Multi-Target Tracking using Human Teachable Robots

E.M. Saad, M.H. Awadalla, A. M. Hamdy and H. I. Ali

***Abstract*— This paper focuses on developing a team of mobile robots capable of learning via human interaction. A modified Q-learning algorithm incorporating a teacher is proposed. The paper first concentrates on simplifying the Q-learning algorithm to be implemented on small and simple team of robots having limited capabilities of memory and computational power. Second it concentrates on the incorporation of a human teacher in the Q-learning algorithm. Real and simulated experiments using the well-known robot simulator Webots on a proof of context both single and multi-target tracking tasks have been conducted. The achieved results show the success of the proposed algorithm in the overall system performance.

Keywords— Machine learning, Human-robot interaction, Reinforcement learning, Q-learning

I. INTRODUCTION

Human interactive learning for robots is an important field where the learning process itself will be faster and more efficient. This comes from the fact that a human could select the most appropriate actions for the robot. Therefore, it saves the time for selecting the appropriate actions.

Many learning mechanisms for target tracking were proposed [1-5] that vary in complexity and effectiveness. So, developing a learning mechanism that is easy to implement and use is a very important task. This paper introduces such a simple and easy to use learning mechanism while incorporating a human teacher that makes the learning task faster, more efficient and surprising.

This paper focuses on two points regarding target tracking tasks. The first is to develop a learning mechanism that needs limited memory as well as processing power. The second point is to develop a human teachable robot that will considerably reduce the training time. Human interaction helps robots to learn more complicated tasks in a small amount of time.

Manuscript submitted October 17, 2008

All authors are with the Electronics and Communication Engineering Department, University of Helwan, Cairo, Egypt. (corresponding author e-mail: hosam_ib_ali@ yahoo.com).

O. Akanyeti et. al. [6] developed a robot training mechanism using system identification. In this work a mathematical model describing the relation between the robot wheels speed and the sensors readings is built. Then they use an approximation algorithm for estimating the model parameters. The training starts by controlling the robot using a human while recording the sensors readings versus the wheels speeds. This record is used by the approximation algorithm for the estimation of the mathematical model parameters. This work , as seen, needs large processing power and memory for approximation process.

Florin Stoica [7], proposes an abstract finite state machine that could learn the best possible actions based on data received form robot sensors. The work also needs large processing power and memory.

Masaya Yoshikawa [8], proposes a Q-learning algorithm based on Genetic Algorithm and has a hierarchical evolutionary mechanism. The proposed learning algorithm introduces new adaptive action value tables and it enables sharing knowledge among agents effectively. This work suffers from its complexity to be implemented on small mobile robots having small memory and processor.

Kvetoslav Belda [9], proposed a Range-space predictive controller for optimal robot motion. The range-space modification investigated herein takes into account only the limits of the required robot movement and its end point. Such approach can just solve manipulation issues, where the accurate achievement of some trajectory is not important, but the robot has to move through known corridor described by appropriate output range and has to reach some defined end point. This work needs a huge mathematical manipulation dealing with matrices.

The most related work to the theme of this paper includes that of Asadpour [5]. A compact Q-learning algorithm with limited memory and processing power needs was developed. This compact Q-learning algorithm was applied to the task of robot safe wandering.

However, the work of this paper addresses the problems of excessive training effects, environment of training and the false training due to moment of inertia of the robot body.

Andrea et al proposed a Q-learning algorithm with an interactive human teaching [1]. Different scenarios in which a human teaching can be incorporated in Q-learning were studied. They showed that incorporating interactive human teaching had the effect of speeding up the learning in addition to its interesting nature. However, the work of this paper tries to incorporate human teaching with the compact Q-learning algorithm such that small and simple robots could learn more complicated behaviors such as target tracking.

Furthermore, in this paper an infra-red target locator mechanism proposed in [10] is modified such that both angle and distance of a target can be calculated using a small lookup table and multiplication of one byte integers only. The infra-red target locator is necessary for the target tracking behavior.

The rest of the paper is organized as follows. Section 2 formalizes the problem. Section 3 presents the proposed system for multi target tracking compromising an illustration of the hardware platform for the robots, learning of the avoidance behavior, following behavior and proposed multi target tracking system. Simulation and real implementation results are presented in section 4. Finally, section 5 gives the conclusions.

II. PROBLEM FORMULATION

A set of robots (targets) are freely wandering in predefined area. Another set of robots (chasers) tries to track all target robots by following them. Both sets exhibit collection of behaviors (avoidance, follow). These behaviors are to be gained by the robots by learning such that they start with no knowledge describing how to behave and by learning each robot could accomplish its task.

An important assumption is that all robots have limited processing power and memory. Also, they have simple transducers (distance sensors, etc…) to detect their states. Another point to note is that all robots behave independently and take their decision locally.

III. THE PROPOSED SYSTEM FOR MULTI-TARGET TRACKING

The proposed hardware platform used for Multi target tracking task will be introduced firstly. A simplification for the infra-red locator mechanism proposed in [10] is introduced and described in details. The simplification is directed to simplify the calculations needed to compute a target location such that it is suitable to be implemented on a simple micro-controller having limited processing power and memory. The controller for the chaser robots is designed based on the subsumption architecture [11,12]. The avoidance and the follow behaviors will successively be learned by the chaser robot. The target robots exhibit the avoidance behavior only. In the following subsections, the design implementation of each behavior will separately be introduced.

A. Hardware platform

The proposed system is designed and verified using both simulation and real implementation. First, the simulated robots architecture will be explained in details. Second, the real implementation of the robots is illustrated. The following subsections explain both the simulated robots and the real implementation successively.

Simulated Robots

The Webots simulation shell used to design and implement all robots. Webots provides a very good tool for design and simulation of robotics applications. The hardware of the chaser and target robots will be described next in this section.

