Continuous shared control in prosthetic hand grasp tasks by Deep Deterministic Policy Gradient with Hindsight Experience Replay

Zhaolong Gao, Rongyu Tang, Luyao Chen, Qiang Huang and Jiping He

Abstract
Grasp using a prosthetic hand in real life can be a difficult task. The amputee users are often capable of planning the reaching trajectory and hand grasp location selection, however, failed in precise finger movements, such as adapting the fingers to the surface of the object without excessive force. It is much efficient to leave that part to the machine autonomy. In order to combine the intention and planning ability of users with robotic control, the shared control is introduced in which users’ inputs and robot control methods are combined to achieve a goal. The shared control problem can be formulated as a Partially Observable Markov Decision Process. To find the optimal control policy, we adopt an adaptive dynamic programming and reinforcement learning-based control algorithm-Deep Deterministic Policy Gradient combined with Hindsight Experience Replay. We proposed the algorithm with a prediction layer using the reparameterization technique. The system was tested in a modified simulation environment for the ability to follow the user’s intention and keep the contact force in boundary for safety.

Keywords
Shared control, reinforcement learning, adaptive dynamic programming prosthetic hand, telerobotics

Introduction
A major task for an anthropomorphic prosthetic hand is to perform a dexterous and stable grasping in daily life. A success grasp control consists of problems in different phases. In the pre-grasp phase, the control problem is about grasp planning which can be addressed as several complex factors including positioning the arm, orienting the wrist, and shaping the fingers subject to object placement and distribution, environment obstacles, and so on. On most occasions, these factors lie in a higher multimodal dimension and the solution may be intractable. Many studies have
proposed different methods for grasp planning, but in anthropomorphic robots, human planning still has its privilege over robot algorithms. Meanwhile, the precise force control is also needed in the grasping phase, especially during the manipulation tasks. Traditional robotic hands or prostheses are lack of rapid and intuitive feedbacks in control. In practice, prosthetic hand control for amputees usually takes a long time in training and is hard for the user to comprehend the states of the robotic hand. The limited sensory-motor control abilities make it difficult for subjects to adjust their fingers to the shape of the object. To avoid the drawbacks in two different phases, the shared control (SC) is introduced which combines human grasp planning ability with the automated grasp algorithm. SC makes fine adjustments to the fingers by processing information from the force sensors placed on the prosthetic hand fingers. In the pre-contact grasp phase, the movement planning can be done by users to achieve better embodiment since it is more intuitive. After the contact, the algorithm should try to keep grasp stable while taking the users’ intention into consideration. More generally speaking, SC strategies aim to bridge the gap between human intentions and efficient execution of the intended task by using information from the sensors.

The idea of “shared control” of combining the user’s command with robot algorithms has a long history and is widely applied in the robotic field. Salisbury proposed a control strategy for a robotic hand, where the robot autonomy can be intervened by the users’ commands, and user control could be augmented by the robot. Kim et al. proposed an SC structure for brain–machine interfaces where the goal is to share control arising from the user’s brain and robot sensors. Fong et al. designed a cooperative system relying on dialogue via prompting for decision-making and robot assurance. Especially, SC is widely used in dexterous manipulation in teleoperation and prostheses, where the target of the robotic hand is given remotely or by decoding the electrogams (e.g. electromyography (EMG), electroencephalogram (EEG), etc.). Since the input signal type of users varies from case to case, in order to illustrate the SC problem, we focus on the algorithms about how the target signal generated by the different controller mixed with each other without considering how these signals are extracted. Some adaptive algorithms used in SC are aimed to help people complete their tasks by keeping them in safety. Lopes et al. presented a fuzzy control-based SC algorithm for the navigation of assistive mobile robots. Saen et al. implemented an SC method in grasp tasks of a robotic hand by regulating the finger force with tactile sensors. By attaching force sensors to the surface of the hand, the robot can predict the properties of the grasped object and deal with the uncertainties. Javdani et al. address the SC approach as a Partially Observable Markov Decision Process (POMDP) in which the environment was not fully observable. POMDP can be solved with adaptive dynamic programming (ADP), which is a model-free approach suitable for application.

