Proc. 2016 Int. Electronics Symp., Bali, Indonesia, 2016

Generation of a Socially Aware Behavior of a Guide Robot Using Reinforcement Learning

Bima Sena Bayu Dewantara and Jun Miura
Department of Computer Science and Enginnering, Toyohashi University of Technology, Japan
Email: bima@aisl.cs.tut.ac.jp, miura@tut.ac.jp

***Abstract*—This paper proposes a generation of guiding behaviors of a guide robot under a social force framework that is aware of a human social aspect. This framework is supported by a Q-learning algorithm to optimize the social force parameters to deal with a variety of stimulations. It implies that we let our robot learn by itself by interacting with the environments directly. We named this framework as Q-Learning based Social Force Guiding Model (QL-SFGM). However, let the real robot learn in the real environments under Q-learning framework is difficult, time-consuming, and hazardous. Therefore, in this study, we utilize a realistic simulator, V-Rep, for both training and testing. The simulation results show that our proposed framework is effective to reduce over-reactive behavior of our guide robot so that smoothness, safety, and comfort can be achieved.**

I. Introduction

Human-Robot Interaction (HRI) has become a rapidly growing research area in last two decades. Along with the development of social needs, interaction between human and robot is expanded in a wider range of applications, such as a companion robot [1], person following robot [2], guide robot [3], [4], [5], [6], [7], etc. To support person's activities, an appropriate service provided by a robot is a crucial factor for successful interaction. This service reflects two important things, that are skill and behavior. The skill is used to indicate the ability of a robot to meet the purpose of design, while the behavior is an impact exhibited by a robot when executes its skill to react stimulations. When interacting with a human, a behavior demonstrated by a robot should be acceptable. It means that behaving as smooth as possible by still considering safety is crucial. In this paper, we focus our work on the generation of a socially aware behavior of a guide robot when guiding a target person.

Basically, a guiding task can be decomposed into two smaller tasks: navigation among social environment and coordination with the target person. A conventional robot navigation system usually concerns of the generation of a path (safe and shortest). However, this concern sometimes does not deal with human comfort. In a socially aware robot navigation, human comfort becomes a major concern than just the shortest path. Comfort is a psychological condition that a human feels safe supporting by the absence of a sense of being intimidated or threatened by something. Having an enough space to perform his/her current or upcoming activity is crucial. Hall [8] proposed proxemic interpersonal distance to categorize the spaces around human with respect to social interactions and norms. This space categorization has now become very popular and is adopted in some robot applications [1], [9], [10].

A coordination with the target person is a one-to-one or a private interaction task. In this task, a guide robot monitors the target person activities and measures his/her awareness with respect to the guiding task completion. The main goal of this task is to provide an appropriate action which is proportional to the target partner state, for example, the robot will stop or wait when the target partner left behind, and the robot will speed up when the target partner tends to follow at a closer distance. We propose to use three features which are obtained from the target partner such as a relative distance to the robot, a movement direction, and a head orientation to indicate his/her intention and attention.

In this work, we combined both tasks using the Social Force Model (SFM) [11]. The reasons we used SFM are: (1) this model is specifically designed for observing agent behavior when to interact each other, (2) this model has been adopted and successfully implemented by many researchers who work with socially aware robot applications for example in [1], [9], [10], and (3) this model is flexible, because we can easily combine tasks. However, working with SFM requires some parameters that must be tuned. Some previous works [1], [9], [10] dealt with the tuning of these parameters in advanced by utilizing evolutionary-based optimization approaches, i.e., Genetic Algorithm (GA) [12]. This approach may be practical; however, the optimized parameters are not always in accordance with various conditions, i.e., a robot which is trained for indoor may exhibit strange behavior when operated in an outdoor environment and vice versa. As the consequence, the robot makes an unexpected behavior that is probably dangerous, threatening to the others, and potentially damaging itself.

Therefore, we assume that adaptively tuning the parameters is an effective way to affect the interaction behavior. We perform an online learning by utilizing Q-learning to optimize the parameters, and we call it as Q-learning based Social Force Guiding Model (QL-SFGM). There are two main contributions of this paper: (1) To the best of the authors' knowledge, there are no previous works which exploiting all potential features from a human for handling a guiding task under social force framework. (2) This work is also the first which adjusts all of the SFM parameters adaptively for a mobile guide robot application. Adaptive adjustment of the parameters is effective to reduce an over-reactive behavior of the robot.

The remainder of this paper is organized as follows. We present our social force guiding model in section II. Our proposed behavior learning strategy is described in section III. Section IV shows the experimental results. And finally, the conclusion of our work is delivered in section V.