Chaser Robot

The chaser robot consists of four main parts, the main cylindrical body equipped with differential wheels, an infra-red transmitter, distance sensors for collision avoidance, and target locator. Differential wheels are used to achieve rotation to the right and left, as well as forward and backward movements. The infra-red transmitter sends the robot ID and the ID of the target robot (if captured). The transmitter uses a unique carrier frequency to avoid interference between signals transmitted by different robots. The distance sensors for collision avoidance are used for identifying whether an obstacle is found on a certain distance or not.

The target locator is the same as that proposed in [10]. The criteria for locating a target using that mechanism are by measuring the signal strength for an infra-red transmitter by two receivers, both distance and angle of the target can be calculated. The calculations proposed in [10] are floating point calculations. This in turn needs a heavy processing power capability from the microprocessor that will be used for implementation, so, a simplification for the calculations is introduced.

Fig. 1 illustrates the target locator mechanism. Eqs.(1-3) show how to calculate the target distance and angle with respect to the chaser robot [10]. It is clear that these equations are complicated with respect to small robots having limited processing power. New simplified equations are proposed in this work. Eqs. (4-5) show the new way of computation. As seen, multiplication and division of two integers (after normalization) are only needed after measuring the signal strengths directly from the receivers. A simple equation for calculating the distance (Eq. (4)) of a target is proposed. The distance could be calculated as a function of the two measured signal strengths (by the two receivers) and the fixed separation between the two receivers. This equation could be simply proved using trigonometric calculations. Band pass filters are used for data gathering from all neighboring robots at the same simulation step.

$$L = (1 / S) \wedge 0.5 \quad (1)$$

$$D^2 = (L_1^2 + L_2^2 - 2 * d^2) / 2 \quad (2)$$

$$Cos\ \varphi = (L_1^2 + L_2^2) / (4 * d * D) \quad (3)$$

$$D^2 = ((S_1+S_2)/(S_1 S_2) - 2 * d^2) / 2 \quad (4)$$

$$Cos\ \varphi = ((S_1+S_2)/(S_1 S_2)) / (4 * d * D) \quad (5)$$

where,

L = the distance to the target

S = the received signal strength

L1, L2 are the distances of the target measured by the two receivers

S1, S2 are the signal strengths received from the target as measured by the two receivers

D = the actual distance to the target
d = the separation between the receivers
φ = the angle of the target

Second, the cosine of the target angle is used for the following purpose; reducing the calculation effort needed to calculate the angle itself. Also the equation for calculating the cosine of the target angle is modified so that it depends only on the two measured signal strengths and the computed target distance. The modification is presented in Eq. (5). The square root is needed to calculate the value of D (after calculating D2 in Eq. (4)). A look up table is used to store all squared values for the range (1-250) in 16 bits locations. The size of the look up table is only 250*2 bytes. The square root could be obtained by using binary search for D2 calculated by Eq. (4). An important point to note is that the cosine is a purely floating point number, which is more difficult to handle on simple micro-controllers. So, the numerator of Eq. (5) is firstly multiplied by 100. This gives normalized integer values for the cosine from 0 to 100.

It is clear that the calculation of either the distance of the target or the cosine of the target angle is simplified such that it could be implemented on simple micro-controllers. The distance of the target and the cosine of the target angle will be used to facilitate target tracking.

A light system is added to the chaser robot to enable an observer to detect positive and negative errors in follow angle of the target. Also, it indicates whether the target is too close or too far. These indicators help a teacher to choose the suitable action for the robot to do. The final view of the chaser robot is shown in fig.2.

Target robot

The target robot has a simpler hardware than the chaser. It consists of three main parts, main cylindrical body equipped with differential wheels and an infra-red transmitter, distance sensors for collision avoidance. These parts are the same as those for the chaser robot.

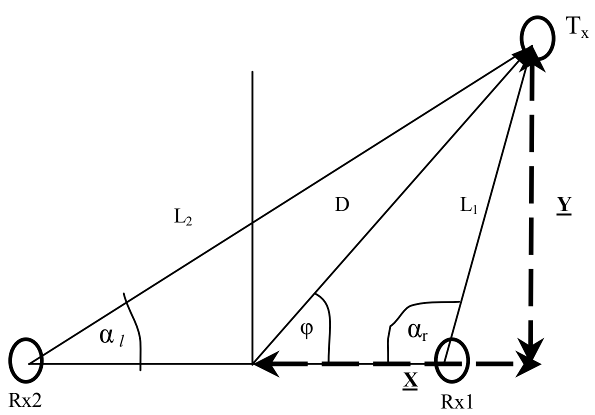

Fig. 1: Calculating the distance and angle of a target robot.

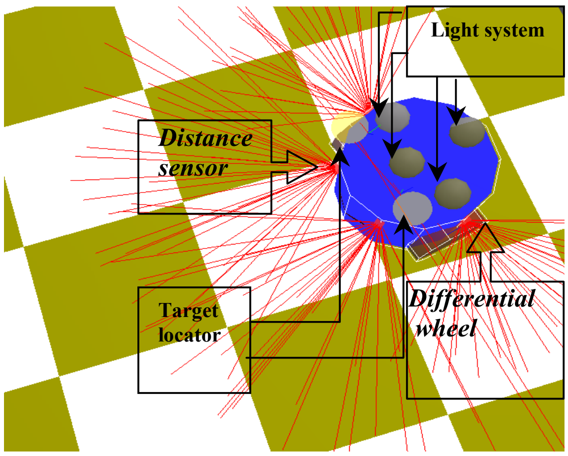

Fig. 2: Final view of the chaser robot

Real Implementation of the Robots

Chaser robot

The real implementation of the chaser robot approximately conforms the simulated robots. Really, some simplifications have been introduced for the target locator. The robot consists of four layers:

1. Differential wheels
2. Distance sensors for collision avoidance
3. Target locator
4. Control board

The differential wheels have been implemented as two bipolar stepper motors. The driver board for each motor has been designed such that by two control bits the motor could be turned forward, backward or stopped. The generator for the required sequences and the oscillator all have been built on the control card. In this way, all the top layers are isolated from the implementation details of the differential wheels layer. Fig. 3 shows this layer.

