Two-Level Reinforcement Learning Framework for Self-Sustained Personal Robots

Koyo Fujii[1][0009−0007−8606−9031], Patrick Holthaus[2][0000−0001−8450−9362], Hooman Samani[2,3][0000−0003−1494−2798], Chinthaka Premachandra[1][0000−0002−5775−5047], and Farshid Amirabdollahian[2][0000−0001−7007−2227]

[1] Department of Electronic Engineering, Shibaura Institute of Technology, 3-7-5 Toyosu, Koto-ku, Tokyo 135-8548, Japan
{ag20045,chinthaka}@shibaura-it.ac.jp
[2] Robotics Research Group, University of Hertfordshire, College Lane, Hatfield, AL10 9AB, United Kingdom
{p.holthaus,f.amirabdollahian2}@herts.ac.uk
[3] Creative Computing Institute, University of the Arts London London, SE5 8UF, United Kingdom
h.samani@arts.ac.uk

Abstract. As social robots become integral to daily life, effective battery management and personalized user interactions are crucial. We employed Q-learning with the Miro-E robot for balancing self-sustained energy management and personalized user engagement. Based on our approach, we anticipate that the robot will learn when to approach the charging dock and adapt interactions according to individual user preferences. For energy management, the robot underwent iterative training in a simulated environment, where it could opt to either "play" or "go to the charging dock". The robot also adapts its interaction style to a specific individual, learning which of three actions would be preferred based on feedback it would receive during real-world human-robot interactions. From an initial analysis, we identified a specific point at which the Q values are inverted, indicating the robot's potential establishment of a battery threshold that triggers its decision to head to the charging dock in the energy management scenario. Moreover, by monitoring the probability of the robot selecting specific behaviours during human-robot interactions over time, we expect to gather evidence that the robot can successfully tailor its interactions to individual users in the realm of personalized engagement.

Keywords: Personalized interaction · Companion robots · Battery Management · Reinforcement learning

1 Introduction

As social robots become more and more integrated into everyday human life, their handling becomes an increasingly complex issue. One of the most important aspects to consider is managing the robots' battery life[5]. Especially during

long-term human-robot interactions (HRIs), it would be cumbersome for users to continually monitor their robots' battery status to send them to the charging dock when the battery is close to depletion. Furthermore, robot adaption to user preference is certainly a key element of long-term interactions between humans and robots[7]. Hence, it would be beneficial if a robot could autonomously navigate to its charging dock and replenish its battery at an optimal time determined by its own algorithms, considering social interaction. At the same time, it would be advantageous if home robots could tailor their interactions to individual users, enhancing their utility and user experience.

In this paper, we present approaches that employ Q-learning [16] to combine both these aspects. The primary contributions of this paper include (1) determining the optimal timing for the robot to approach the charging dock using Q-learning in a simulated environment; and (2) enabling the robot to adapt to individual users over time during human-robot interactions by leveraging Q-learning with a real Miro-E robot. For that, we first present some existing approaches to battery management and personalized user engagement in Section 2 and introduce theoretical backgrounds about Q-learning in Section 3. After that, we present our method by describing our own implementation of Q-learning for self-sustained energy management and personalised user engagement in Section 4. We further provide an initial proof of concept of our approach in Section 5 before concluding the paper.

2 Background

In the field of battery management, a diverse range of methodologies have been established. Some approaches do not incorporate learning but rely on estimation functions [4], or model predictive control [10]. Many others instead [1,3,12,8] used strategies involving energy storage and decision-making frameworks using some form of reinforcement learning, allowing for dealing with uncertainties effectively. Likewise, our approach is based on a form of reinforcement learning (Q-learning, c.f. Section 3).

To personalize and adapt a robot's user engagement, frameworks have been proposed by [9], while [6] have presented designs, implementations, and assessments for socially assistive robots. [9] allows robots to understand children with ASD's emotions using physiological signals, while [6] motivates elderly users to exercise via a vision-equipped robot. However, the adaptation techniques vary. [9] utilizes random phrase selections during exercises to avoid repetitiveness, while [6] employs Support Vector Machine (SVM)-based modelling to interpret children with autism's physiological signals. The work presented here combines such reinforcement learning-based behavioural adaption systems (e.g. [11,14]) with reinforcement-based solutions for autonomous battery management.