II. Social Force Guiding Model

A. Social Force Navigation Model

Under Social Force perspective, a guide robot is a mobile robot with mass m which tries to reach a goal with a desired speed v^0 in the desired direction e^0, thereby the desired velocity $\mathbf{v}^0 = v^0 e^0$. The robot tends to adapt its actual velocity $\mathbf{v}$ to the desired velocity within a period of relaxation time τ. Hence, the basic equation of motion of the robot towards a goal is given by the social force term:

$$\mathbf{F}^g = m\frac{\mathbf{v}^0 - \mathbf{v}}{\tau} \tag{1}$$

During its movement from the start position to the goal, the robot will try to keep a distance both from the closest dynamic obstacle, κ, and the closest static obstacle, $\hbar$, by the interaction force $\mathbf{F}^\kappa$ and $\mathbf{F}^\hbar$ respectively, as shown in Fig. 1(a). This behavior is referred to in term of the repulsive effects. Both $\mathbf{F}^\kappa$ and $\mathbf{F}^\hbar$ are the result of a combination of social repulsive force, $\mathbf{f}_{soc}$, and physical repulsive force, $\mathbf{f}_{phy}$. $\mathbf{F}^\kappa$ can be expressed as

$$\mathbf{F}^\kappa = \mathbf{f}^\kappa_{soc} + \mathbf{f}^\kappa_{phy}, \tag{2}$$

$$\mathbf{f}^\kappa_{soc} = k^\kappa \exp\left(\frac{r_{R\kappa} - d_{R\kappa}}{\Psi^\kappa}\right)\mathbf{e}_{R\kappa}\omega, \tag{3}$$

$$\mathbf{f}^\kappa_{phy} = k^\kappa (r_{R\kappa} - d_{R\kappa})\mathbf{e}_{R\kappa}, \tag{4}$$

where k^κ is the magnitude of force w.r.t. a dynamic obstacle. $r_{R\kappa} = r_R + r_\kappa$ is a sum of the robot's radius, r_R, and the dynamic obstacle's radius, r_κ, at an intersection point between their interaction space. $d_{R\kappa}$ is a distance between the robot to the closest dynamic obstacle. Ψ^κ is an effective range of influence of the force w.r.t. a dynamic obstacle. $\mathbf{e}_{R\kappa}$ is a vector indicating the direction from the dynamic obstacle to the robot. When the distance between the robot and the dynamic obstacle, $d_{R\kappa}$, is larger than $r_{R\kappa}$, the force will be ignored.

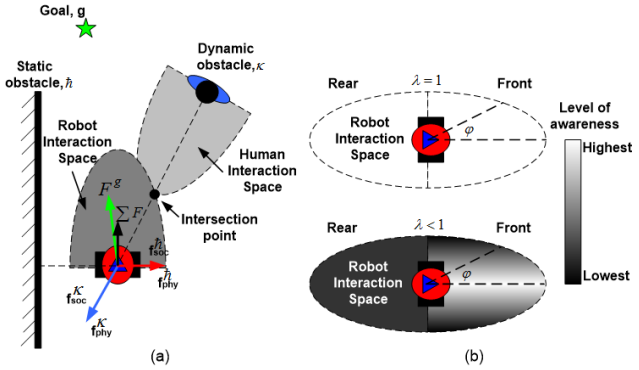

Fig. 1. The social force for modeling robot navigation. (a) An example of behavior of the robot when interacts with the environment, and (b) An anisotropic factor to define a limited awareness of an agent; $\lambda = 1$ means the same level of awareness for all φ, and $\lambda < 1$ means the level of awareness is gradually decreasing depends on φ.

Due to a limited field of view of a human, an anisotropic factor expressed by $\omega = \lambda + 0.5(1-\lambda)(1+\cos(\varphi))$ is introduced into the equation. λ is a parameter which defines an anisotropic influence region that represents the fact that the obstacle in front of an agent are usually more relevant than that located behind it and φ is an angle between obstacle and robot as illustrated in Fig. 1(b). $\mathbf{F}^\hbar$ can be expressed as

$$\mathbf{F}^\hbar = \mathbf{f}^\hbar_{soc} + \mathbf{f}^\hbar_{phy}, \tag{5}$$

$$\mathbf{f}^\hbar_{soc} = k^\hbar \exp\left(\frac{r_R - d_{R\hbar}}{\Psi^\hbar}\right)\mathbf{e}_{R\hbar}\omega, \tag{6}$$