Recently, the reinforcement learning (RL) techniques have aroused attention in robot SC with ADP. Xu et al. proposed an RL-based SC algorithm for walking-aid robot where the robot is able to autonomously adapt to different user operation habits and motor abilities. The RL algorithm used here is Q-learning, which updates the action-value function for control. The Q-learning can only be applied to sparse action space, whereas the action inputs of the prosthetic hand are usually continuous signals. Besides, the Q-learning cannot update the action policy directly. Lillicrap et al. combined the deep learning techniques with RL for continuous control with an actor-critic network structure shown in Figure 1. The algorithm originating from ADP is called Deep Deterministic Policy Gradient (DDPG). DDPG can directly update the parameterized policy and intrinsically suit for continuous application. The object grasp task contains multiple considerations and the goal of it is consequently complex. To improve DDPG performance in a multi-goal environment, the Hindsight Experience Replay (HER) is introduced.

We proposed an SC system for robotic hand control with DDPG and HER algorithms. The data flow of the system is shown in Figure 2. The user intention controls the movements of fingers and the agent regulates the user control signal by taking the robot states and sensors into account.
consideration. Before the robot finger contact with the surface of the object, agent should act like transparent by deliver the user control signal directly to the robot. Once the fingers contact with the object, the force should be constrained to keep a steady grasp.

In the following sections of this article, a simulation environment of grasp task was built. The agent with DDPG+HER algorithm and environment predictor was trained in the simulator. In the end, the trained agent was tested concerning the performance in free move and force adjustment in grasping.

**Methods**

**Environments setup**

The simulation environment for grasping tasks is built using OpenAI Gym.²⁷ The environment contains a model of a Shadow Dexterous Hand (Shadow Robot Company, London, UK) which is an anthropomorphic robotic hand function as a prosthesis. There are 24 degrees of freedom in the hand and 20 of them can be controlled independently. In grasp task, a block is placed on the palm of the hand for grasping. The goal of the grasp task is a 15-dimensional vector containing the Cartesian position of each fingertip of the hand. The OpenAI gym uses the MuJoCo²⁸ physics engine for fast and accurate simulation.

The initial states environments are shown in Figure 3, which simulate the starting of the contact phase in a grasping task. Notice that the block may randomly not appear, which indicates that the fingertip position is in a free moving trial. The simulation runs step by step with an action frequency of \( f = 25 \) Hz. The state of the robotic hand is fully observable including 24 positions and velocities of the hand joints. Additional force sensors are attached to the surface of the hand for contact identification.

At the beginning of each trial, an object appears, and its position and rotation are not directly observable. Then, the fingertip goals are generated. The main goal of this environment is to achieve mean distance between fingertips and desired positions less than a certain threshold, which assumed to be generated by a user’s intention. When the goal lies inside the boundary of the block, the corresponding finger is not intended to contact with the object. If the goal lies inside, the finger should try to contact with the object surface within presetting force bounds. The safety requirements and grasp conditions were taken into consideration in choosing the adequate lower and higher boundaries.

The reward function of this environment can be continuous or sparse. However, considering the DDPG algorithms described in the following sections, the sparse reward function has better performance. Thus, the reward will be 0 if the distance between the finger position and the goal is less than the threshold or the reading of touch sensors on the fingertip lies in the force boundary. Otherwise, the trial is considered failed and the reward will be −1. The detailed trial setup will be explained in the following sections.

**Shared control**

SC algorithms combine control ability and machine control ability in grasping tasks by following the users’ intention while taking grasp stability, force limitation, and other limitations into consideration. An SC problem can be defined as a POMDP.²⁹ In a Markov Decision Process (MDP) with states \( S \), actions \( A \), transitions \( T : S \times A \times S \to [0, 1] \), reward function \( R : S \times A \times S \to \mathbb{R} \), and discount factor \( \gamma \in [0, 1] \), if the states \( S \) are not fully observable, we can reparametrize the states into \( S = \{ S', \Omega \} \), where \( S' \) is the observable part of states and \( \Omega \) are set of observations usually from information gathered by additional sensors implicating the unobservable part of states \( S \). Thus, the MDP can be generalized to a POMDP with observation function as a conditional probability \( \Omega = \mathcal{O}(\cdot | S', A) \). At timestep \( t \) with state \( s_t \), action \( a_t \), and reward function \( r(s_t, a_t) \), the return is defined as the sum of discounted future reward \( R_t = \sum_{i=t}^{\infty} \gamma^i r(s_i, a_i) \) following the policy \( \pi : S \times A \to [0, 1] \). The goal is to maximize \( R_\pi \) over trajectories \( \tau = (s_0, a_0, s_1, a_1, \ldots) \) with policy \( \pi \).

