International Journal of Industrial Electronics, Control and Optimization © 2019 IECO

Vol. 2, No. 4, pp. 297-304, Oct (2019)

Robot Action Space of Tractable Subsumption Architecture

Farnaz Sabahi[1,†]

[1] Department of Electrical Engineering, Faculty of Electrical and Computer Engineering, Urmia University, Urmia, Iran.

ABSTRACT

In this article, a new hybrid feedback system is introduced, which integrates the behavior- based planning by reactive agent-based control scheme through subsumption architecture. At first, subsumption protocol studies the interactions of the robot with its environment which cover problems including translating of agent action into an outcome, interactions with the environment, and cooperative actions. Second considers deliberative behavior given the prevailing protocol. It determines the best and quickest response for each agent and tunes the actions based on an objective function obtained by a leader agent. More specifically, tasks are arranged as a hierarchy, where the high-level task is obstacle avoidance. Conflicting lower level tasks are removed by the leader agent decisions. Indeed, the leader agent can adjust the priority of all action to provide optimal behavior. In other words, our new agent-based method optimizes the subsumption architecture by producing an approximating objective function that made not only behaviors but also optimization done in the incremental procedure. We also define an emergency avoidance factor that made higher speed still stable and better interaction of robot in the presence of obstacles. For obstacles avoidance, the leader agent projects a plane to investigate the space ahead and continues. Finally, the leader agent makes a basic stand by task sharing behaviors in a decentralized manner using subsumption architecture to draw an optimal path. Simulation results show that although the proposed approach has little knowledge about the unexpected and ad-hoc situation in the robot's environment, it can provide suitable performance.

Article Info

Keywords:
Agent, Obstacle Avoidance, Robot, Subsumption

Article History:
Received 2018-07-31
Accepted 2019-04-01

I. INTRODUCTION

To reach a goal, designing a controller is the most important part. However, the repetitive manner of rules in control systems leads to complex behavior. Achievements in the field of complex systems in general and control systems, in particular, have led to the emergence of intelligent techniques including agent-based methods [1]-[2]. From these approaches, in [3], an innovative Artificial Potential Field (APF) algorithm finds all possible paths that an enhanced Genetic Algorithm (EGA) is used to improve those initial paths to find the optimal path. Moreover, a membrane evolutionary artificial potential field is proposed in [4] for solving the mobile robot path planning problem. In [5], a biomimetic radar sensor is proposed for autonomous navigation.

However, control schemes based on agent provide a higher degree of flexibility and robustness than the other intelligent methods. When discussing the characteristics of complex systems (of which control systems are an example), there is a factor known as Emergent Functionality. This factor says that the overall functionality is not achieved by the conventional centrally tightly coupled way; however, it is indirectly achieved in a decentralized and distributed manner by the interaction of relatively fundamental components with the environment and among own selves [6]-[7].

†Corresponding Author: f.sabahi@urmia.ac.ir, farnazsabahi@ymail.com
Tel: +98- 44-32752741-,Fax: +98-44 32752746, Urmia University
Faculty of Electrical and Computer Engineering, Urmia University, Urmia, Iran

In the case of agent-oriented software engineering, it is noticed that complexity presents several regularities [8]:

- Having the form of a hierarchy
- Choosing a primitive component of the systems is arbitrary to some degree.
- With existing stable intermediate, complex systems will evolve from simple systems faster.
- Distinguish between the interaction among subsystems and interactions between components within them are possible. It means that complex systems are nearly decomposable.

Also, the agent-oriented approach, on the other hand, is a suitable tool for complex systems due to the following properties [8]:

- The problem space of a complex system can be effectively partitioned by agent-oriented decomposition.
- Key abstraction models of agent-oriented mindset are a natural tool for modeling complex systems
- To deal with the interaction and dependencies in a complex system; agent-oriented philosophy can be used.

Therefore, an agent-based approach helps to decompose complex (control) systems into subsystems with task sharing behaviors in a decentralized manner. The classical control approach is to decompose the task vertically into different sections, typically including perception, modeling, planning, execution which is called SPA (Sense-Plan-Act) [9]. However, one of the best-known architecture of this type system in the robotic domain is the subsumption architecture [10] that provides a powerful tool with defining the layered composition of simple behaviors to build complex robotic system functionality. It is applied to many applications other than robotic. A curve negotiation behavior is also proposed within subsumption architectures [11].