$$\mathbf{f}^\hbar_{phy} = k^\hbar (r_R - d_{R\hbar})\mathbf{e}_{R\hbar}, \tag{7}$$

where $k^\hbar$ is a magnitude of force w.r.t. a static obstacle. r_R is a robot's radius. $d_{R\hbar}$ is a distance between the robot to the closest static obstacle. $\Psi^\hbar$ is an effective range of influence of the force w.r.t. a static obstacle. $\mathbf{e}_{R\hbar}$ is a vector indicating the direction from the dynamic obstacle to the robot. When the distance between the robot and the static obstacle, $d_{R\hbar}$, is larger than r_R, the force will be ignored. We can define a resulting force with respect to the navigation among social environment, $\mathbf{F}^{nav}$, as follows.

$$\mathbf{F}^{nav} = \mathbf{F}^g + \mathbf{F}^\kappa + \mathbf{F}^\hbar. \tag{8}$$

B. Social Force Coordination Model

In the case of a guide robot, the robot navigation is not influenced by interactions with obstacles only, but it is also affected by an interaction with the target partner state. We defined the state of the target partner by using three features: a relative position of the partner, ρ, a movement direction, α, and a head orientation, β.

The human partner's relative position under social force framework is expressed as a combination of a social force and a physical force (both can be repulsive or an attractive) as a function of distance. We adopted a bipolar sigmoid function for this social force. All forces are then expressed as follows.

$$\mathbf{f}^\rho_{soc} = \left[2k^\rho \frac{1}{1+\exp\left(-\frac{d_{R\rho}-\mu_\rho}{\Psi^\rho}\right)} - k^\rho\right]\mathbf{e}_{R\rho}, \tag{9}$$

$$\mathbf{f}^\rho_{phy} = k^\rho (d_{R\rho} - \mu_\rho)\mathbf{e}_{R\rho}, \tag{10}$$

$$\mathbf{F}^\rho = \mathbf{f}^\rho_{soc} + \mathbf{f}^\rho_{phy}, \tag{11}$$

where $\mathbf{F}^\rho$, $\mathbf{f}^\rho_{soc}$, and $\mathbf{f}^\rho_{phy}$ are respectively a total force, a social force and a physical force given by the target partner to the guide robot w.r.t. his/her relative position. k^ρ is a magnitude of force w.r.t. the target partner relative position. $d_{R\rho}$ is a relative distance between the human partner and the guide robot. μ_ρ is a mean value which represents a comfortable distance when

the human partner follows the guide robot. Currently, this mean value is set to 2.2 meters. We conducted a separate study to determine this value by collecting data from six persons. Ψ^{ρ} is an effective range of influence of the force. $\mathbf{e}_{R\rho}$ is a vector indicating the direction from the human partner to the guide robot, and vice versa.

We also formulate the target partner movement direction under social attractive force assumption as follows.

$$\mathbf{F}^{\alpha} = k^{\alpha}(1 - \cos(\alpha + \varphi))\mathbf{e}_{R\rho}, \tag{12}$$

where $\mathbf{F}^{\alpha}$ is an attractive force due to the human movement direction, k^{α} is a magnitude of the force w.r.t. human movement direction, α and φ are a human partner movement direction and a relative direction of the human partner from the guide robot, respectively. The last feature is the human partner's head orientation that can be expressed as follows.

$$\mathbf{F}^{\beta} = k^{\beta}(1 - \cos(\beta + \varphi))\mathbf{e}_{R\rho}, \tag{13}$$

where $\mathbf{F}^{\beta}$ is an attractive force due to the human partner's head orientation, k^{β} is a magnitude of the force w.r.t. human partner's head orientation, β is a human partner's head orientation. Since the human partner movement direction and head orientation only affect to the social related context, we defined them as the social forces only. Therefore, we can define a resulting force with respect to the coordination with the human partner, $\mathbf{F}^{coord}$, used for influencing the motion planning and control of the guide robot as follows.

$$\mathbf{F}^{coord} = \mathbf{F}^{\rho} + \mathbf{F}^{\alpha} + \mathbf{F}^{\beta}. \tag{14}$$

Finally, we can obtain our Social Force Guiding Model (SFGM) by combining both tasks and resulting a guiding force, $\mathbf{F}^{guide}$, by following an expression as follows.

$$\mathbf{F}^{guide} = \mathbf{F}^{nav} + \mathbf{F}^{coord} \tag{15}$$

C. PID Controller

We employ a Proportional-Integral-Derivative (PID) controller as shown in Fig. 2 to reduce bouncing effects exhibited by SFM and to speed up focusing our robot heading to the goal direction. Employing this PID controller can greatly speed up the Q-learning convergence or reducing the number of learning episodes. All of the PID coefficients are also optimized.