Our approach extends our previous work [2] in which we effectively utilized Q-learning for "Energy Autonomy" and "User's Preferences" in a study involving an early version of the Aibo robot[4]. There, we demonstrated a robot that could

[4] See: https://electronics.sony.com/more/c/aibo

operate for extended periods without depleting its energy source and had successfully learned an effective policy for engaging users through real-world interactions. Current work follows up on this study, replicating the original methods using a modern Miro-E[5] robot. Additionally, we have expanded and improved some methodological aspects of the original work, c.f. Section 4.

3 Theory

In this section, we will briefly introduce the theoretical background to the learning algorithm used in this work. Specifically, we discuss Q-learning [16], the epsilon-greedy [13], and the softmax [15] policies, which we consider in our implementation. The goal of Q-learning is to find optimal Q values, q_*, which means to find an optimal policy π_* as the policy $\pi(a|s)$ that maximizes the expected total reward from a given state. Q values are a measure of the expected return after taking a specific action in a specific state with a particular policy. The learned Q values directly approximate q_*, independent of the policy being followed [13] because Q-learning is an off-policy algorithm and its updates always reflect the maximum expected reward. This specifically enables early convergence of a chosen policy and the target policy can be deterministic, while the behaviour policy can continue to sample all possible actions [13]. Therefore, Q-learning is a simple way for agents to learn how to act optimally in controlled Markovian domains as articulated by Christopher[16]. The update for Q-learning is defined as follows:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)) \quad (1)$$

In off-policy algorithms, the policy used to generate behaviour called the behaviour policy, may in fact be unrelated to the policy that is evaluated and improved, called the target policy. The Q-learning updates its Q-values to align with the optimal (or "target") policy. However, while the behaviour policy could in theory be any policy, it should be soft (i.e. it should consider all actions in all states with nonzero probability) in order to explore all possibilities [13].

In reinforcement learning, maintaining an appropriate balance between exploration and exploitation is a crucial aspect. A simple yet effective strategy for managing the exploration-exploitation trade-off is the epsilon-greedy action selection mechanism [13]. With this approach, the agent selects an action that maximizes its Q-value for a given state with a probability of $1 - epsilon$ and chooses an action randomly with a probability of $epsilon$. The epsilon-greedy policy treats the selection probability of all non-greedy actions equally, thereby neglecting the estimated Q-values for these actions.

However, softmax [15] uses action-selection probabilities which are determined by ranking the Q-value estimates using a Boltzmann distribution. In practical applications, to prevent overflow and ensure numerical stability, τ denotes

[5] See: https://miro-e.com/robot

a positive parameter known as the 'temperature':

$$\pi(a|s) = Pr\{a_t = a|s_t = s\} = \frac{e^{\frac{Q(s,a)-\max_b Q(s,b)}{\tau}}}{\sum_b e^{\frac{Q(s,b)}{\tau}}} \quad (2)$$

4 Method

In this section, we describe our implementation of Q-learning on the Miro-E robot to allow for self-sustained energy management and personalized user engagement. The goal of self-sustained energy management is to determine an optimal threshold for charging, thus enabling Miro-E to engage in extended periods of interaction for enhanced human-robot interaction. To this end, we extended the original approach with a negative reward system [2], which encourages Miro-E to engage in play and discourages battery depletion at the same time. In addition to the original approach [2], where the state dimension was one-dimensional, we introduced an additional dimension called "people's faces" in personalized user engagement. This addition is anticipated to facilitate more personalized interactions and provide flexibility in the learning process. By making these modifications to the original work, we aim to develop a robot that optimizes battery use and potentially offers personalized features for each user. To efficiently facilitate the training of self-sustained energy management in simulation and trial user engagement in the real world, this work addresses both aspects individually.

4.1 Self-sustained energy management

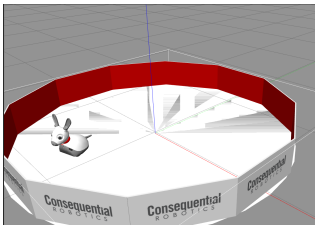
A robot must visit the battery charging dock to maintain autonomous movement. Ideally, it should be able to play around in a room for extended periods and approach the charging dock with optimal timing. To achieve this autonomous behaviour and expedite convergence as compared to on-policy learning methods such as SARSA [13], we employed Q-learning in a simulation environment.