Fig. 3: Differential wheels layer

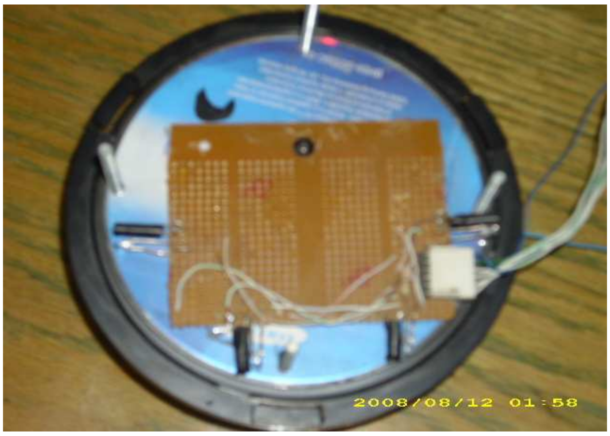

Fig. 4: The avoidance layer

The distance sensors, shown in fig. 4, for collision avoidance have been built using an infra-red LED and infra-red sensor. They are organized as follows; a small radar such that any nearby obstacle will reflect the infra-red radiation that could be received by the sensor. The output of the four sensors shown in fig. 4 are ready to be fed to 4 analog to digital converters (ADC's) built in the control board. By this way, the sensitivity of the robot to obstacles could be varied simply by the firmware loaded into the control board.

The target locator implemented has been simplified with respect to that simulated one. In real implementation, three infrared receivers are spatially located as shown in fig. 5. Three white cones fixed around each receiver collect the infrared radiation. Each receiver output is connected to tone detector chip to identify the existence of the target robot. Now, if the transmitter of the target robot is located on front of the chaser robot then all tone detectors will fire. If the transmitter of the target robot is located on right of the chaser robot then both the middle and the right tone detectors will fire or the right tone detector only. By the same analogy, both the middle and the left tone detectors will fire or the left tone detector only. In this way, the control board could determine the target robot location with respect to the chaser robot.

The distance to the target robot could be determined by measuring the average of signal magnitude received from the three receivers. The average of the received signal magnitude can be obtained using an integrator. Now, the control board could determine how far or close is the target robot by using certain thresholds. Values of these thresholds are determined empirically.

The control board is based on the AVR micro-controller atmega8535. This chip has eight analog to digital converters multiplexed on one of its ports. Additional three 8-bit I/O ports are used for direct LED driving. Also, it supports the RS232 serial communication protocol. Many compilers for the AVR family are available for free. So, a very small and compact control board could be designed and implemented using such chip. Fig. 6 shows the schematics of the control board as well as its real implementation. Many light displays

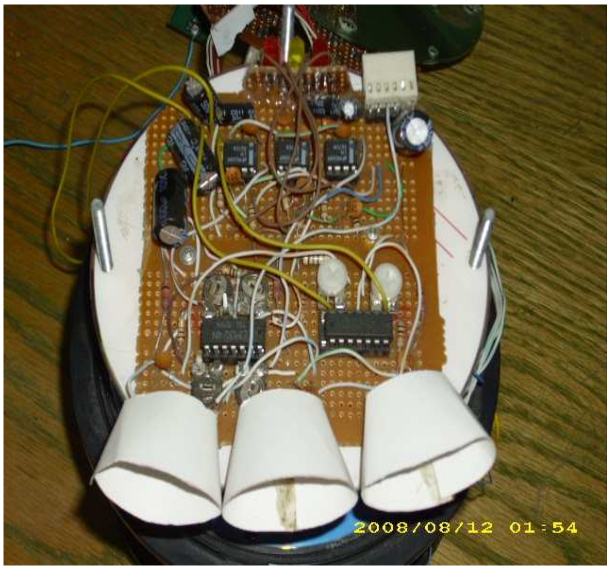

Fig. 5: The follow layer

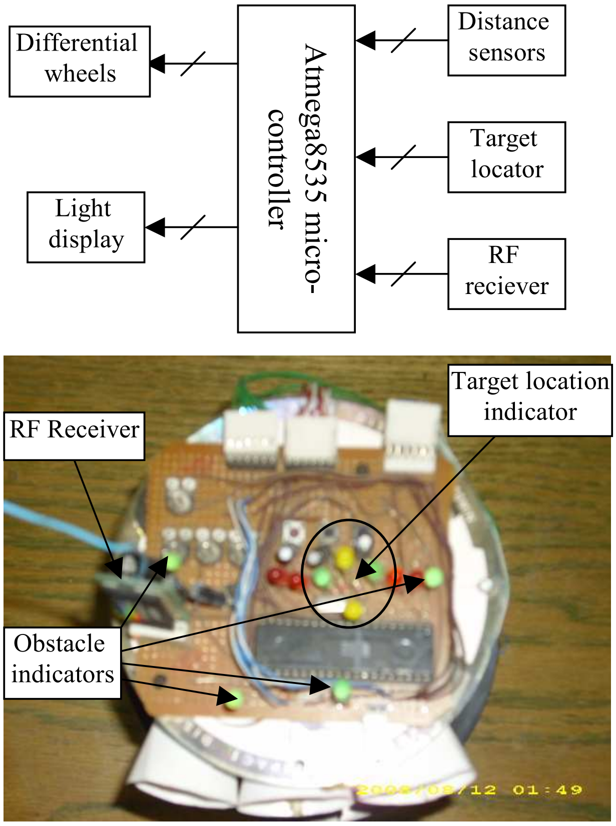

Fig. 6: The control board

have been incorporated to inform the human teacher about the robot states.