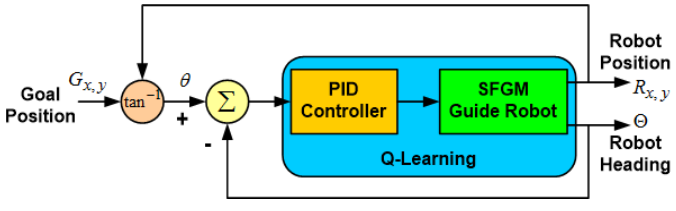

Fig. 2. Block diagram of our proposed social force guiding model.

III. Reinforcement-based Behavior Learning

A. Problem definitions

Under social force perspective, the motion of a guide robot is driven by a velocity as the result of the navigation force and the coordination force at that time step, $\mathbf{F}_t^{guide}$. Obtaining an appropriate $\mathbf{F}_t^{guide}$ for each circumstance is, however, crucial to exhibit a socially acceptable behavior, i.e. smooth and safe. With respect to smooth and safe navigation, a trade-off between keeping direction towards the goal, abilities to avoid collision with obstacles, and keeping coordination with the human partner is a crucial factor to produce a reliable $\mathbf{F}_t^{guide}$. Therefore, we argue that this trade-off problem can be solved using Reinforcement Learning (RL) technique.

B. Q-learning based Parameters Optimization

In general, the RL problem can be formulated as a discrete time, finite state, finite action Markov Decision Process (MDP) [13]. The learning environment can be modeled by a 4-tuple $\{x, a, p, r\}$, where:

- $x \in \mathcal{X}$; $\mathcal{S}$ is a finite set of states.
- $a \in \mathcal{A}$; $\mathcal{A}$ is a set of actions that the agent can perform.
- $p \in \mathcal{P}$; $\mathcal{P} : \mathcal{X} \times \mathcal{A} \rightarrow \Pi(\mathcal{X})$ is a state transition function, where $\Pi(\mathcal{X})$ is a probability distribution over $\mathcal{X}$. $p(x, a, x')$ represents the probability of moving from state x to x' by performing action a.
- $\mathcal{R} : \mathcal{X} \times \mathcal{A} \rightarrow R$ is a scalar reward function.

The goal of the agent in an RL problem is to learn an optimal policy $\Pi^* : \mathcal{X} \rightarrow \mathcal{A}$. We use Q-learning (QL); the most successful method of RL. In Q-learning, the policy is computed using Q-value which is referred to as the state-action value and is updated by $Q(x, a) = Q(x, a) + \eta(r(x, a) + \gamma maxQ(x', a) - Q(x, a))$, where $Q(x, a)$ is a Q value of the state, x, and action, a, pair. η is a learning rate within range (0,1). $r(x, a)$ is a direct reward value for the state-action pair. γ is a discount factor. $maxQ(x', a)$ is the estimated maximum Q-value of the next state.

C. Configuring A State

A state describes a local situation faced by the robot during its travel. The local situation is composed of several features as shown in Fig. 3. We built and specify our own interaction zone by modifying the proxemic interpersonal distance that was introduced by Hall in [8]. In the proxemic interpersonal distance, each distance is specified within a specific shape, e.g., circular or elliptical. Due to simplify the computational problem and the generation of states, we discretized all features using binary segmentation and assigning a constant real value, w, as the identity of each segment as shown in Table. I. The state w.r.t. each feature can be expressed as

$$\begin{aligned} x_1 &= \textstyle\sum_{i=0}^{3} p_i w_i, \\ x_2 &= \textstyle\sum_{i=0}^{2} b_i w_i, \\ x_3 &= \textstyle\sum_{i=0}^{3} f_i w_i, \\ x_4 &= \textstyle\sum_{i=0}^{2} h_i w_i, \\ x_5 &= \textstyle\sum_{i=0}^{4} s_i w_i, \\ x_6 &= \textstyle\sum_{i=0}^{4} d_i w_i, \end{aligned} \tag{16}$$