Q-learning implementation For learning self-sustained energy management, we implemented an epsilon-greedy policy for the selection of actions to allow Miro-E to determine action probabilities based on epsilon, independent of Q values, which are updated to maximize the next Q value in Q-learning. We configured the reinforcement learning parameters as follows:

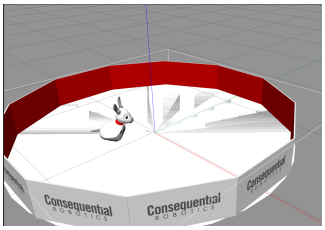
- State space (two-dimensional Q-table): "charging" or "playing", the battery level is divided into levels ranging from 6 (fully charged) to 0 (nearly empty). We designed this two-dimensional state space to enable Miro-E to select its next action based on its current engagement and battery level.
- Initial state: when the first dimension of the state space is "playing" and the second dimension (the battery level) is 6 (fully charged).

- Terminal state: either when the first dimension of the state space is "playing" and the second dimension (the battery level) reaches 0, or when the steps within a single episode reach 500.
- Action space: "play" or "go to charging dock".
- Reward: a reward of +100 is provided when Miro-E opts to play to incentivise longer playtime, and a reward of -100 is given when the robot decides to proceed to the charging dock to discourage unnecessary returns. If Miro-E depletes its battery, a penalty is assessed that is 100 times the number of steps taken, with this counter resetting once Miro-E returns to the charging dock. We have chosen this penalty structure to prevent the battery from running out, ensuring that the penalty magnitude exceeds the reward value associated with choosing to play.

Evaluation environment We used a simulation environment to determine whether the robot can change its behaviour from engaging a user to going to a virtual charging location using the above implementation. Figure 1a depicts the moment when Miro-E is playing while Figure 1b captures the moment when Miro-E is moving to a predetermined position. We configured the following parameters for the Q-learning algorithm in the simulation: Learning rate at 0.1, Discount factor at 0.9, Initial epsilon for the epsilon-greedy method set to 0.3, Epsilon discount rate of 0.99, the maximum number of steps set to 500, and a total of 200 episodes.



(a) example of "play" action



(b) example of "going to dock" action

Fig. 1: Examples of Miro-E actions in simulation.

4.2 Personalized engagement

Individual preferences for behaviour vary and consequently, a robot should adapt to the specific person it is interacting with. To achieve this, we also employed Q-learning. In our use case, the robot interacted with an actual person in the real world, as it needs to adapt to existing individuals. We introduced a novel element to facilitate personalized engagement. Specifically, we enabled the robot

to recognize a human face, allowing the robot to adapt to the specific preferences of the identified individuals. In the following section, we describe the implementation of Q-learning and outline the experimental setup and procedure.

Q-learning implementation For personalizing user engagement, we implemented a softmax policy for the selection of actions, allowing Miro-E to determine action probabilities based on their corresponding Q-values and to ensure that actions have a nonzero probability of being selected during an interaction. Additionally, our updating strategy aims to facilitate dramatic changes in Q values compared to on-policy methods like SARSA [13] to allow for faster user adaptation. We configured the reinforcement learning parameters as follows:

- State space (two-dimensional Q-table): the person's face, "tracking a ball", "responding to sound", "detecting a person's face" or a state of inactivity.
- Initial state: when the robot is not engaged in any actions.
- Terminal state: when the user sends a signal.
- Action space: "track a ball," "respond to sound," "detect a person's face".
- Reward: a reward of +10 when a person pats Miro-E on its head, indicating a preferred action, while no rewards are given for other actions.

Evaluation environment To evaluate our approach, we implemented an interactive learning routine using a real Miro-E robot as follows: At the beginning of each episode, the first state dimension is determined by recognizing a pre-registered person's face. Then, one of the actions is selected using the softmax method and executed.

