pp. 490-497, 2001.

[10] R. C. Luo and T. M. Chen, "Development of a multi-behavior based mobile robot for remote supervisory control through the Internet," *IEEE/ASME Trans. on Mechatronics*, vol. 5, pp. 376 -385, 2000.

[11] T. M. Chen and R. C. Luo, "Development and integration of multiple behaviors for autonomous mobile robot navigation," presented at Proc. of the 24th Annual Conf. of the IEEE Industrial Electronics Society, IECON '98, 1998.

[12] L. E. Parker, "A performance-based architecture for heterogeneous, situated agent cooperation," presented at AAAI Workshop on Cooperation Among Heterogeneous Intelligent Systems, 1992.

[13] R. C. Arkin, T. Balch, and E. Nitz, "Communication of behavioral state in multi-agent retrieval tasks," presented at IEEE Int. Conf. on Robotics and Automation, 1993.

[14] D. Eustace, D. P. Barnes, and J. O. Gray, "A behavior synthesis architecture for co-operant mobile robot control," presented at Int. Conf. on Control, Control '94, 1994.

[15] M. Schneider-Fontan and M. J. Mataric, "Territorial multi-robot task division," *IEEE Trans. on Robotics and Automation*, vol. 14, pp. 815-822, 1998.

[16] C. Ma, W. Li, and L. Liu, "Mobile robot motion by integration of low-level behavior control and high level global planning," presented at IEEE Int. Conf. on Systems, Man and Cybernetics, 1996.

[17] S. S. Ge and Y. J. Cui, "New potential functions for mobile robot path planning," *IEEE Trans. on Robotics and Automation*, vol. 16, pp. 615-620, 2000.

[18] S. Yannier, "Realization of Reactive Control for Multi Purpose Mobile Agents," in *Electronics Eng. and Comp. Sciences*. Istanbul: Sabanci University, 2002, pp. 107.

[19] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," presented at IEEE Int. Conf. on Robotics Automation, St. Louis, MO, 1985.

[20] J. Borenstein and K. Y., "The Vector Field Histogram-Fast obstacle avoidance for mobile robots," *IEEE Transactions on Robotics & Automation*, vol. 7, pp. 278-287, 1991.

[21] J. Guldner and V. I. Utkin, "Sliding Mode Control for Gradient Tracking and Robot Navigation Using Artificial Potential Fields," *IEEE Trans. on Robotics and Automation*, vol. 11, pp. 247-254, 1995.

Selim Yannier was born in 1978, Istanbul, Türkiye. Received his B.S. degree in Physics with honors from Middle East Technical University, Ankara, Türkiye in 2001. He then began M.S. studies at Mechatronics Engineering Program, Sabanci University, Istanbul, Türkiye and completed at 2002. He is currently Ph.D student at the same institution.

He worked as a part time Technical Supporter and Software Developer in a computer company from 1993 to 1997. Then Yannier worked as a part time Software Consultant in Scala Business Solutions between 1998 and 2000. Currently he is working as a Research and Teaching Assistant in Sabanci University.

M.Sc. Yannier’s research interests are motion control, mobile robotics and control for nanotechnologies

Asif Sabanovic received B.S. degree in Electrical Eng. (in 1970), M.S. degree and Dr. Eng. degree from Univ. of Sarajevo. Since 1970 till 1991 he has been with Energoinvest Institute for Control and Computer Sciences, Sarajevo. From 1991 he has been with Univ. of Sarajevo, Department of Electrical Eng.. He has been Visiting Researcher at Institute of Control Science, Moscow (1975-1976). Visiting Professor at California Institute of Technology, Pasadena (1984-1985), Hitachi Chair Professor at Keio Univ., Yokohama (1991-1992), Full Professor at Yamaguchi Univ., Ube (1992-1993), Visiting Professor at Univ. of Maribor, Maribor, Slovenia (Acad. Year 1986, 1988, 1989), Head of CAD/CAM and Robotics Department at Tubitak Marmara Research Centre, Istanbul (1993-1995), Head of Eng. Department of B.H. Eng. and Consulting (1995-1999). He is a Full Professor at Sabanci Univ., Istanbul.

Prof. Dr. Sabanovic’ fields of interest include Control Systems, Motion Control Systems, Robotics, Mechatronics and Power Electronics.

Ahmet Onat was born in Istanbul in 1969. He received his B.S. degree in Electronics and Communication in 1991 from Istanbul Technical University, Turkey, and his M.S. and Ph.D. degrees from Kyoto University, Japan in 1995 and 1999 respectively, in Machine Learning. He is currently a member of Mechatronics Program, Sabanci University, Istanbul, Turkey. Dr. Onat is a member of the IEEE and the Steinbeis Foundation.

Muhammet Bastan was born in Trabzon in 1980. He received his B.S. degree in Mechanical Engineering Middle East Technical University, Ankara, Türkiye in 2001. He received his M.S. degree in Mechatronics Engineering Program, Sabanci University, Istanbul, Türkiye in 2003 and completed at Spring 2002.