In this article, the states \( S' \) are the positions and velocities of the hand joints and the \( \Omega \) is generated from a prediction of force sensors. The action space \( A \) are

![Figure 3. The initial states with or without the target block in a grasp environment. (a) The block is put on the palm with random position and rotation. (b) The block is not put on the arm on some occasion and the colored ball indicates the goal of the fingertip.](image-url)
20-dimensional continuous normalized outputs of actuators. The policy and parameters selection are described in the next section.

**DDPG with HER**

One algorithm for finding the optimal policy of POMDP in continuous action space by improving the deterministic policy function $\pi$ is DDPG. DDPG is a model-free policy gradient method which requires no prior knowledge of the system. Comparing to other RL algorithms such as Deep Q-Network, DDPG has advantages in sample efficiency, action exploration, generalization, and reproducibility in solving problems with continuous high-dimensional action and state space. Specifically, DDPG maintains a parametric approximation $Q(s_t, a_t; \phi)$ to the action-value function $Q^\pi(s, a_t)$ and chooses $\phi$ to minimize

$$E_{(s_t, a_t, s_{t+1})} \left[ (Q(s_t, a_t; \phi) - r(s_t, a_t) - \gamma Q(s_{t+1}, a_{t+1}); s_t, a_t) \right]$$

where $\phi$ is determined by policy $\pi(s; \theta)$. Then the policy parameters $\theta$ are updated according to

$$\Delta \theta \propto E_{(s, a)} \left[ \frac{\partial}{\partial \theta} Q(s, a; \phi) \frac{\partial}{\partial \theta} \pi(s; \theta) \right]$$

The original DDPG algorithms added experience relay which collects experiences into a buffer and update $\theta$ and $\phi$ using random selection of mini-batch from the buffer. However, in a multi-goal environment, the agent usually failed to reach the goals. If agents only learn from success, the experience stored in buffer provides little toward the optimal policy. HER can learn from the failures as well. During failure episodes, HER considers whatever the agent reaches as a modified goal and stores them in the buffer as well. HER does great in a sparse rewards environment with multiple goals.

The network structure of implementation of DDPG with HER is shown in Figure 4. There are some modifications in the input layer where a predictor is added for feature extraction of the object via force sensors. The predictor using a technique called reparameterization trick which widely used in variational autoencoders. The predictor assumes a bunch of hidden variables $Z \sim \mathcal{N}(\mu, \sigma)$. The $\mu$ and $\sigma$ are predicted by the predictor and then a new input vector sampled from $Z$ will be transmit to the next layer as the prediction of the environments.

The algorithms were implemented in PyTorch with CPU on Windows platform. The training process utilized Message Passing Interface for distributed computation to enhance the training speed and the detailed algorithms are presented in Table 1.

**Simulation experiments**

The simulation environment ran in OpenAI Gym with MuJoCo physics engine and the algorithm was implemented in PyTorch. The source code of the environment and agent is available at https://github.com/ZhaolongGao/sc_ddpg.git.

- **Trail detail:** The environment runs in a stepwise mode. Each step contains 20 sub-steps sum up to 0.04 s which means the environment responds to the action control signal at a frequency of 25 Hz. At the beginning of each trial, the object is placed on the palm by chance. The object position and rotation are randomly selected in a preset range. The desired goal of the trial was selected randomly for each fingertip. During the trial, an action vector of 20-dimensional is fed into the environment at each step. The states of the robot and sensors will update afterward. The reward will be calculated as well. A sparse reward
function was chosen for better performance according to Plappert et al. The achieved goal is set to be the current fingertips positions. When there is no contact, the trial will be considered successful if the distance between achieved goal and desired goal is under the presetting threshold. When contacting with the object, the trial will be considered successful if the reading lies inside the boundary. Reward function will return 0 if the trial is successful, otherwise \(-1\).

Training: Multiple agents which instantiate the DDPG + HER were generated and attached to an environment separately. The parameters of the agents are sharing with each other during training. The training takes 200 more epochs with early stop if the evaluation successful rate stays above 0.9 for 5 epochs.

Evaluation: Evaluation will be done after each step. At the evaluation phase, the reparametrized hidden variables will not be sampled from the probability. The \(\mu\) learned from the trials will be used instead.

Results and discussions
The trained agents are tested with two scenarios: free movement and object grasping. In both scenarios, the desired goals are prearranged randomly by the environments which are observable to the agent. These goals will be replaced by the goals generated from user’s intention in practice. Meanwhile, the object position is unknown to the agent, which represents the unknown target in a grasping task.