Furthermore, an RF remote control has been built such that a human teacher could advise the chaser robot during learning. The RS232 serial communication support in the atmega8535 chip was used for that purpose. Fig. 7 shows the RF remote control module. An image for the whole chaser robot body is shown in fig. 8.

The chaser robot has two modes of operations. The first is the learning mode. During which it receives an advice from a human teacher to follow the target robot and updates the Q-table according to the obtained reward. The second mode tests the learned following behavior. In this test, the robot selects the proper actions independently using the Q-table built during the learning mode. The control board has two buttons to switch among these modes. Extra button has been added to read the Q-table updated by the robot during learning.

Fig. 7: RF remote control module

Fig. 8: the whole chaser robot body

Target robot

The target robot consists of the same layers of the chaser robot except for the target locator layer, which is replaced by an omni-directional infra-red transmitter. The target robot exhibits the avoidance behavior only. Fig. 9 shows the real implementation of the target robot.

B. The Controller Architecture

The controller for each robot is built according to the subsumption architecture. The chaser robot has two behaviors, avoidance and following. If there is no target in its scope, it will exhibit the avoidance behavior. Once it detects a target it starts exhibiting the following behavior. The target robot exhibits the avoidance behavior only.

Both of these behaviors will be learned by the robots as will be illustrated in the following subsections. So, each robot has two modes of operation, the learning mode and the test mode. The target robot is self organized to learn the avoidance behavior using the compact Q-learning [5]. The chaser robot will learn the following behavior using the proposed compact Q-learning with a teacher. Reasons for such difference in learning will be discussed later.

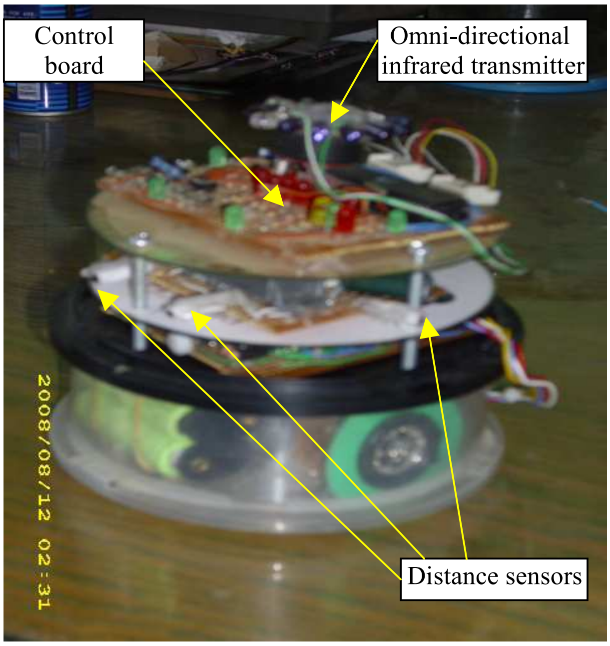

Fig.9: The target robot

Learning the Avoidance Behavior

The Compact Q-Learning algorithm proposed in [5] is used for learning the avoidance behavior. It is very simple and clear to be implemented on the simulator. Such simplification (compaction) for the traditional Q-Learning algorithm is intended to facilitate the real implementation on robots having limited memory and processing power. Algorithm 1 illustrates the compact Q-Learning procedure. The avoidance hardware has been modified so that there are two front distance sensors. This will increase the coverage area around the robot. Q-table will have 16 states and 3 actions, turn right, turn left and move forward. This algorithm used for robot self-training of the avoidance behavior.

Based on the achieved results, a Q-table has been built, by which similar robots could exhibit avoidance behavior. In the Next subsection, a modification of the compact Q-learning algorithm that allows robots to learn a more sophisticated behavior, namely the follow behavior will be presented.

```
While learning do
        Sense current states
        Using roulette wheel selection select an
            action a for the current state s
        Execute a and transit to s'
        Compute the reward r for the action a
        Update Q-table:
            Q(s,a) = Q (s,a) +  (r) + f (s, a, s' )
end while
```

Algorithm 1. Compact Q-Learning

Following Behavior Learning

First of all, the obstacle avoidance capability has been introduced to the chaser and the target robots using the Q-table learned in the previous subsection. Second, the robot state in case of following a target has been presented by two parameters, the error in the follow angle and the error in the follow distance. The first parameter has three possible values, 0, 1 and 2. The value of 0 means that the target is on an angle smaller than the minimum allowed deviation. The value 2 means that the target is on an angle larger than the maximum allowed deviation. The value of 1 means no error in the angle of follow.

By the same way, the second parameter is defined such that it has three possible values 0, 1, and 2. Using these two parameters for presenting the robot state, 9 possible states in case of a chaser following a target will exist. Four actions are chosen for the chaser robot, turn right, turn left, move forward and stop. So, the Q-table for the follow behavior will be presented by a 9*4 two-dimensional array.

The first problem faced in learning the following behavior is how to be assured that the target will be in the visible area of the chaser most of the time. This comes from the fact that a target must be visible by the chaser to allow computing errors in both the angle and the distance of following. During the learning phase, the two robots randomly meet together in small time bursts. These small time bursts were not enough for the chaser to learn. Also, due to the random selection of actions during the learning phase, the robot may forget what it has learned in the previous small time burst. So, self-learning of the following behavior by the traditional compact Q-Learning will be very difficult and time consuming. The solution is to add a teacher, which considerably reduces the time needed for learning. This idea is first proposed in [1]. Andrea et. al proposed a framework for incorporating the human teaching behavior in the traditional Q-learning. They give variety of scenarios showing how benefits could be taken from the human teaching behavior.