Broock's pioneering 1986 A Robust Layered Control Systems for a Mobile Robot described how the behavior analysis can be applied to robotic systems with putting the whole system into a number of elemental behaviors which work asynchronously and in parallel [10]. In the light of this method, the control problem is thought by a number of horizontal layers instead of classic vertical slices (Fig. 1) Data from sensors to actuators in the subsumption architecture is all single-oriented. The results from behaviors are competed or cooperated to fuse. There is a close relationship among layers and outputs of lower layers which can be controlled by the higher layer.

Brooks decomposed the robot control system into eight layers [10]:

1) Avoiding obstacles
2) Wandering
3) Exploring
4) Building maps
5) Monitoring changes in the environment
6) Identifying objects
7) Executing plans
8) Reasoning about the plan

In the architecture, new levels of competence are layered above the existing set. Every augmented higher layer subsumes the existing lower layers. Layers of a control system built based on each level of competence adding a new layer that becomes a higher level [12].

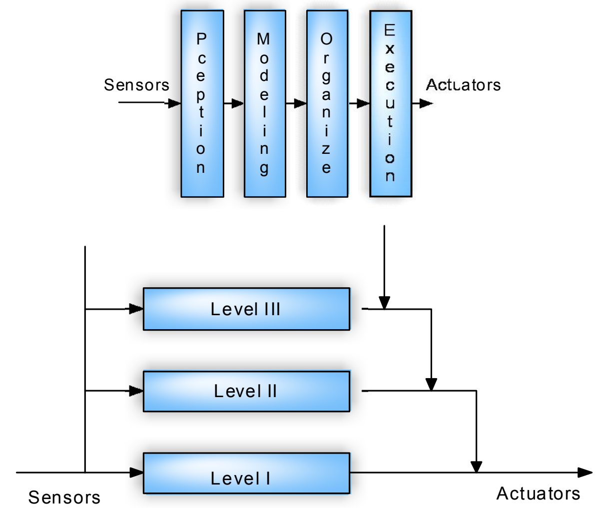

Fig. 1. Decomposition, Vertical (above), Horizontal (bellow).

In fact, higher layers send basic-signals to lower layers to deceive their inputs. As discussed in [13], consider an example of a mobile robot with two motors, one on the right and one on the left, to make it move. The robot moves straight as well as the rotation speeds of two motors match to each other. Feedback of sensors keeps the path straight. If the robot turns right, there is feedback in the right motor to increase its velocity. This method makes the robot turn: When the higher layer makes the robot to move right, a false signal can be given to mimic the robot turning left. Then the feedback of the lower layer adjusts the motor control based on the result; the robot actually turns right, while the lower layer believes it moves straight.

This control method is achieved not only by suppressing the output signal but also by using a signal line into the middle of the lower layers. Losing the modularity is the first shortcoming of this interconnection. The second shortcoming is that the scheme is fixed and cannot be reconfigured for a new situation [13]. This paper studies the latter shortcoming.

Besides, there are other limitations such as [14]:

- Losing important information of the lower layers due to behaviors of the higher level to restrain that of lower level
- Making the internal state of behaviors inaccessible to control layers due to encapsulating of behaviors

- Lacking planning ability which emerged clearly in complicated tasks

Among works which have studied these shortcomings is [15]. Its modifications that have been made to the architecture is in several aspects; for example introducing independent behaviors in each layer, adding behavior manager to fix the priority of each behavior and adding scheduler to control the actuator. Another attempt to overcome the shortcomings is integrating the behavior based planning in Brooks's subsumption architecture [14]. Involving a fuzzy concept is another approach to enhance subsumption architecture. Hierarchical fuzzy control is suggested in behavior-based architecture [16], and also using the fuzzy logic concept for game strategy selection is discussed in [17]. In [18], fuzzy logic is used for robot perception, decision-making, and controlling its speed. In [19], a new method as a generalization of sequential behavior compositions, decisions trees, and the subsumption architecture is proposed. In [20], perception-action learning is proposed in which online learning conducted within a symbolic processing context.

Although the mentioned approaches have provided many merits, seeking and doing optimal behavior is very important. In this paper, we decompose behaviors and then by applying subsumption to the system, through the decisions of the leader agent we seek an optimal still dynamic combination of agents' behaviors in order to do the job optimally. It is coordinated by the leader agent when it constantly monitors the environment. More specifically, this paper proposes an approach that dynamically maximizes the efficiency of the robot motion deployed of Brooks's architecture surveillance and enhances the original robot behavior. In the proposed approach, the leader agent makes the best direction still quickest by considering perception about the probability of success of various strategies in the environment and coordinates the various behaviors. The most important advantage of the proposed approach is its stable, decentralized, and adaptive nature in seeking the optimal path.