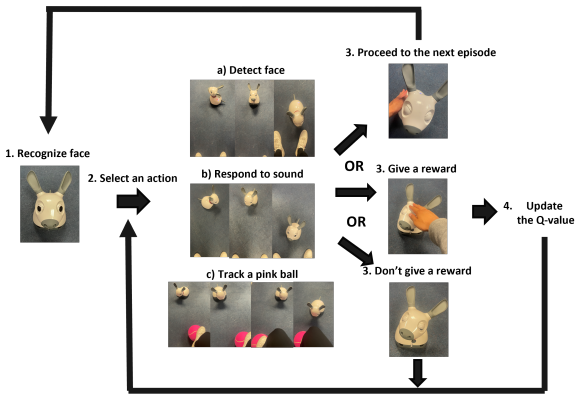


Fig. 2: Behaviour flow of Miro's interactive training routine.

If a reward is given by the user, the Q-value is updated and the subsequent action is chosen. If not, the next action is determined. If the user signals they want more adaptive actions by patting Miro-E on its body, the temperature

parameter is adjusted by multiplying it with the discount factor before selecting the next action. Figure 2 summarises the interactive training steps for the behaviour adaption. We configured the following parameters for the Q-learning algorithm in the real world: learning rate at 0.5, discount factor at 0.9, initial temperature for the softmax method set to 100, and temperature discount rate of 0.9.

5 Proof of Concept

To determine whether our approach can function, we tested the energy management routine and the behaviour adaption separately. Firstly, we tested whether we could find a valid timing for the robot to approach the charging dock and secondly, whether the robot would adapt its behaviour to a user over time.

5.1 Self-sustained energy management

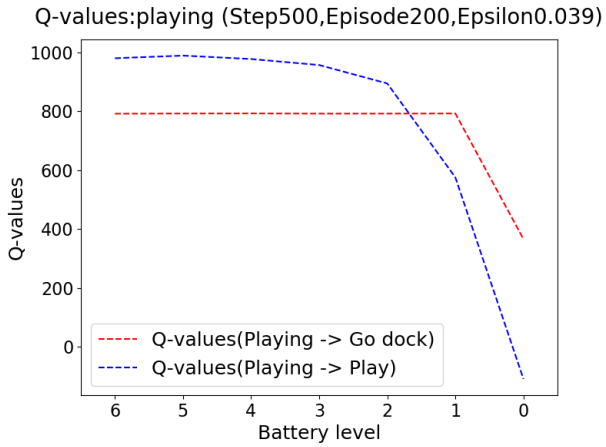
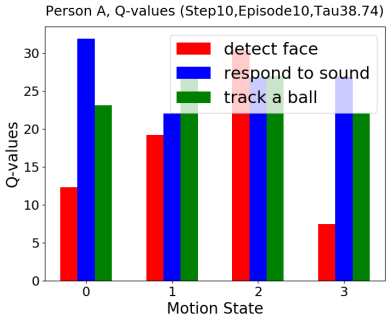


Fig. 3: Q value: Motion state "playing".

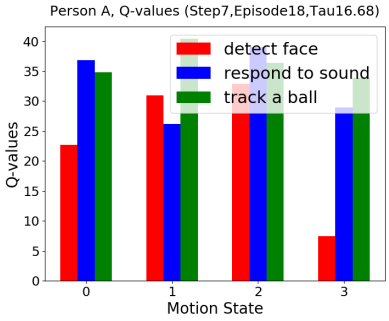
The objective of the first evaluation was to identify the optimal battery threshold that would enable the Miro-E robot to operate for extended periods. For that, we have investigated how the Q-values change when the motion state of the robot is "playing". Figure 3 illustrates that when the Miro-E robot was in a "playing" state, the Q-value for "play" exceeded the Q-value for "go to dock" until the battery level dropped between 2 and 1. Beyond this point, the Q-values inverted, indicating that "go to dock" became the more valued action. Based on the results, the optimal battery threshold appears to be between a battery level of 2 and 1. More precisely, the voltage corresponding to this threshold is 4.4 V, suggesting that the identified threshold is approximately 4.4 V.