In this paper, the human teaching role is reduced to the selection of the proper action for the robot according to the robot state. The light system mounted on top of the chaser robot enables the teacher to detect the state as previously mentioned. Many experiments have been conducted using a human teacher to train a chaser to follow a wandering robot. From these experiments, many observations have been noticed. Based on these observations, a modified version of the compact Q learning algorithm [5] is proposed.

Moment of inertia of the robots

Due to the moment of inertia of the robots, they start shaking especially on a sudden stop. This situation leads to incorrect punishment for the selected action. To clarify this, consider the situation where the robot is too close, a stop action is the best choice. But due to robot shaking after a sudden stop, the robot successively enters both the desired and the undesired states for some time. Also, the same situation happens on most of transitions from desired to undesired state and vice versa.

The proposed solution is to use two thresholds for detecting that the target is too close or too far. The idea is taken from electronics science, taking a benefit from the noise immunity nature of the well-known Schmitt trigger comparator.

New reward function

The reward function used in [5] does not account for the state weight. The state weight is defined as the amount of goodness of the state. As an example, if the target robot is on wrong angle and also too far from the chaser, then the chaser receives a punishment even if it corrects the angle error only. But a deep look to a learning process implies some sort of reward for this partially correct state. Otherwise, the robot will not learn how to behave when it has two kinds of errors at the same time. So a new reward function based on the so-called state weight is proposed. This weight is computed from the state of the robot, previously illustrated. A weight for each parameter of the robot state is proposed, and then these weights are added together. This is further illustrated below:

$$P_\theta = \begin{cases} 1 & \theta_{target} >= \theta_{max.} \\ 0 & \theta_{min.} < \theta_{target} < \theta_{max.} \\ -1 & \theta_{target} <= \theta_{min.} \end{cases}$$

$$P_d = \begin{cases} 1 & D_{target} >= D_{max} \\ 0 & D_{min} < D_{target} < D_{max} \\ -1 & D_{target} >= D_{min} \end{cases}$$

state weight = abs (P_θ) + abs (P_d)

Where,

θ_{target} = angle of the target as seen by the chaser.

D_{target} = distance of the target as seen by the chaser.

P_θ = the weight of the angle parameter

P_d = the weight of the distance parameter

The reward r is computed as follows:

$$r = \begin{cases} 2 & \textit{if state weight } = 0 \\ 1 & \textit{if state weight } = 1 \\ -1 & \textit{otherwise} \end{cases}$$

Training environment

Another problem appeared during the training experiments, not all states of the Q-table occurred, so the robot does not learn how to behave in these states. This comes from the fact that the training environment enforces the robot to encounter a subset of all possible states. So, a special environment has been designed for training, in which the robot is enforced to encounter all possible errors during training phase and hence enters all possible states.

Excessive training

The final problem is the excessive training. The usage of the compact Q-learning algorithm proposed in [5] leads to a situation where two or more actions for some states had the same maximum value of 240 in the Q-table entries. This occurs due to a false training that comes from the moment of inertia of the robots, which has been reduced and not eliminated, as discussed previously. So, a modification for the compact Q-learning algorithm is introduced such that even with excessive training the same ratio is still maintained between the weights of the actions for the same state in the Q-table. The modification is to subtract any positive reward added to a Q-value for an action over 240 (maximum allowed Q-value) from the Q-values of the other actions of the same state. This will maintain the same ratio between the Q-values for actions of a given state even if the robot was trained for large time. The proposed algorithm for learning the following behavior after all the previous modifications is presented by algorithm 2.

While learning do
    Sense current state s and inform the human via a light system
    Wait for the human action a
    Execute a and transit to s'
    Compute the reward r for the action a
    Update Q-table:
        $Q(s,a) = Q(s,a) + (r) + f(s, a, s')$
        If $Q(s,a) > 240$ then
            $\forall\, a', a' \neq a$ let:
            $Q(s,a') = Q(s,a') - (Q(s,a) - 240)$
end while

Algorithm 2. Compact Q-Learning with a teacher for the follow behavior

Multi-Target Tracking

A more complicated task, namely multi-target tracking, in which a team of robots (chasers) try to track and capture another team of robots (targets) is addressed. To do this, a team of four chasing robots and four target robots have been built on the Webots simulation shell. The target robots exhibit the avoidance behavior only to wander in the environment. The chasing robots exhibit both following and avoidance behaviors. The coordination between these two behaviors is performed according to the subsumption architecture. Once a target has been tracked by one of the chasers, a message is sent by the chaser to prevent its colleagues from chasing its captive target. The results of multi-target tracking are shown in section 4.

IV. RESULTS

The proposed approaches have been tested using simulation and real implementation. The results of simulation and real experiments are illustrated in subsection 4.1 and 4.2 respectively. The achieved results of both simulated and real experiments clarify the robustness of the proposed work.

A. Simulation results

Simulations of the proposed work have been implemented in four stages. These stages are illustrated successively in the following paragraphs.

Learning avoidance behavior by one robot

The robot proposed for simulation in subsection 3.1 has been used for testing the compact Q-learning algorithm introduced in subsection 3.2.1. The learned robot is then tested in the same environment. Fig. 10 shows the track of the learned robot during 3 minutes test of a wandering task.

