World Academy of Science, Engineering and Technology 3 2005

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) [1-5].

There are already several successful implementations of behavior based control of mobile robots [1, 4, 6] with variety of obstacle avoidance methods [5, 7]. However, once multiple goal realization, such as avoiding obstacle while driving toward a target point, comes into picture then action selection becomes the key issue. Although Brooks' subsumption architecture [4] works well in less crowded areas, results in real world applications are not satisfying.

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, major requirements are,

Multigoal support: control must 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.

Platform independence: it should be applicable to mobile robots with different properties such as size and shape.

II. PLANT

The plant consists of two entities: agents and obstacles.

A. Simplified Model of Agents

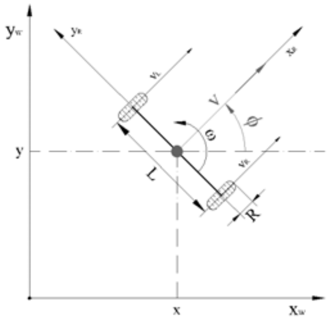

Fig. 1: Wheel set is used as sample physical agent.

Sample mobile agent is differential drive type, nonholonomic robot (Fig. 1) defined by [8],

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

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 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 [8].

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 is obtained,

$$\dot{x}_{ref} = v_{ref} \cdot \cos\phi_{ref} \quad , \quad \dot{y}_{ref} = v_{ref} \cdot \sin\phi_{ref} \qquad (2)$$

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 found.

$$e_r = e_x \cdot \cos\phi + e_y \cdot \sin\phi \quad , \quad e_\phi = -e_x \cdot \sin\phi + e_y \cdot \cos\phi \qquad (3)$$

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

$$\begin{matrix} r_{ref}^{corr} = r_{ref} + e_r \\ \phi_{ref}^{corr} = \phi_{ref} + e_\phi \end{matrix} \quad \Rightarrow \quad \begin{matrix} e_r^{corr} = r_{ref}^{corr} - r = \sigma_r \\ e_\phi^{corr} = \phi_{ref}^{corr} - \phi = \sigma_\phi \end{matrix} \qquad (4)$$

Choosing $u_1 = v_R + v_L$ and $u_2 = v_R - v_L$ as controls and using

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).

r_{ref}, eq-1 becomes;

$$\dot{r} = u_1/2 \quad , \quad \dot{\phi} = u_2/L \tag{5}$$

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

The control should be chosen such that components of the positive definite Lyapunov function candidate $\gamma = \sigma^T\sigma/2 \geq 0$ satisfy Lyapunov stability criteria. Since both equations are independent, we can control the components of the error vector separately;

$$\begin{aligned} &\dot{\gamma}_i = \sigma_i\dot{\sigma}_i = -D_i \cdot \sigma_i^2 \\ &\sigma_i\left(\dot{\sigma}_i + D_i \cdot \sigma_i\right) = 0 \end{aligned} \quad , \quad i = r, \phi \tag{6}$$

where, $\gamma_i \geq 0$ and $\dot{\gamma}_i \leq 0$, for $i = r, \phi$ and for some constant $D > 0$. In the above equation, either σ_i or $\left(\dot{\sigma}_i + D_i \cdot \sigma_i\right)$ is zero. If $\left(\dot{\sigma}_i + D_i \cdot \sigma_i\right)$ is zero for $\sigma_i \neq 0$, then obviously σ_i will tend to zero.

Solving above equation for discrete time systems where small computational delays are neglected we obtain [8];

$$\begin{aligned} u_1^k &= u_1^{k-1} + (1/dt) \cdot \left(\left(1 + dt \cdot D_r\right) \cdot \sigma_r^k - \sigma_r^{k-1}\right) \\ u_2^k &= u_2^{k-1} + (1/dt) \cdot \left(\left(1 + dt \cdot D_\phi\right) \cdot \sigma_\phi^k - \sigma_\phi^{k-1}\right) \end{aligned} \tag{7}$$

where dt is discrete time interval, k denotes the k^{th} time interval. Finally, actual references for the right and left wheel velocities for wheel velocity controllers are found as,

$$v_R^{ref} = \left(u_1 + u_2\right)/2 \quad , \quad v_L^{ref} = \left(u_2 - u_1\right)/2$$

(8)

2) *Sensors of the Agent*

Agents have sensors, which are required for feedback control. Sensors include, but are not limited to, encoders and ultrasonic distance measurement sensors.

B. *Obstacles*

Obstacles are entities that are either preventing the agent to move or limiting its actions.

III. PROPOSED SOLUTION: SYSTEM LAYER DESIGN

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

For both OA and DTG behaviors, a force control method is proposed. Used repulsive forces has the form,

$$F_{obs} = -A \cdot \sum_{i=1}^{n} \frac{1}{d_i^2} \cdot \hat{r}_i \tag{9}$$

where the sum runs over all 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.