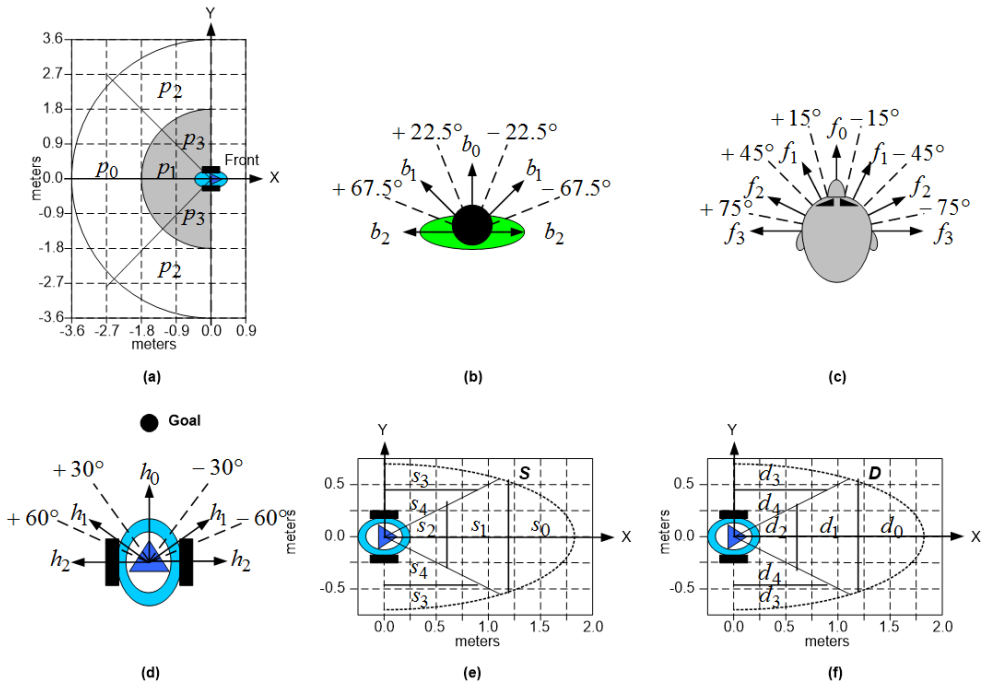

Fig. 3. Features for composing the state: (a) A zone for observing the human partner position, (b) A division of the human partner's movement direction, (c) A division of the human partner's head orientation, (d) A division of the guide robot's heading direction, (e) A zone for observing static obstacles, and (f) A zone for observing dynamic obstacles.

TABLE I. IDENTIFYING REGIONS W.R.T. EACH FEATURE USED IN DEFINING STATES

Feature	w_4	w_3	w_2	w_1	w_0
Partner's position	-	4	3	2	1
Partner's movement direction	-	-	15	10	5
Partner's head orientation	-	80	60	40	20
Robot's heading direction	-	-	200	100	0
Static obstacle	2,400	1,200	900	600	300
Dynamic Obstacle	28,800	14,400	10,800	7,200	3,600

$$x = x_1 + x_2 + x_3 + x_4 + x_5 + x_6, \tag{17}$$

where x_1, x_2, x_3, x_4, x_5, and x_6 are the state of each feature w.r.t. the target partner's position, the target partner's movement direction, the target partner's head orientation, the robot's heading direction, the static obstacle, and the dynamic obstacle, respectively. p, b, f, h, s, and d are codes of fulfilling circumstance of each feature in a binary digit "0" and "1". x is a state which represents the combination of all features.

D. Defining and Selecting an Action

In the MDP framework, action is a set of variables that can be chosen and executed to switch from current state to the other. In our case, an action is defined as a set of social force guiding model parameters, $a = \{k^{\kappa}, \Psi_{R\kappa}, k^{\hbar}, \Psi_{R\hbar}, \lambda, k^{\rho}, \Psi_{R\rho}, k^{\alpha}, k^{\beta}, K_p, K_i, K_d\}$. Taking an action means adjusting those parameters. An action is selected based on its highest probability of a state-action pair, $p(x, a)$, from Q-value normalization as follows.

$$p(x,a) = \begin{cases} 1.0 & \text{if } Q(x,a) = \max Q(x) \\ \frac{Q(x,a) - \min Q(x)}{\max Q(x) - \min Q(x)} & \text{otherwise} \end{cases} \tag{18}$$

E. Reward Value

A reward value is used to update the Q-value of a state-action pair. Referring to section III. A, the main objective of using Q-learning is to solve the trade off problem between those three objectives. We proportionally calculate this reward value using a linear equation for representing the reward function as follows.

$$r = mf(g,h) + c \tag{19}$$

where r is a reward value, m is a constant multiplier or a slope, $f(g,h)$ is a function derived from a specific objective, and c is a constant. A small positive reward value (e.g. +1) is required to keep the action to stay on the list of the best actions when a learner successfully does a job. A negative reward value (e.g. -10) is required to remove the action from the list when a learner failed. If we expect the result of $f(g,h) = [0,1]$ and c is a negative reward value ($c \leftarrow -10$), then we got $m = 11$. By following the formulation, we heuristically defined three objectives as follows.