The rest of this paper is organized as follows: Section II describes the objects of the proposed agent-based system and then the using the strategy for robot perception and decision-making according to the behavior's hierarchy are introduced along Section III. The theoretical description of the proposed structure is discussed in Section IV. Simulation results are provided in Section V. Finally, conclusions are drawn in Sections VI.

II. OBJECTS IN THE PROPOSED AGENT-BASED SYSTEM

In the behavior-based control system, behavioral fusion is important as well as each basic behavior. This section put the answer to this question: "How to hierarchy levels of different and independent agents to reach the objective". Here, we have four different behaviors and so there are four agents.

A. *Wandering*

Wandering behavior has the lowest priority. When wandering behavior is done, it can be concluded that other behaviors are inactive in robot motion. As soon as the activation of other behaviors occurs, this behavior is suppressed. Here, it commands to the robot to move in a circular curve to know its environment to reach the goal in the direction of the goal.

B. *Goal Seeking*

The next priority belongs to Goal seeking behavior. In no obstacles placed in the environment, the robot will move towards the goal straightly. The leader agent directly guides this behavior because this agent has the necessary information about the environment. Then based on this agent's instruction, robot changes its orientation towards goal consciously and is conservable about detecting obstacles.

C. *Obstacle Avoidance*

The highest priority belongs to obstacle avoidance behavior. Obstacle avoidance has reactive behavior so that the robot keeps a distance from an obstacle. Obstacle avoidance behavior depends on the leader agent's decision according to the obstacle's distance and obstacle's region. If obstacle avoidance behavior is activated, the robot will act based on leader instruction. The width of the pulse in our schedule is used as a measure of the distance of the goal. Obstacle's region is known for the leader agent, using the sensor data. When the robot is near the obstacle, the leader agent informs it about existence obstacle.

Emergency avoidance factor is added to compensate with the shortage of using only one ultrasonic sensor for obstacle avoidance [15]. Here, it is used as a tool to ignore two near obstacles when there is enough place to pass the robot about freely. But it needs more time for the leader agent to decide if to do it or not. This behavior, when activated, makes the robot to react to the world faster and avoid the obstacle as soon as possible. This behavior is very useful in a dangerous environment. The region computed for the dangerous obstacle is more than other ordinary obstacles to avoid it completely.

D. *Leader Agent*

This agent is the most important agent in our system. The underlying knowledge structure of leader agent behavior is formalized by means of strategies. A strategy is defined as policies f of which tactic is chosen to achieve a particular set of goals. In other words, a strategy is a map from states to tactics [21]. This agent distributes tasks and improves the subsuming structure of the whole system. It has a scheduler for decisions and has the necessary information about the environment including the region of known obstacles and constantly monitors the environment to detect obstacles. Physically speaking, it has this information through various sensors.

III. Proposed Strategy

The three layers in the system come from progressively quantizing important behaviors (Fig 2). The leader agent constantly monitors the environment and typically keeps the monitoring by a state as an ensemble of agents' perception and environment's changes to the special task-dependent recognizers.

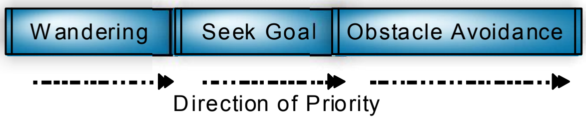

Fig. 2. Priority of Behavior

Fig. 3 show behaviors of the robot in the environment with obstacles, sub goals and goal points in different traces. The leader's decision is an interaction between condition and action. The agent tries to enable the robot to pursue specific path according to a predicted plan. In time, each agent in the system forms a pair of (reaction, objective) according to one-shot action, the incremental procedure of the methodological maxims [10] and coherent with optimizing the objective function. And so, once a time a level of competence may be added for more complex behavior.

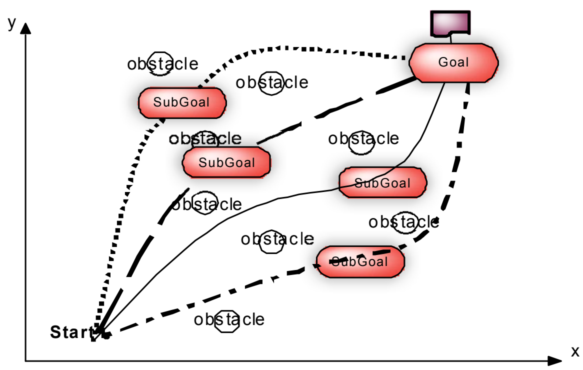

Fig. 3. Different Traces for the Robot

The most important note is the subsumption method is not optimal. Subsumption architecture is an approach to program a reactive agent. This agent shows the behavior of acting without thinking. So, this paper proposes a new agent-based method to optimize the architecture by approximating an objective function drawn by the leader agent. At first we optimize every behavior in its layer not the whole behavior which has done according to the schedule. In other words, there is a priority in the optimization of behavior (Fig. 4). In fact, not only behaviors are done in the incremental procedure of the methodological maxims, but also optimization is done that way too. But between optimality and behavior, the latter has a higher priority. In this way, when there is a situation, in which we are near an obstacle in optimizing goal or we far away goal in optimizing an obstacle avoidance, optimizing opens the door for doing the special behavior (Fig.5).