Used attractive force toward the goal point has the form,

$$F_{atr} = B \cdot d^2 \cdot \hat{r} \tag{10}$$

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

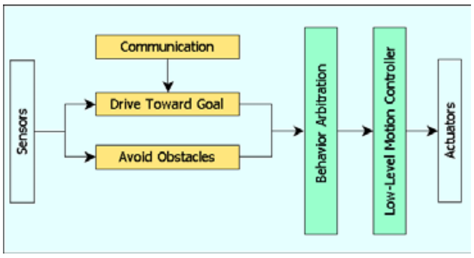

Fig. 2: Structure of the proposed solution.

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. In this work, the choice of separate treatment of goal and obstacle forces aims to implicitly realize such a circumnavigation behavior.

A. *Obstacle Avoidance (OA)*

This layer aims to orient the robot such that the net repulsive force from the obstacles is oriented with the axis joining the two wheels. For this purpose, a sliding mode controller (SMC) is used. First, the net repulsive force F_{obs} is calculated using sensor measurements. Then decomposed to two components: one along velocity direction of the agent F_r and other in the direction perpendicular to it, F_ϕ.

$$\begin{aligned} F_r &= \left\|F_{obs}\right\| \cdot \cos\theta \\ F_\phi &= \left\|F_{obs}\right\| \cdot \sin\theta \end{aligned} \quad , \quad \begin{aligned} &\theta = \phi - \theta_{obs} \\ &-\pi \leq \theta \leq \pi \end{aligned} \tag{11}$$

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

The rate of change of those components is,

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

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,

$$\dot{F}_r = u_r^{OA} \quad , \quad \dot{F}_\phi = u_\phi^{OA} \tag{13}$$

one can design a controller. Defining errors to be minimized,

$$e_r^{OA} = F_r^{ref} - F_r \quad , \quad e_\phi^{OA} = F_\phi^{ref} - F_\phi \tag{14}$$

For a safe travel, the agent must be reoriented to keep F_r, minimum, generally zero, $F_r^{ref} = 0$ while maximizing F_ϕ. Then using Lyapunov Function candidate $\mathbf{e}_{OA}^T\mathbf{e}_{OA}/2 \geq 0$ and procedure described in section II.A.1 we obtain

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

Using eq-13 and eq-15 together,

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

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)

In this layer, two different controllers are running: one for orientation and another for velocity reference generation.

1) Orientation Control

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 F_{atr} and its components: one along velocity direction of the agent G_r and other in the direction perpendicular to it G_ϕ are calculated.

$$\begin{aligned} G_r &= \left\|F_{atr}\right\| \cdot \cos\theta' \\ G_\phi &= \left\|F_{atr}\right\| \cdot \sin\theta' \end{aligned} \quad , \quad \begin{aligned} \theta' &= \phi - \theta_{atr} \\ -\pi &\le \theta' \le \pi \end{aligned} \tag{17}$$

To obtain an orientation toward the goal point the force along the heading direction should be maximized $G_r^{ref} = \left\|F_{atr}\right\|$, while the other component must be minimized, $G_\phi^{ref} = 0$. By following the same reasoning in OA, we can find the controls as,

$$\begin{aligned} u_r^{DTG,k} &= u_r^{DTG,k-1} + (1/dt) \cdot \left(\left(1 + dt \cdot D_r^{DTG}\right) \cdot e_r^{DTG,k} - e_r^{DTG,k-1} \right) \\ u_\phi^{DTG,k} &= u_\phi^{DTG,k-1} + (1/dt) \cdot \left(\left(1 + dt \cdot D_\phi^{DTG}\right) \cdot e_\phi^{DTG,k} - e_\phi^{DTG,k-1} \right) \end{aligned} \tag{18}$$

and the desired change in the current orientation of the robot,

$$\Delta\phi^{DTG} = \phi - \tan^{-1}\left(-\frac{u_r^{DTG}}{u_\phi^{DTG}} \right) \tag{19}$$

2) Velocity Control

The control described above does not involve any velocity generation. Let v^{DTG} be the scalar desired velocity that should be linearly 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 is

$$v^{DTG} = \min\left(a \cdot t, v_{\max} \sqrt{2 \cdot a \cdot d}\right) \tag{20}$$

where d is the distance to the goal point. [9].

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 and must be combined such that both requests are partially fulfilled. For this purpose, a behavior arbitration layer calculating the weighted sum of $\Delta\phi^{DTG}$ and $\Delta\phi^{OA}$ is proposed. Used non-constant weights are calculated from geometry.

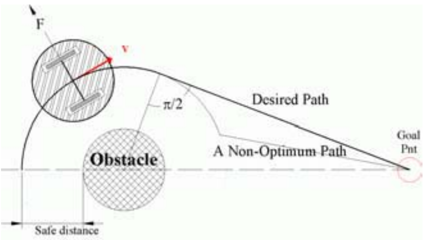

Fig. 3: Optimum and non-optimum path example for an agent while avoiding an obstacle.

