Trust Learning for Initiating Physical Human-Robot Interaction

Juyoun Park
School of Engineering and Applied Science
The George Washington University
Washington DC, United States
juyoun726@gwu.edu

Chung Hyuk Park
School of Engineering and Applied Science
The George Washington University
Washington DC, United States
chpark@gwu.edu

***Abstract*—Earning trust is essential for physical human-robot interaction, and a robot need to learn how to gain trust before initiating interaction. Current research on trust has conducted experiments where people interact with real robots. In this paper, especially in relation to the COVID-19 situation, we develop a simulation in which the robot learns the policy of moving closer to the person to initiate a simple medical interaction such as temperature measuring. We incorporate facial expression of a person in confrontation of a robot and the distance between the person and the robot in our reinforcement learning process. In the developed simulation, the robot moves according to the learned trust knowledge to lower the discomfort level and successfully approach a person.**

***Index Terms*—Physical human-robot interaction, policy network, reinforcement learning, simulation, trust**

I. Introduction

Physical human-robot interaction (pHRI) is inevitable. In particular, robots can effectively reduce the risk that comes from situations where contact between people can be dangerous. Robots can provide alternative solutions in conducting basic medical tasks such as temperature measurements while reducing physical contacts between people, aiding medical staffs and first responders. In pHRI, trust is one of the most important factors to be considered. If people feel uncomfortable, it is not appropriate for robots to get closer. There have been researches on trust such as quantifying the effects of a human, robot, and environmental factors on perceived trust in human-robot interaction [1]. Perceiving the responses from human and applying to real-time policy learning in pHRI is essential, however the real-world experiments have many challenging issues. Therefore, we developed a hybrid simulation that a simulated robot learns the policy for moving while extracting features from the human's responses from real-world with a camera who is observing the robot's actions in simulation, which reflects the robot and environmental mapping representing human intentions. We employ reinforcement learning (RL) algorithms to process features that are important to trusting the robot that reflects the policy to determine the next moves. In the developed simulation, the robot moves according to the knowledge learned so that people do not feel uncomfortable.

Human-robot interaction is difficult to train in advance using simulation. In a typical RL training environment in simulation, the robot has its state, and after observing the environment, it acts and gets rewarded in the environment. In this completely autonomous environment, it is difficult to reveal human intentions and reflect them in experiments. Therefore, we developed a new simulation to connect the simulation environment with the intention expression of real users. This will allow the robot to learn in advance the policy regarding human-robot interaction in simulation.

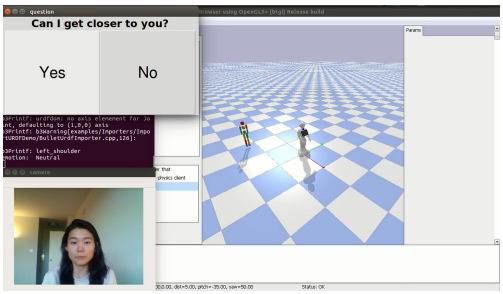

Fig. 1. Simulation environment for trust policy learning.

II. Method

We developed our simulation using Pepper, a humanoid robot, as shown in Fig. 1. Since we used Pepper, we created a simulation environment based on qiBullet [2], which is a Bullet [3]-based python simulation for SoftBank Robotics' robots. And, we implemented an RL *OpenAI gym env* [4] for Pepper so trust policy can be trained using a variety of RL methods. We used a webcam on the PC to get the image of the user sitting in front of the PC in real-time. During the simulation, the user can see how Pepper moves around the model of the person, which is considered to be the user, in the simulation environment. Based on the RGB image, real face features of the user are extracted in real-time and reflected in the learning of the trust policy that determines how Pepper moves.

First, the *dlib* [5] library was used to crop and align the face parts in the RGB image. By integrating the facial expression recognition network [6], facial features were extracted. We assume that the user's facial expression can show how the user feels about the situation that Pepper is getting closer to the user. The distances measured by the three laser sensors on

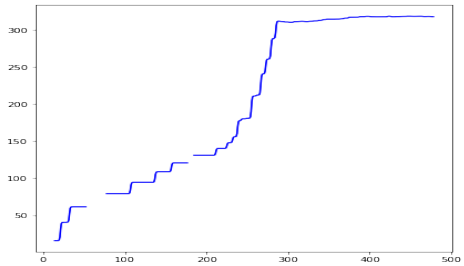

Fig. 2. Results of training the policy network using A2C. The change in reward (y-axis) over episode (x-axis) was plotted. The empty parts of the graph had negative value rewards and were not displayed.