5.2 Personalized engagement

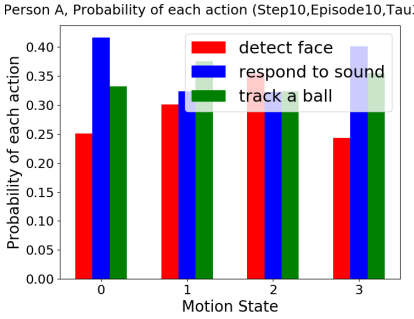
The second part of our evaluation looks at whether Miro-E would adapt its behaviour during an interaction. For that, we provided the system with different rewards in a test run lasting for approximately 60 minutes. Figures 4a and 4b depict the Q-values at episodes 10 ($\tilde{3}0$ minutes) and 18 ($\tilde{6}0$ minutes), respectively, while Figures 4c and 4d show the probabilities of selecting each action at the same episodes and corresponding times, which show that the probabilities associated with each action evolve over time, indicating Miro-E's adaptation to a specific person's preferences. Consequently, Miro-E likely selects "respond to sound" following actions "detect a person's face" and "track a ball". Additionally, "track a ball" is probably chosen after "respond to sound" or at the episode's outset.



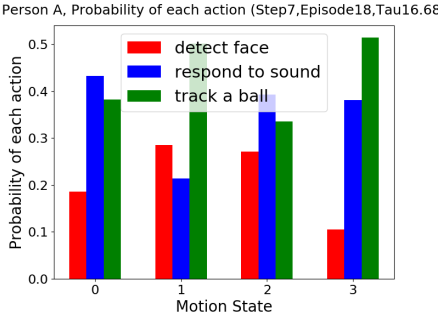
(a) Q-values in ep. 10



(b) Q-values in ep. 18



(c) Probability of each action in ep. 10



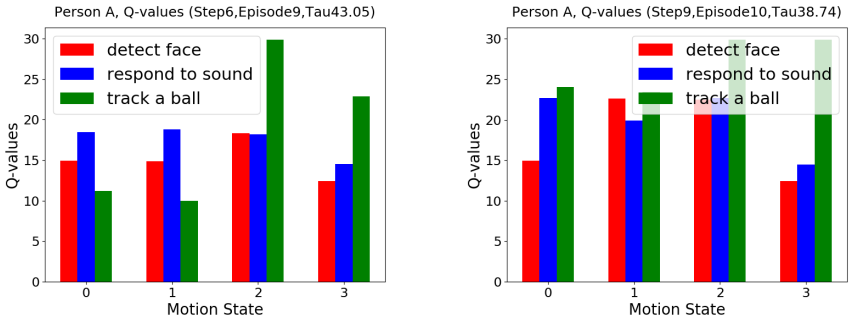
(d) Probability of each action in ep. 18

Fig. 4: Test run: Q-values and action probabilities in some example episodes.

Miro-E chose actions using the softmax formula outlined in Equation 2. The temperature parameter was adjusted throughout each episode, especially when the user signaled a desire for more adaptive interactions by patting Miro-E. Initially, Q-values had minimal influence on action choices due to a high temperature parameter. But as episodes advanced and the temperature decreased,

the influence of Q-values on action selection grew stronger. This behaviour is evident in Figure 4, which displays results for one person with the first state space dimension set to "1", Motion State "3" in Figures 4a and 4b showing Miro-E inactive at the beginning of an episode. Here, Q(13, 0) remains unchanged between episodes 10 and 18. However, in Figures 4c and 4d, despite static Q values, there's a growing difference in the likelihood of Miro-E choosing "track a ball" and "respond to sound over "detect a person's face". This suggests Miro-E gradually refines interactions based on both exploring user preferences and leveraging past experiences.

Its adaptability to changing user preferences was enhanced by Q-learning. The Q-learning formula (Equation 1) ensures that if an action was rewarded, the related Q value would adjust to improve future rewards. This could mean large increases in Q values for less-favored actions, thereby increasing their chances of selection and allowing Miro-E to quickly modify its interactions. Rapid changes in Q values across episodes can be observed, for instance, between episodes 9 and 10 in Figures 5a and 5b. Notably, the test run depicted in Figure 5 is unrelated to that in Figure 4. Q values, such as Q(10,2), Q(11,2), and Q(13,2) exhibited significant changes within the span of just one episode.