1) Completing an episode:
$$r_{success} = \begin{cases} +100 & \text{if robot can reach the goal} \\ -10 & \text{if robot fails} \end{cases}.$$
2) Minimizing the robot runs in different direction with the goal:
$r_{\Theta} = 11\cos(\theta_t - \Theta_t) - 10,$

where θ_t and Θ_t are the relative goal direction from the robot's position and the robot's heading at time t, respectively.

3) Controlling the magnitude of forces: the dynamic obstacle, ($r_{f_t^\kappa}$), the static obstacle, ($r_{f_t^\hbar}$), and the human partner force, ($r_{f_t^{\rho\alpha\beta}}$).

$$r_{f^\kappa} = \begin{cases} 11\exp^{(-(F_t^\kappa - F_{max}^g)^2/(2(F_{max}^g)^2))} - 10 \\ \text{if region } \mathbf{D} = d_1, d_2, d_4, d_1 \cup d_4 \text{ or } d_2 \cup d_4 \\ 11\exp^{(-F_t^\kappa/F_{max}^g)} - 10 \\ \text{otherwise} \end{cases}$$

$$r_{f^\hbar} = \begin{cases} 11\exp^{(-(F_t^\hbar - F_{max}^g)^2/(2(F_{max}^g)^2))} - 10 \\ \text{if region } \mathbf{S} = s_1, s_2, s_4, s_1 \cup s_4 \text{ or } s_2 \cup s_4 \\ 11\exp^{(-F_t^\hbar/F_{max}^g)} - 10 \\ \text{otherwise} \end{cases}$$

$$r_{f_t^{\rho\alpha\beta}} = \begin{cases} 11\exp^{(-(F_t^{\rho\alpha\beta}/F_{max}^g)} - 10 \\ \text{if region } \mathbf{P} = p_1 \text{ or } p_3 \\ 1 \\ \text{otherwise} \end{cases}$$

where F_{max}^g is the maximum F^g without repulsive forces, when $\mathbf{v} = 0$.

The direct reward value of each step, r_i, can be expressed as

$$r_i = r_\Theta + r_{f^\kappa} + r_{f^\hbar} + r_{f_t^{\rho\alpha\beta}}. \tag{20}$$

The total reward value, R, after completing one episode is expressed as

$$R = \sum_{i=1}^{N} \gamma^i r_i + r_{success}. \tag{21}$$

where $i = 1, 2, 3, ..., N$, N is the number of steps, and γ is a discount rate applied to the expected maximum Q-value of the next state.

IV. Experimental Results

A. Experimental Platform and Robot Model

We validated our system using the V-Rep simulator [14]. Several simulated environments are built for training and testing purposes. The Social Force Guiding Model (SFGM) and Q-learning (QL) algorithm are implemented using Visual C++ programming that is remotely connected to V-Rep through an API client-server. We used a modified Pioneer P3DX robot model that is equipped with a Hokuyo laser sensor (LRF) in front of the robot and a back-facing camera on the top of the robot. The LRF is used to detect the static and dynamic objects (in this paper, the dynamic object positions are obtained directly from the simulator). The back-facing camera is used to capture the target partner actions (in this paper, we bypass its function by directly simulating the target partner actions using a separated partner behavior module).

B. Modeling The Target Partner's Action

The target partner's actions are modeled by following our design as follows.

1) The target partner is designed to always follow the robot from behind with a random speed from 0 up to 0.75 m/sec.
2) The target partner movement direction is determined by V-Rep when he is blocked by an obstacle, and is determined using the robot position when following the robot.
3) The target partner head orientation is represented using three models, that are always looking at the robot, always alternating his head (modeled using sinusoidal function), and looking at a certain direction (modeled using sigmoid function) along guiding task. The executed model is chosen randomly for each episode. Execution time of the model is also randomized.

C. Learning Phase

1) Fixed-parameter setting: Before starting the training, we set several robot's parameters: $m = 20$ kg, $v_{max}^0 = 1$ m/sec, $\tau = 0.015$ sec, $k_{max} = 20$ N, $\Psi_{max} = 3.6$ m, $\lambda_{max} = 1$, and the ellipse's major and minor radius are 1.8 m and 0.7 m, respectively. The Q-learning algorithm parameters setting is shown in Table. II. All of the parameters with subscript max are proportionally divided by the number of actions.