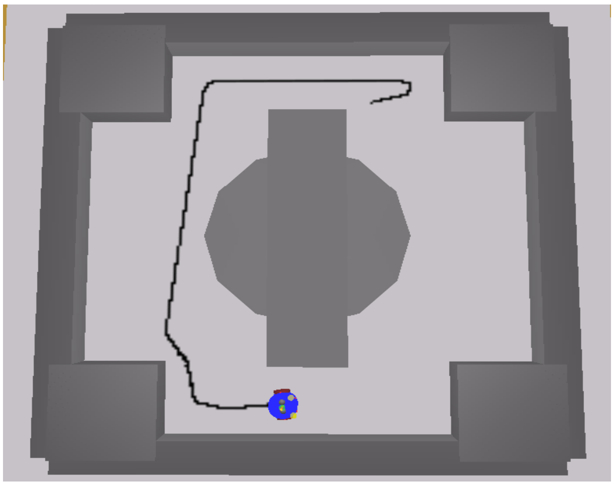

Fig. 10: The track of the learned robot during 3 minutes test

Learning avoidance behavior by four robots

In another test for the compact Q-learning algorithm, an environment having four robots has been created using the Webots simulation shell. The four robots are given 16 minutes to learn the avoidance behavior. Fig. 11 shows records for rewards obtained by each robot. A test has been made for the learned behavior for the four robots. They are allowed to use the learned Q-table for exhibiting the avoidance behavior. The tracks of the robots are marked for one minute long as shown in fig. 12. The results show that the compact Q-leaning algorithm is also suitable for multi robot environment.

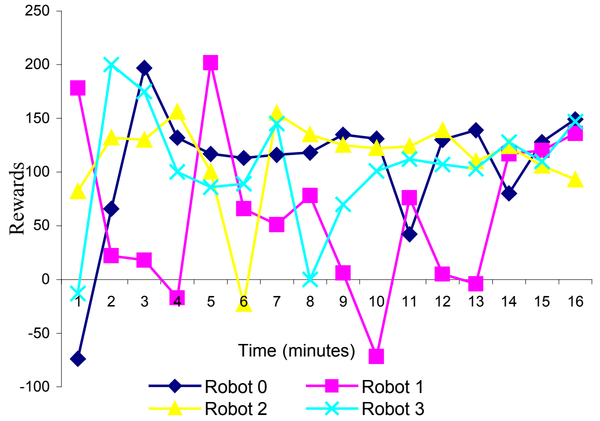

Fig. 11: Received rewards during training of avoidance behaviors for four robots

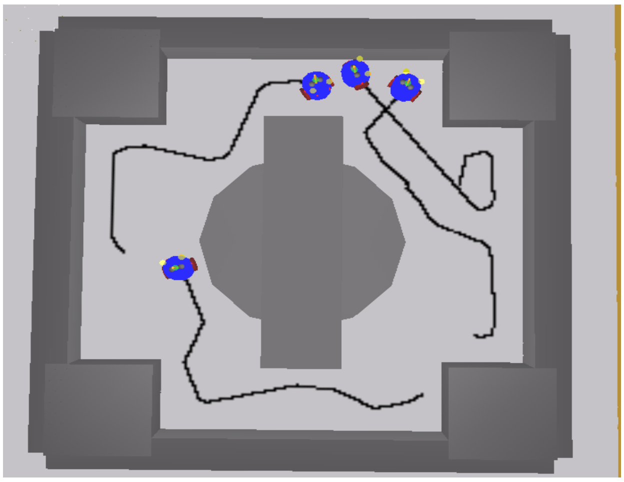

Fig. 12: The tracks of the robots marked during one minute test

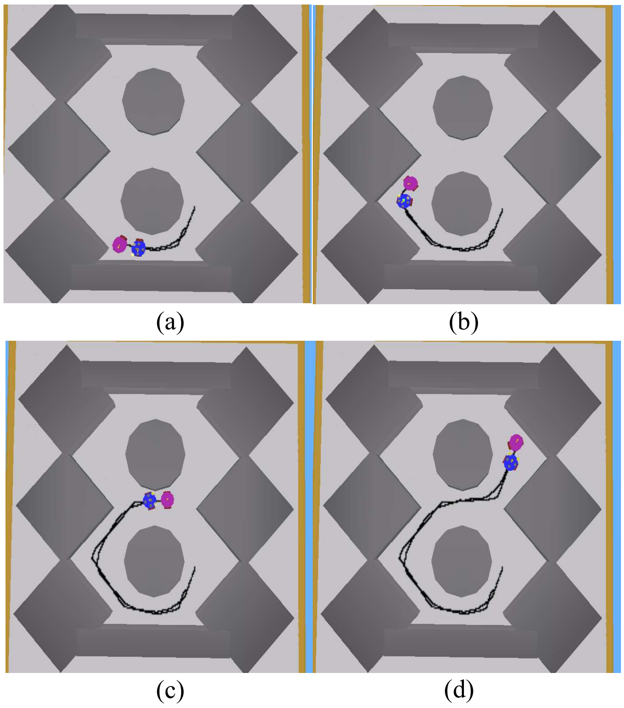

Fig. 13: Successive snapshots during a 2 minutes test for the learned follow behavior

Table I
Time needed to track all targets for six iterations

Iteration	Time (seconds)
1	62
2	10
3	43
4	31
5	51
6	65

Learning the follow behavior

The proposed algorithm for compact Q-learning with a teacher has been tested using the Webots simulator. The chaser robot is trained for 10 minutes via a human teacher. After training, a Q-table representing the follow behavior is obtained. Finally, a test has been performed for the trained robot to follow a wandering target robot. Figs. (13.a - 13.d) show successive snapshots for a two minutes test. For the sake of clarity, a pen has been attached for each robot such that a record of the path will be obtained. The lines drawn in fig. 13 represent the path of the chaser and target robots. It is clear that the two lines are nearly coinciding.