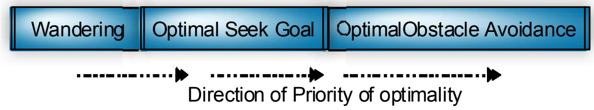

Fig. 4. Priority of Optimality

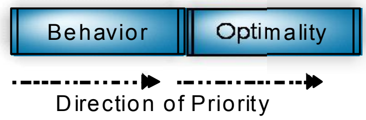

Fig. 5. Priority Between Optimality and behavior

Finding the functions for optimization is owing to approximating dynamic of the environment. There is an assumption here that the leader agent has the ability (from sensors) of finding the necessary information such as approximate place of obstacles, so it has an overview of the function to be minimized along the subsumption architecture that is used to make the priority of the whole system (Fig 5). In Figure 5 block subsumption has valuable information that gives to approximation block in order to approximate a function for optimal behavior and to find the priority of optimal action. With giving these valuable preconditions, the leader agent built the triple:

(Precondition, Approximation, Best Action)

The agent has more than one relevant behavior in a given situation, so it faces a decision matrix that its row is the above triple.

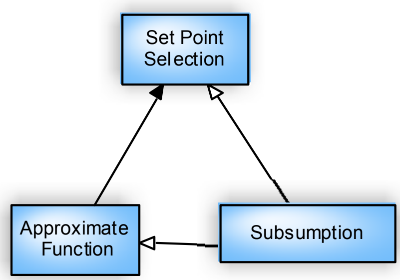

Fig. 6. Proposed Method

To deal with uncertain experience, the agent uses a random factor in its computations to decide about the new position of the robot. Leader agent in its scheduler does this task. The schedule is shown in Fig. 7. The dotted line is somehow done in the same time with the solid line.

Based on the above discussion, each module is described as:

I) Wandering

II) Motion toward the goal (at the same time)

A) Minimize the distance of the goal

III) Avoiding obstacles (at the same time)

A) Maximize the distance of an obstacle

B) Minimize the distance of the goal

C) Choosing suitable subgoals

V) Storing routes

The priority is as discussed in Section II and depicted in Figs. 2,4.

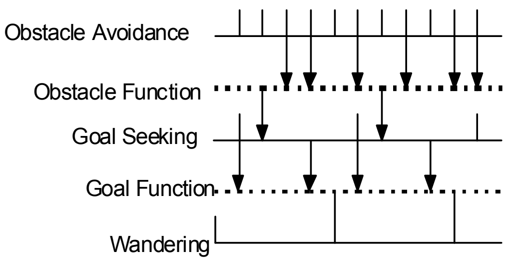

Fig. 7. Scheduler of Leader Agent

IV. THEORETICAL DESCRIPTION OF THE PROPOSED APPROACH

Consider the deterministic affine continuous system representing the environment as follows:

$$\dot{x} = f(x(t)) + g(x(t))u(t) \tag{1}$$

where $x(t)$ represents the state vector of the system and $u(t)$ the control vector. We assume $f(x) + g(x)u$ is Lipchitz on the compact set Ω and there is a signal control that stabilizes the system. The proposed approach for the system ensures that $x(t)$ tracks the desired trajectory $x_d(t)$ optimally. The tracking error can be computed as follows:

$$e(t) = x(t) - x_d(t) \tag{2}$$

The final control signal is provided by the leader agent and is a tactful combination of different strategies provided by agents. We assume the leader agent can provide a signal control $u_d(t)$ that makes $x(t)$ equal to $x_d(t)$. Using Equ. (1), we can obtain $u_d(t)$ as follows:

$$u_d(t) = \frac{1}{g(x_d(t))}(x_d(t) - f(x_d(t))) \tag{3}$$

However, due to the dynamic nature of the environment, $x_d(t)$ is unknown; thus, we show an approximation of it by $\tilde{x}_d(t)$. Note that $\tilde{x}_d(t)$ is updated for every step time α. Therefore, the control signal is also updated and is as follows:

$$\tilde{u}_d(t) = \frac{1}{g(\tilde{x}_d(t))}(\tilde{x}_d(t) - f(\tilde{x}_d(t))) \tag{4}$$