2) Scenarios: We built three scenarios as shown in Fig. 4 to train our robot. These scenarios are used to complete as much as possible states.

3) Training results: As the training result, we show several examples of the behaviors of the robot during learning in Fig. 5. From those figures, we can show that our Q-learning based system tries to obtain the optimal policy in each episode by slowly but surely updating the Q-values for each parameter.

D. Testing Phase

1) Scenarios: We built three scenarios as shown in Fig. 6 to evaluate our robot performance. We performed 30 trials for each scenario.

TABLE II. Q-learning parameters settings

Parameters	Value
Number of training episodes	2,000
Number of states	43,200
Number of actions	25
Initial ϵ	0.6
Learning rate α	0.1
Discount factor γ	0.7
$\mathbf{F}_{max}^g$	200 N

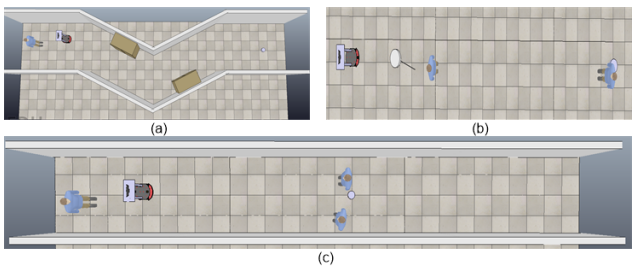

Fig. 4. Training scenarios: (a) indoor with static obstacles, (b) outdoor with two persons (static and dynamic with defined-trajectory route), and (c) narrow indoor with two persons (dynamic and free movement).

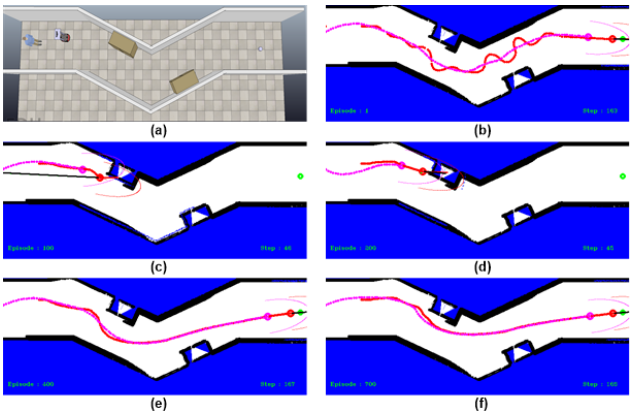

Fig. 5. Examples of the training behaviors of the robot in indoor scenario. The environment setting (a), and the progress of the online learning is presented by the last step of each learning episode (b-f). Red lines represent the robot's trajectories and magenta lines represent the human partner's trajectories. Green circle represents the target goal. Robot achieves smoother trajectory after 400 episodes.

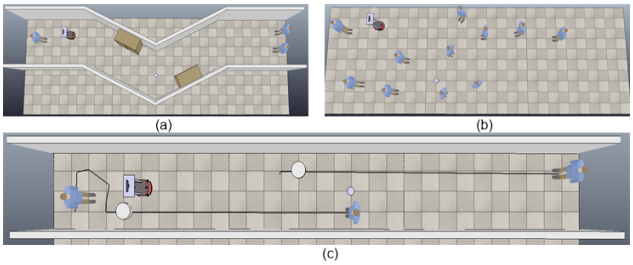

Fig. 6. Testing scenarios: (a) indoor with static obstacles and two free moving persons, (b) outdoor with 10 free moving persons, and (c) narrow indoor with two persons (dynamic and defined-trajectory routes). A bluish cylinder represents the target goal.

2) Smoothness: A smoothness of the robot motions is measured using a mean and a standard deviation of the robot altered-headings along its movements toward the goal. A simple moving difference filter is applied on the robot's heading with a kernel $[-1, 0, 1]$ to count how many the robot makes sudden unexpected movements with an angle more than 30^o. The smoothness of our robot movements is shown in Table. III. From scenario (a) and (c) our robot is able to minimize the number of sudden movements because of a little number of dynamic obstacles. In a more complex situation (b), our robot also successfully curbs its unecessary movements.

3) Safety: Safety can be measured using a percentage of successful task completion without threatening humans or the robot. The percentage of successful and safe movements of our robot are shown in Table. III. From the table, our robot achieves 66.67% of successful trials of all scenarios. Almost all failures are caused by unnatural movements of the simulated dynamic obstacles (the other persons), where how they avoid the collision with the robot often even crashing into the robot.