Observing an idealized motion of the agent while avoiding an obstacle (Fig. 3), we can see that when the angle between F_{obs} and v is close to π, OA must gain importance and whenever this angle is greater or equal to $\pi/2$ collision has low probability and the DTG must gain importance;

$$\phi^{ref} = \phi + A^2 \cdot \Delta\phi^{OA} + B^2 \cdot \Delta\phi^{DTG} \tag{21}$$

where ϕ is the actual orientation of the robot and A and B are the complimentary weights $A + B = 1$ in the summation. They are both used as square to increase smoothness of ϕ^{ref} and are derived using θ, the angle between v and F_{obs}:

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

The output of this layer is the reference velocity $v^{ref} = v^{DTG}$ 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, which can be used for transmission of commands to the robot and transmission of collected data by the robot. Moreover, communication can be used in multi-robot collaboration where small time delays due to the transmission are not important.

IV. SIMULATIONS, EXPERIMENTS AND RESULTS

A. Experiment: Stationary Obstacles

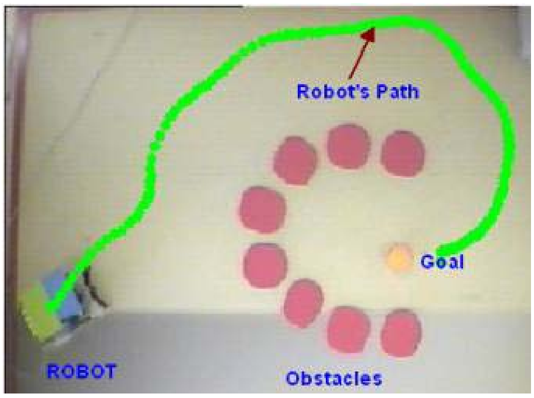

Fig. 4: Avoidance of stationary obstacles.

Stationary obstacles are placed in the environment together with one agent (Fig. 4). We can observe smooth and safe navigation through obstacles.

In this experiment, we can see the work done by the behavior arbitration: first, the agent was moving toward the obstacle. Then the OA layer influenced the robot in such a way that, the robot moved around it. At a later point, where obstacle is not between the agent and the goal point anymore, the behavior arbitration inhibited the output of the OA layer.

B. *Experiment: Stationary Obstacles*

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

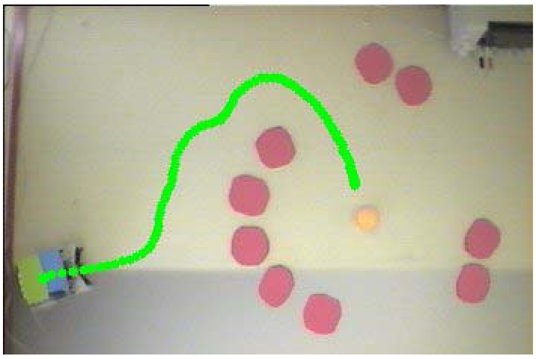

Fig. 5: Avoidance of stationary obstacles.

C. *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. 6, an agent is told to move from point S to point T.

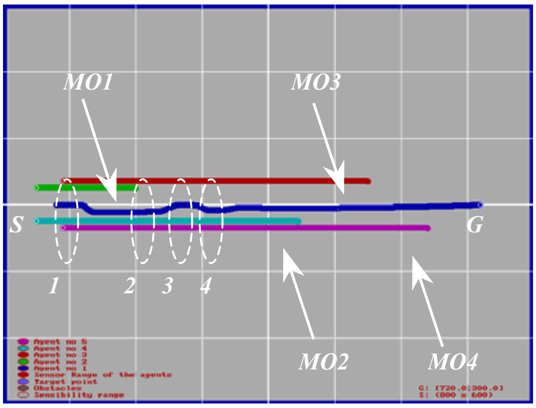

Fig. 6: Avoidance of moving obstacles.

First confrontation happened with MO1 (circled area 1). The agent reacted 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.

D. *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. 7).

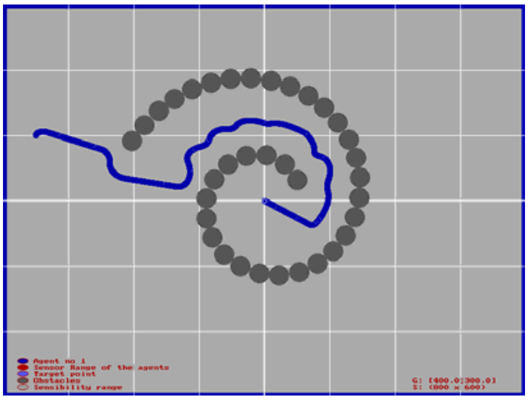

Fig. 7: A robot is trying to move toward the inner point of spiral.

V. CONCLUSION

In this work, we suggested a new approach for the simplest configuration of a mobile robot. This configuration includes OA and DTG implemented as parallel layers and a behavior arbitration layer for fusing those two behaviors. 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 in parallel to the control to augment richness of the behaviors observed.

Proposed control is tested on simulations and real robot. 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] S. Yannier, "Realization of Reactive Control for Multi Purpose Mobile

[9] 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.