We define the signal control produced by the leader agent as $u(t) = \tilde{u}_d(t) + \sum_{i=1}^{m}(1-\theta_i)u_{e_i}(t)$, where $u_{e_i}(t)$ is the control input feed backed by the agents, θ_i is the priority factor, and m is the number of agents. Now, by substituting, we reach:

$$\begin{aligned}
\dot{e}(t) &= \dot{x}(t) - \dot{\tilde{x}}_d(t) \\
&= f(x(t)) + g(x(t))\Big(\sum_{i=1}^{m}(1-\theta_i)u_{e_i}(t) + \tilde{u}_d(t)\Big) - \dot{\tilde{x}}_d(t) \\
&= f(e(t) + \tilde{x}_d(t)) + g(e(t) + \tilde{x}_d(t))\sum_{i=1}^{m}(1-\theta_i)u_{e_i}(t) \\
&\quad + g(e(t) + \tilde{x}_d(t))\frac{1}{g(\tilde{x}_d(t))}(\tilde{x}_d(t) - f(\tilde{x}_d(t))) - \dot{\tilde{x}}_d(t)
\end{aligned} \tag{5}$$

By appropriate definition, we have,

$$\begin{aligned}
\dot{e}(t) &= f_e(t) + g_e(t)\sum_{i=1}^{m}(1-\theta_i)u_{e_i}(t) \\
g_e(t) &= g(e(t) + \tilde{x}_d(t)) \\
f_e(t) &= f(e(t) + \tilde{x}_d(t)) + g(e(t) + \tilde{x}_d(t))\frac{1}{g(\tilde{x}_d(t))}(\tilde{x}_d(t) - f(\tilde{x}_d(t))) - \dot{\tilde{x}}_d(t)
\end{aligned} \tag{6}$$

To design an optimal path, the leader agent produces a control signal by minimizing the following performance index:

$$J(e(t), u_e(t)) = \int_t^{\infty}\Big[e^T(t)Qe(t) + \sum_{i=1}^{m} u_{e_i}^T(t)(1-\theta_i)^T R(1-\theta_i)u_{e_i}(t)\Big]dt \tag{7}$$

where Q and R are symmetric positive definite. In addition, to guarantee convergence, we define a condition in the form of fothe llowing function:

$$V(x(t)) = \int_t^{\infty} \gamma\Big[x^T(t)Q\,x(t) + \int_0^{u(t)} \varphi^{-T}(t)\Big(\sum_{i=1}^{m} u_{e_i}^{-1`}(t)(1-\theta_i)^{-1}s\Big)\sum_{i=1}^{m}(1-\theta_i)u_{e_i}(t)\,R ds)\Big]dt \tag{8}$$

where $0 \le \gamma \le 1$ and the second part is used to constraint the control input to guarantee convergence, where $\widetilde{U} = diag[\overline{(1-\theta_i)u_i}, 0, \ldots, 0]$, $i = 1, \ldots, m$ and $\varphi(t)$ is a bounded one-to-one monotonic odd function satisfying $\|\varphi(t)\| < 1$. Note that $\bar{u}_i$ is received based on the proper scheduling and subsumption structure. We can define $V^*(x)$, the value function corresponding to the optimal control input as follows:

$$V^*(x) = \min_{u \in \Omega} \left[\int_t^{\infty} (x^T(\tau) Q\, x(\tau) + \int_0^{u(t)} \varphi^{-T}(t) \left(\sum_{i=1}^{m} u_{e_i}^{-1`}(t)(1 - \theta_i)^{-1} s \right) \sum^{m} (1 - \theta_i) u_{e_i}(t)\, R ds) d\tau \right] \quad (9)$$

The leader agent minimizes this function.

V. Simulation Results

The whole simulation is done by Matlab 2018a. The start and goal points are pre-defined while it is tried to not have collisions with obstacles (see Figs. 8-9). The field representation of the environment is shown in Fig. 10.

Two scenarios are done by the leader agent as follow:

I) Without Emergency Avoidance Factor

II) With Emergency Avoidance Factor

Fig. 11 and Fig. 12 show the path found by the proposed method in two scenarios.