Multi-target tracking task

The task of multi-target tracking presented in section 3.2.3 is implemented and tested using the Webots. A set of experiments has been conducted to examine the system. In each experiment, both chasers and targets are distributed randomly in the environment. The time needed in each experiment to capture all target robots is recorded in table 1.

B. Real Implementation Results

Real implementation of the proposed work has been done in two stages. In the first stage, a single robot is trained for the avoidance behavior then a test is made for that behavior. In the second stage, the chaser robot introduced in subsection 3.1.2 has learned the follow behavior with the aid of a human teacher. The task of multi-target tracking has no real implementation since it needs a large effort for implementing eight robots, also testing the following behavior by two robots will clarify the robustness of the system. These stages are illustrated successively in the following paragraphs.

Learning avoidance behavior by one robot

As mentioned above, the target robot is self organized for learning the avoidance behavior in an environment with no obstacles. Then, a 2 minutes period test is performed for the robot. For the sake of clarity, a pen has been attached to the robot body to record the robot path. Fig. 14 shows successive snapshots for the robot path. It is clear how safe is the wandering trip of the learned robot.

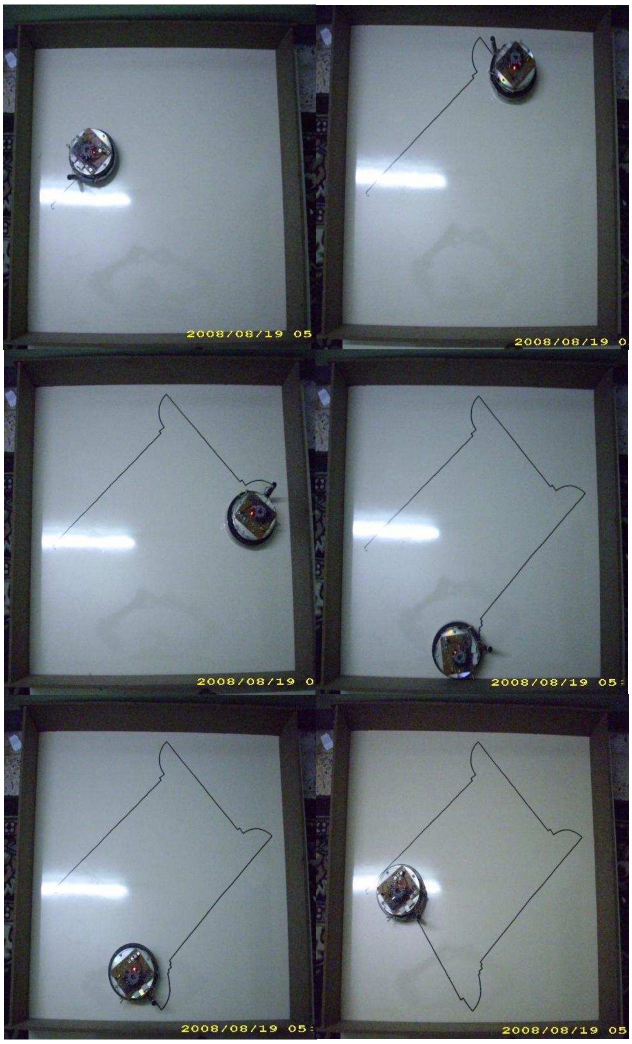

Fig. 14: Robot path during the test of the avoidance behavior

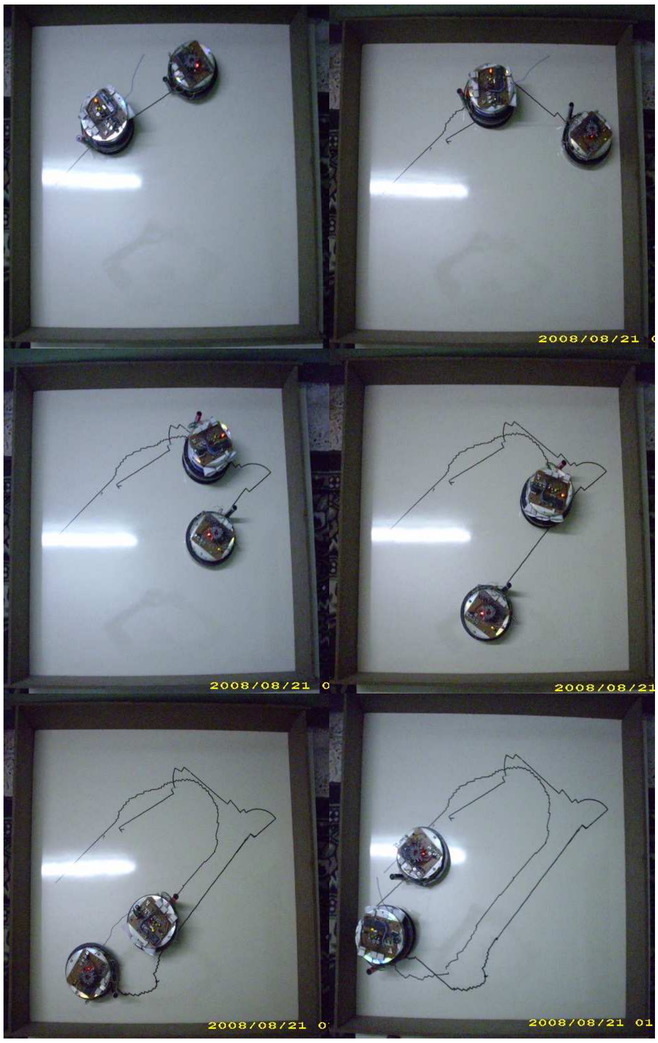

Fig. 15: target and chaser paths during a test for the follow behavior

Learning the following behavior

The learning environment has been set as follows. First, the target robot is left for wandering in the training environment. The target robot transmits a unique tone through its omni-directional infra-red transmitter. Second, the chaser robot has been adjusted to the learning mode and placed in the training environment. Third, a human teacher used the RF remote control module for sending the advice to the chaser robot. The human teacher could monitor the chaser robot state by the visual indicators placed on top of the robot. These visual indicators indicate the location of the target robot as seen by the chaser robot. Then the human teacher could select the appropriate action and sends it to the chaser robot using the RF remote control module.