TABLE III. THE SMOOTHNESS OF MOTIONS, SUCCESSFUL TRIALS AND TIME NEEDED TO COMPLETE THE TASK.

Scenario	Smoothness (times)	Successful Trials (%)	Time (sec)
(a)	4.53 ± 4.24	70.00	16.73
(b)	10.90 ± 6.77	63.33	8.59
(c)	5.60 ± 4.29	66.67	15.60

V. CONCLUSION

We have developed a self-learning framework for a mobile guide robot which is able to navigate among social environments while guiding a target partner. The proposed framework is more emphasis on the behavior of the robot that meets smoothness and safety when performing its task. Our robot is equipped with Q-Learning based Social Force Guiding Model (QL-SFGM) to find the best socio-psychological-based control action for each particular situation. Experimental results show that using the self-learning framework, the over-reactive behaviors can be minimized by still considering the safety. The experimental results using the simulator show that our method is very promising and is applicable to the real robot. Implementing this method in a real robot application and measuring the real human comfort will be our next focus.

ACKNOWLEDGMENT

We would like to thank to the Directorate General of Higher Education, Ministry of Research, Technology, and Higher Education of Indonesia for financially supporting the first author under grant No. 224/E4.4/K/2012. This work is also in part supported by JSPS Kakenhi No. 25280093.

REFERENCES

[1] G. Ferrer, A. Garell, and A. Sanfeliu, *Robot Companion: A Social Force Based Approach With Human Awareness-Navigation in Crowded Environments*, IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.1688-1694, 2013.

[2] J. Satake, M. Chiba, and J. Miura, *Visual Person Identification Using a Distance-Dependent Appearance Model for a Person Following Robot*, Int. Journal of Automation and Computing, Vol.10(5), pp.438-446, 2013.

[3] I. Macaluso, E. Ardizzone, A. Chella, M. Cossentino, A. Gentile, R. Gradino, I. Infantino, M. Liotta, R. Rizzo, and G. Scardino, *Experiences with Cicerobot, a Museum Guide Cognitive Robot*, in: S. Bandini, S. Manzoni (Eds.), AI*IA 2005, Springer-Verlag, pp.474482, 2005.

[4] E. Pacchierotti, H.I. Christensen, and P. Jensfelt, *Design of an Office-Guide Robot for Social Interaction Studies*, IEEE/RSJ International Conference on Intelligent Robotics Systems, pp.4965-4970, 2006.

[5] T. Kanda, M. Shiomi, Z. Miyashita, H. Ishiguro, and N. Hagita, *An Effective Guide Robot in a Shopping Mall*, The 4th ACM/IEEE International Conference on Human Robot Interaction, pp.173-180, 2009.

[6] C. Feng, S. Azenkot, and M. Cakmak, *Designing a Robot Guide for Blind People in Indoor Environments*, ACM/IEEE International Conference on Human-Robot Interaction, pp.107-108, 2015.

[7] R. Triebel, K. Arras, R. Alami, L. Beyer, S. Breuers, R. Chatila, M. Chetouani, D. Cremers, V. Evers, M. Fiore, H. Hung, O.A.I. Ramirez, M. Joosse, H. Khambhaita, T. Kucner, B. Leibe, A.J. Lilienthal, T. Linder, M. Lohse, M. Magnusson, B. Okal, L. Palmieri, U. Rafi, M.V. Rooij, and L. Zhang, *SPENCER: A Socially Aware Service Robot for Passenger Guidance and Help in Busy Airports*, Field and Service Robotics, Vol.113, pp.607-622, 2016.

[8] E. Hall, *The Hidden Dimension*, Anchor Books, 1966.

[9] F. Zanlungo, T. Ikeda, and T. Kanda, *Social Force Model With Explicit Collision Prediction*, A Letters Journals Exploring The Frontiers of Physics, Vol.93, 2011.

[10] M. Luber, J.A. Stork, G.D. Tipaldi, and K.O. Arras, *People Tracking with Human Motion Predictions from Social Forces*, IEEE International Conference on Robotics and Automation, pp.464-469, 2010.

[11] D. Helbing and P. Molnar, *Social Force Model for Pedestrian Dynamics*, Physical Review E, Vol.51, No.5, pp.4282-4286, 1995.

[12] J.H. Holland, *Genetics Algorithm, Scientic American*, pp.66-72, 1992.

[13] R.S. Sutton and A.G. Barto, *Reinforcement Learning: An Introduction*, The MIT Press, 2012.

[14] http://www.coppeliarobotics.com/, *Virtual Robot Experimentation Platform*.