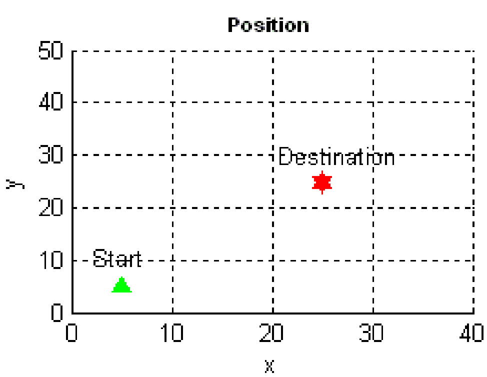

Fig. 8. Start & Target points

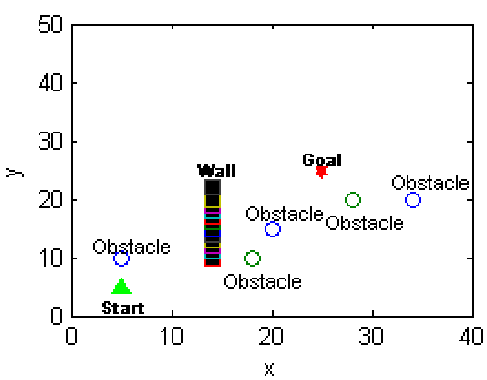

Fig. 9. Obstacles

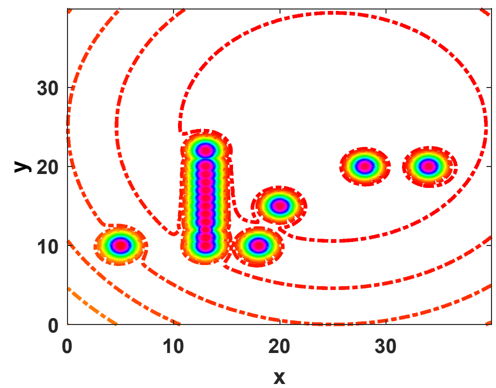

Fig. 10. Gradient Field

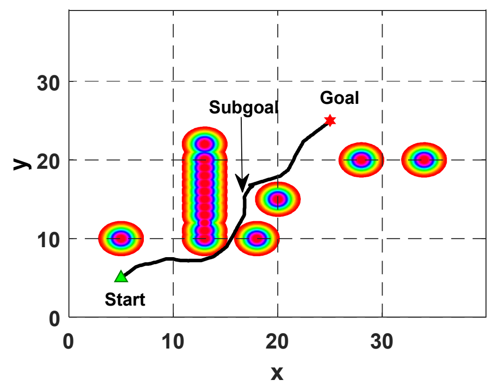

Fig. 11. Obtained Path in First Scenario

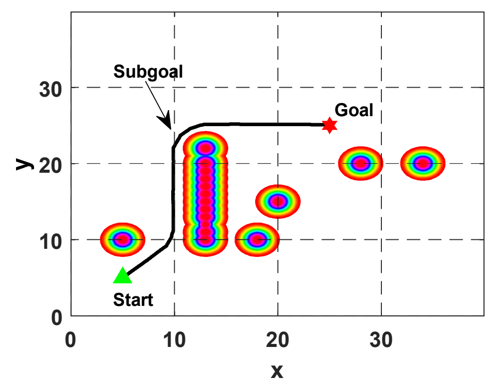

Fig. 12. Obtained Path in Second Scenario

It is obvious from these figures that Emergency Avoidance Factor in the special use in this paper gives insurance that the path is the best yet the most straight forward path with the least obstacles. In fact, the second obstacle plays a major role here than it should. Although in both scenarios the target point is reached, the robot moves more smoothly in Scenario II. In addition, it is the fastest path. This is a promising advantage when used for soccer robots where the speed is very important. Besides, by defining Emergency Avoidance Factor, high speed cannot cause a collision. In Scenario I, as seen in Fig. 11, the robot brake harder near an obstacle to solving the problem, which leads to lower speed while increasing the chance of avoiding the obstacle.

Table I compares the different characteristics of these two scenarios. As clear, the number of actions done in the second scenario is more than that of the first scenario but with less elapsed time. This occurs, because the scenario II uses Emergency Avoidance Factor, and so has more actions to decide about the direction of motion. On the other hand, the subgoals in the middle of the path of the second scenario are less than Scenario I. As seen the amount of acceleration in Scenario II is higher. The standard deviation of velocity for Scenario II is also less than Scenario I, 57 against 81, that verifies the robustness of the proposed approach in Scenario II. This shows the variation around the mean velocity is lower in Scenario II in comparison to Scenario I. Action selection of both scenarios is depicted in Fig. 13 and Fig. 14. In both scenarios, action-selection is continuous. It means the robot continuously acts in pursuit of its agenda. However, in Scenario II curve is steeper than the first scenario. As seen, in the first scenario, changes in actions are smoother.