(a) Q values in episode 9 (b) Q values in episode 10

Fig. 5: Q values in episodes 9 and 10 during the second test run.

6 Conclusion

In this paper, we utilized Q-learning with the Miro-E robot to successfully attain self-sustained energy management and personalized engagement. For self-sustained energy management, we showed that the robot could determine the optimal timing for approaching the charging dock in a simulated environment. For personalized engagement, we anticipate that our method will adeptly adapt the robot's interactions over time to meet the preferences of an individual user during human-robot interactions.

Our future work is to evaluate these algorithms in an interactive trial involving different individuals with free choices of interaction, as offered by Miro-E and implemented additionally. Moreover, we consider the expansion of the state space, e.g. introducing an idling state to the self-sustained energy management component.

References

1. Cao, J., Harrold, D., Fan, Z., Morstyn, T., Healey, D., Li, K.: Deep reinforcement learning-based energy storage arbitrage with accurate lithium-ion battery degradation model. IEEE Transactions on Smart Grid **11**(5), 4513–4521 (2020)
2. Castro-González, A., Amirabdollahian, F., Polani, D., Malfaz, M., Salichs, M.A.: Robot self-preservation and adaptation to user preferences in game play, a preliminary study. In: International Conference on Robotics and Biomimetics. pp. 2491–2498 (2011)
3. Chaoui, H., Gualous, H., Boulon, L., Kelouwani, S.: Deep reinforcement learning energy management system for multiple battery based electric vehicles. In: 2018 IEEE Vehicle Power and Propulsion Conference (VPPC). pp. 1–6. IEEE (2018)
4. Chellal, A.A., Lima, J., Gonçalves, J., Megnafi, H.: Battery management system for mobile robots based on an extended kalman filter approch. In: 2021 29th Mediterranean Conference on Control and Automation. pp. 1131–1136. IEEE (2021)
5. Deshmukh, A., Aylett, R.: Socially constrained management of power resources for social mobile robots. In: International Conference on Human-Robot Interaction. pp. 119–120 (2012)
6. Fasola, J., Mataric, M.J.: Using socially assistive human–robot interaction to motivate physical exercise for older adults. Proceedings of the IEEE **100**(8), 2512–2526 (2012)
7. Gockley, R., Broz, F., Forlizzi, J., Michalowski, M., Mundell, A., Rosenthal, S., Sellner, B., Simmons, R., Snipes, K., Schultz, A., Wang, J.: Designing robots for long-term social interaction. pp. 1338 – 1343 (09 2005)
8. Kuznetsova, E., Li, Y.F., Ruiz, C., Zio, E., Ault, G., Bell, K.: Reinforcement learning for microgrid energy management. Energy **59**, 133–146 (2013)
9. Liu, C., Conn, K., Sarkar, N., Stone, W.: Online affect detection and robot behavior adaptation for intervention of children with autism. IEEE Transactions on Robotics **24**(4), 883–896 (2008)
10. Liu, Y., Zhang, J.: Self-adapting j-type air-based battery thermal management system via model predictive control. Applied Energy **263**, 114640 (2020)
11. Mitsunaga, N., Smith, C., Kanda, T., Ishiguro, H., Hagita, N.: Adapting robot behavior for human–robot interaction. IEEE Transactions on Robotics **24**(4), 911–916 (2008)
12. Natella, D., Vasca, F.: Battery state of health estimation via reinforcement learning. In: 2021 European Control Conference (ECC). pp. 1657–1662. IEEE (2021)
13. Sutton, R., Barto, A.: Reinforcement learning: An introduction. MIT press (2018)
14. Tapus, A., Ţăpuş, C., Matarić, M.J.: User-robot personality matching and assistive robot behavior adaptation for post-stroke rehabilitation therapy. Intelligent Service Robotics **1**, 169–183 (2008)
15. Tokic, M., Palm, G.: Value-difference based exploration: adaptive control between epsilon-greedy and softmax. In: Annual conference on artificial intelligence. pp. 335–346. Springer (2011)
16. Watkins, C.J.C.H., Dayan, P.: Q-learning. Machine Learning **8**(3), 279–292 (1992)