Pepper were also used. The facial features and laser sensor values were used as the state of the RL environment. Actions were defined in five ways: move forward, move backward, move right, move left, and stop. During RL, a policy network is learned that generates action from the features used as the state. We also added aspects of social interaction to the simulation by reflecting user responses as to whether Pepper can come to the user (the human model of the simulation). The human model moves according to the human responses, and the responses are reflected in determining the reward.

We set the goal as Pepper approaching the user a certain distance ($50cm$). Each episode ended when the goal was achieved, the human moved away from Pepper and cannot be detected, or the step exceeds a large value. The first case is considered successful and the other two are considered failures. The reward was set for each step to 10 times the value of the target distance minus the distance from Pepper to the human model. If the user allowed Pepper to approach the human model, the reward was determined by 30 times the value of the previous distance minus the current distance. And if the episode was successful, a large positive value (300) was provided. If the episode failed, that is, if the user was not detected far from Pepper, or if the step was greater than a large predefined value (1000), the episode failed and a negative value (-30) was provided. Simulations can be done in real-time and policy networks can be trained.

III. Results

The real-time demonstrations were shown as a video attachment. As shown in Fig. 1, the user can view the simulation environment in real-time and the RGB image loaded from the webcam is also displayed. For each step, the user can answer whether Pepper can get closer to the human model by clicking on the displayed button.

When training the policy, one of the three RL algorithms can be chosen: Advantage Actor-Critic (A2C) [7], Proximal Policy Optimization (PPO) [8], or Scalable trust-region method for deep reinforcement learning using Kronecker-factored approximation (ACKTR) [9]. A2C is a synchronous deterministic version of A3C [7]. We used the open-source code for training and evaluation [10]. In this paper, we used A2C to train policy and showed the results as a reward graph, as shown in Fig. 2. The change in reward over episode was plotted. The reward increases over about 300 episodes and then converges to about 300 given as a reward for success. From the results of the increased reward, it was verified that our trust policy was effectively trained in the developed simulation environment.

IV. Discussion

Since RL uses rewards obtained in response to random situations for policy network learning, it is difficult to apply RL to pHRI that utilizes real robots and involves human interaction. Also, generating a random situation in pHRI is extremely dangerous because robots have great power and there must be physical contact during the interaction. In this sense, our simulation reduces these issues and allows to train the policy in a safe situation. Besides, in the experiments using a real robot, it takes a lot of time and works to prepare for the experiment, such as setting up the robot and recruiting people for interaction. Our model can reduce them. We expect our trust policy learning to be generally used in a variety of situations involving physical interactions. This is because it is a general way to reflect facial features and sensor values in the robot's motion decision.

For our future work, we plan to reflect the intentions of people inherent in the semantic features of sentences by adding dialogue interactions based on natural language processing. We also plan to use the developed simulation to learn the trust policy as a fast adaptation stage, and then test and finetune it in real interaction situations.

Acknowledgment

This research is supported by the National Science Foundation (NSF) Disability and Rehabilitation Engineering (DARE) program under the grant #1846658.

References

[1] P. A. Hancock, D. R. Billings, K. E. Schaefer, J. Y. Chen, E. J. De Visser, and R. Parasuraman, "A meta-analysis of factors affecting trust in human-robot interaction," *Human factors*, vol. 53, no. 5, pp. 517–527, 2011.

[2] M. Busy and M. Caniot, "qibullet, a bullet-based simulator for the pepper and nao robots," *arXiv preprint arXiv:1909.00779*, 2019.

[3] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2019.

[4] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," 2016.

[5] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[6] D. Meng, X. Peng, K. Wang, and Y. Qiao, "frame attention networks for facial expression recognition in videos," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 3866–3870.

[7] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*, 2016, pp. 1928–1937.

[8] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[9] Y. Wu, E. Mansimov, R. B. Grosse, S. Liao, and J. Ba, "Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation," in *Advances in neural information processing systems*, 2017, pp. 5279–5288.

[10] I. Kostrikov, "Pytorch implementations of reinforcement learning algorithms," https://github.com/ikostrikov/pytorch-a2c-ppo-acktr-gail, 2018.