TABLE I
PATH ON SCENARIOS

Scenario	**Characteristics**			
	Elapsed Time Seconds	***Number of Actions***	***Number of Subgoals***	***Minimum Acceleration***
I	15	300	3	2.5
II	10	350	2	6

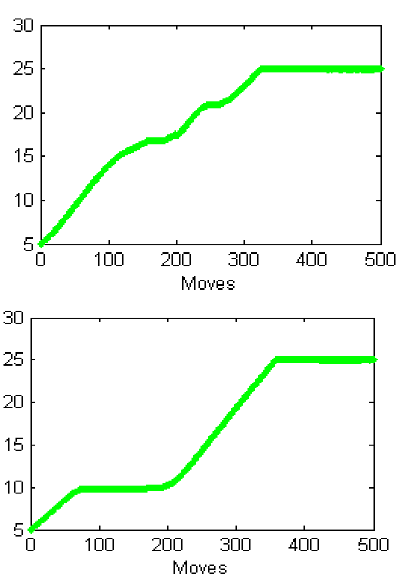

Fig. 13. Action Selection in First path (above) in Second path (bellow)

Fig. 14 provides paths by changing the place of obstacles for both scenarios. As seen, when the obstacle near the wall has enough distance from it, both scenarios provides a path among obstacles. But, when this distance lessons, the path is chosen with enough distance from the obstacles.

The practical interesting of the proposed approach that autonomously plans an optimal path can be applying in dangerous places due to its high ability of obstacle avoidance and exploiting in hospitals due to its high ability of good interaction in high speed.

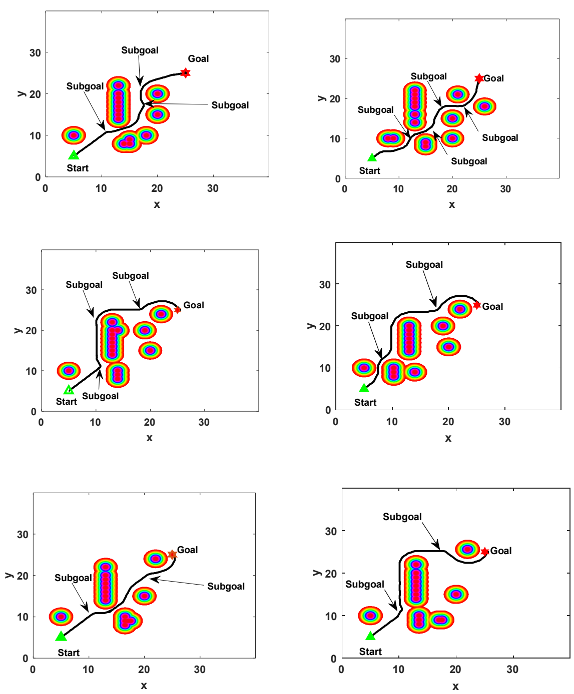

Fig. 14. Scenarions' Path, which it is the same.

VI. Conclusion

Trajectory planning is a crucial issue in the field of automation in general and in the field of robotics in particular. Based on this trend, an approach for mobile robots is proposed and developed. In fact, a novel control architecture which can be viewed as an optimized layered underlying the subsumption architecture is proposed here. In the proposed approach, we define obstacle avoidance, goal seeking, wandering, and the leader agent for each part of the main task in reaching the goal. The leader agent arranges the priority of all optimal behaviors. We also define an emergency avoidance factor that leads to higher speed. In fact, the proposed approach makes a basic stand by aiming to represent path planning by decomposing the control system considering subsumption as a background for the leader agent design in order to draw an optimal or strictly speaking a nearly optimal path. Specifically, the paper emphasizes the importance of optimizing subsumption architecture in doing tasks. In fact, having a stable and debugged core of function behavior in developing subsumption architecture that applied for mobile robots is the main aim of this paper. The proposed approach would be a useful forward step to promote the subsumption-based control structure with optimality manner.

REFERENCES

[1] Y. Zuo, Y. Wu, G. Min, and L. Cui, "Learning-based network path planning for traffic engineering," *Future Generation Computer Systems,* vol. 92, pp. 59-67, 2019.

[2] E. Oland, T. S. Andersen, and R. Kristiansen, "Subsumption architecture applied to flight control using composite rotations," *Automatica,* vol. 69, pp. 195-200, 2016.