After 15 minutes of human teaching, the chaser robot is switched to testing mode. Again, a pen has been attached to both robots such that a record of the robots paths could be obtained. Fig. 15 shows these records. As obviously seen in this figure, the two paths are similar.

V. CONCLUSIONS

The paper proposes a system in which a robot could learn behaviors from a human teacher. Therefore, more sophisticated behaviors can be easily learned by a robot. A modified compact Q-learning algorithm is proposed such that it could be implemented on small robots having limited memory and processing power. For the same reason, a simplified target locator mechanism is proposed. The paper introduced a new reward function that differentiates between relative goodness of the states rather than pure punishment or reward. This allows the robot to transit from one state to another better state and finally reaches the best state. The modified learning algorithm accounts for the problems that occur in excessive learning. The effects of the training environment and the nature of the robot mechanical design are also studied and resolved. Finally, a controller is designed and tested based on the learned behaviors (avoidance and follow) suitable for multi target tracking purposes. Verification of the proposed work is done using both simulation and real implementation. The future work includes the real implementation of the multi-target tracking task.

REFERENCES

[1] Andrea L. Thomaz, Cynthia Breazeal, Teachable robots: Understanding human teaching behavior to build more effective robot learners. Artificial Intelligence Vol. 172, 2008, pp. 716–737.

[2] Ching-Wen Ma_, Ching-Cheng Teng, tracking a near-field moving target using fuzzy neural networks. Fuzzy Sets and Systems Vol. 110, 2000, pp. 365-377.

[3] Ying Gao et al, Multi-scale receptive field neural networks for object tracking, International Congress Series, Vol. 1291, 2006, pp. 229– 232.

[4] Ilke Turkmen et al, Genetic tracker with neural network for single and multiple target tracking, Neurocomputing, Vol. 69, 2006, pp. 2309–2319.

[5] Masoud Asadpour, Roland Siegwart, Compact *Q*-learning optimized for micro-robots with processing and memory constraints, Robotics and Autonomous Systems 48 (2004) 49–61.

[6] O. Akanyeti, U. Nehmzowa, S.A. Billings b, Robot training using system identification, Robotics and Autonomous Systems, Vol. 56, pp. 1027-1041, 2008.

[7] Florin Stoica, "An executable model for an Intelligent Vehicle Control System", International Journal of Computers, Issue 4, Volume 1, pp. 325-332, 2007.

[8] Masaya Yoshikawa, Takeshi Kihira, Hidekazu Terai, Q-learning based on hierarchical evolutionary mechanism, WSEAS Transactions on Systems and Control. Vol. 3, Issue 3, pp. 219-228, 2008.

[9] Kvetoslav Belda, Josef B.hm, "Range-space predictive control for optimal robot motion, International Journal of Circuits Systems and signal processing, Issue 1, Vol. 1, pp. 1-7, 2007.

[10] E. M. Saad, M. H. Awadalla, A. M. Hamdy, H. I. Ali, Robot Formations using Local Sensing and Limited Range Vision, In Proc. 25th NRSC' 2008, Egypt.

[11] M. H. Awadalla, Adaptive co-operative mobile robots. Ph.D Thesis, Manufacturing Engineering center, Cardiff University, UK, 2005.

[12] Slim Yannier, Realization of reactive control of multipurpose mobile agent. M.Sc. Thesis, Sabanci University, Istanbul, Turkey, 2002.

E. M. Saad was received the B.Sc degree in electrical (Communication) Eng., Cairo University at 1967, Dipl.-Ing, and Dr-Ing. from Stuttgart University at 1977, 1981 respectively. He is a member of ECS, and EEES. He is currently Prof. of electronic circuits, faculty of Eng. University of Helwan.

Medhat Awadalla has obtained his B.Sc. from Helwan university, 1991 in electronics and communications department and got his M.Sc in the field of reconfigurable computer architecture in 1996 from Helwan university and His PhD from Cradiff university, UK in the filed of mobile robots in 2005. He has a post Doc. at Cradiff university. in 2006 and currently he is working as a lecturer in local and private universities in Egypt as Helwan university, Misr international university., and Misr university. for science and technology.

A. M. Hamdy was born in Cairo in Egypt, on August 17, 1966. He graduated from the Telecommunications and Electronics Department, Faculty of Engineering and Technology of Helwan University, Cairo, Egypt in 1989. He received the M.Sc. degree in computer engineering from the same university in 1996 and the Ph.D. degree from the Faculty of Electrical Engineering, Poznan University of Technology, Poland in 2004.

His special fields of interest include image processing, pattern analysis, and machine vision.

H. I. Ali received his BSc. degree in Electrical Engineering from Faculty of Engineering, Helwan University, Cairo, Egypt in 2000. He received the M.Sc degree in Computer Engineering from the same university in 2004. He worked as an associated lecturer at Faculty of Engineering, Helwan University, Cairo, Egypt, from 2004 till now. He is currently pursuing his PhD study in the area of artificial intelligence at the Department of Computer Engineering, Faculty of Engineering, Helwan University, Cairo, Egypt. His research interests include design of mobile platforms for collective robotics experiments, swarm-based optimization, natural language processing, and most of its disciplines.