[3] M. Nazarahari, E. Khanmirza, and S. Doostie, "Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm," *Expert Systems with Applications,* vol. 115, pp. 106-120, 2019.

[4] U. Orozco-Rosas, O. Montiel, and R. Sepúlveda, "Mobile robot path planning using membrane evolutionary artificial potential field," *Applied Soft Computing,* vol. 77, pp. 236-251, 2019.

[5] G. Schouten and J. Steckel, "A Biomimetic Radar System for Autonomous Navigation," *IEEE Transactions on Robotics,* pp. 1-10, 2019.

[6] R. Brooks, "Intelligence Without Reason," in *Proceedings of 12th Int.Joint Conf. on Artificial Intelligence,* , Sydney, Australia,, 1991, pp. 569–595.

[7] J. A. Meyer and S. W. Wilson, "Towards a Theory of Emergent Functionality," in *From Animals to Animats:Proceedings of the First International Conference on Simulation of Adaptive Behavior*, ed: MIT Press, 1991, pp. 451-461.

[8] N. R. Jennings and S. Bussmann, "Agent-based control systems: Why are they suited to engineering complex systems?," *IEEE Control Systems,* vol. 23, pp. 61-73, 2003.

[9] J. Simpson, C. L. Jacobsen, and M. C. Jadud, "Mobile Robot Control The Subsumption Architecture and occam-pi " in *Communicating Process Architectures, IOS Press*, 2006.

[10] R. Brooks, "A robust layered control system for a mobile robot," *IEEE Journal on Robotics and Automation,* vol. 2, pp. 14-23, 1986.

[11] P. Bosetti, M. D. Lio, and A. Saroldi, "On Curve Negotiation: From Driver Support to Automation," *IEEE Transactions on Intelligent Transportation Systems,* vol. 16, pp. 2082-2093, 2015.

[12] J. E. Arnold, "Experiences with the subsumption architecture," in *Artificial Intelligence Applications, 1989. Proceedings., Fifth Conference on*, 1989, pp. 93-100.

[13] H. Nakashima and I. Noda, "Dynamic subsumption architecture for programming intelligent agents," in *Multi Agent Systems, 1998. Proceedings. International Conference on*, 1998, pp. 190-197.

[14] Y. Yongjie, Z. Qidan, and C. Chengtao, "Hybrid Control Architecture of Mobile Robot Based on Subsumption Architecture," in *Mechatronics and Automation, Proceedings of the 2006 IEEE International Conference on*, 2006, pp. 2168-2172.

[15] C. Tan Tiong and M. N. Mahyuddin, "Implementation of behaviour-based mobile robot for obstacle avoidance using a single ultrasonic sensor," in *Innovative Technologies in Intelligent Systems and Industrial Applications, 2009. CITISIA 2009*, 2009, pp. 244-248.

[16] E. Dehghan and G. Mitchell, "Cooperative Multi-Agent Robot Soccer Team," ed. Albuquerque, NM: NASA ACE Center, Univ. New Mexico,. 1999.

[17] M. Qingchun, Z. Xiaodong, Z. Changjin, X. Jianshe, W. Yulin, W. Tao*, et al.*, "Game strategy based on fuzzy logic for soccer robots," in *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*, 2000, pp. 3758-3763 vol.5.

[18] P. Vadakkepat, M. Ooi Chia, P. Xiao, and L. Tong Heng, "Fuzzy behavior-based control of mobile robots," *IEEE Transactions on Fuzzy Systems,* vol. 12, pp. 559-565, 2004.

[19] M. Colledanchise and P. Ögren, "How Behavior Trees Modularize Hybrid Control Systems and Generalize Sequential Behavior Compositions, the Subsumption Architecture, and Decision Trees," *IEEE Transactions on Robotics,* vol. 33, pp. 372-389, 2017.

[20] D. Windridge, M. Felsberg, and A. Shaukat, "A Framework for Hierarchical Perception–Action Learning Utilizing Fuzzy Reasoning," *IEEE Transactions on Cybernetics,* vol. 43, pp. 155-169, 2013.

[21] S. Russell and P. Norvig *Artificial Intelligence: A Modern Approach (3rd Edition)*: Pearson, 2009.

Farnaz Sabahi received the Ph.D. degree in control engineering from the Department of Electrical Engineering, Ferdowsi University of Mashhad, Mashhad, Iran. She is currently Assistant Professor and the Director of Soft Computing Laboratory with Department of Electrical Engineering, Faculty of Electrical and Computer Engineering, Urmia University, Urmia, Iran. Her current research interests include the areas of fuzzy logic, decision systems, control, soft computing, robotics, and machine learning.