In free movements, the desired goal was set as a step signal for all fingertips. The goal of each finger represents the desired goal of user. A result is shown in Figure 5, in which the dashed lines are the goals of corresponding fingertips and the solid lines are the positions of the fingertips. The average response time is about 4–6 steps (approximately 200 ms). As a study suggest that the delay should be less than 200 ms to maximize sense of body ownership. Thus, the response time is acceptable for application. Since the last steps are within a small range, the standability of the control algorithm is acceptable as well. This experiment aims to reproduce the situation when the prosthesis user’s movement in a pre-grasping phase. The target goals position represents the user’s plan, for example, the grasp locations, pre-grasping hand gestures, fingers usage, and so on. In this situation, the fingers of the prosthesis should always follow the goals. The results in Figure 5 indicate the trained agent can meet the daily use standard.

In the grasping scenario, the finger will try to approach the desired goal at the beginning of the trial. If the desired goal is inside the block beyond reach, the finger will contact the surface of the object. The force sensors attached to the surface of the fingertips are utilized to predict the unobservable information of grasped object in a lower space. Contacting force of three fingertip sensors during grasping are shown in Figure 6. The contact between fingertip and object happened at the timestep marked by the dashed line. After the contact, the force was kept by the algorithms subject to the force boundary and considered irrelevant to the desired goal until the desired goal leaves the boundary of the object. This will help the robot hand hold the object properly without consuming too much attention about

| Table 1. The implemented DDPG with HER algorithm. |
|-----------------------------------------------|
| Algorithm: DDPG with HER                       |
| Given:                                        |
| Environment E:                                |
| 1) Joint States \(S\)                         |
| 2) Force sensors \(\Omega\)                   |
| 3) Reward Function \(r\)                     |
| DDPG algorithms D                             |
| Initialize DDPG parameters                   |
| Initialize replay buffer \(R\)               |
| for episode=1, M do                           |
| Sample a goal \(g\), initial state \(s_0\) and |
| initial force observation \(o_{t_0}\)         |
| for \(t=0, T - 1\) do                         |
| Sample an action \(a_t\) with policy \(\pi\)  |
| from D:                                       |
| \(a_t \sim \pi(s_t, o_{t_1} || g)\)          |
| Execute the action \(a_t\) and get the new \(s_t\) and |
| \(o_{t+1}\)                                  |
| end for                                       |
| for \(t=0, T - 1\) do                         |
| \(r_t = r(s_t, a_t, g)\)                     |
| Store the transition \((s_t || g, a_t, o_{t_1} || g)\) in \(R\) |
| Sample a set of additional goals for replay \(G\) |
| for \(g' \in G\) do                          |
| \(r' = r(s_t, a_t, g')\)                     |
| Store the transition \((s_t || g', a_t, o_{t_1} || g')\) in \(R\) |
| end for                                       |
| end for                                       |
| for \(t=1, N\) do                            |
| Sample a minibatch \(B\) from the replay buffer \(R\) |
| Train D with minibatch \(B\) for one step     |
| end for                                       |

DDPG: Deep Deterministic Policy Gradient; HER: Hindsight Experience Replay.

Gao et al. 5
precise finger positions. This experiment tests the force control ability of the agent which plays a key role in grasping stability and safety. The lower bound of the contacting force is chosen to be the least force for stable grasping, and the higher one is the break force. The algorithm did not require explicit stability evaluation metrics; the stability can be implicit from the success rate of the trial. In this article, the success rate during the evaluation session is over 90.7%. The success rates during training and testing are shown in Figures 7 and 8 separately. We notice that the average success rate is different during grasp and free move, as shown in Figure 8. The higher success rate and lower variance in free move trials indicate that the stability of grasp can be further improved by enhancing the switch ability of the system. Since the switching between these two types of movements are fully controlled by agent which no specific parameter assignment is required, the sacrifice of success rate is acceptable comparing to the SC approaches with a specific structure for algorithm switching.

The proposed SC system outperformed human user when trying to achieve a proper contact force as shown in Figure 9. When the user was asked to control the finger directly, it takes more time before the contact force stabilized comparing to an SC agent. The gap between the human user and SC agent is mainly attributed to the lack of feedback with high embodiment other than visual clue. In other words, SC agent can enhance the embodiment of a prosthetic hand.

In summary, these results show that the algorithms introduced in this study is capable of achieve finger following
and stable grasping by combining the user intention and machine autonomy. In the meantime, the slightly reduced success rate is a trade-off of generalization in terms of switching between different modes.

Conclusions
In this article, we introduced an SC approach to the prosthesis grasping control problem. We formulated the SC approach as a partially observable MDP and tried to find its optimal policy by DDPG with HER. The algorithm is tested in a modified simulation environment which aimed to test the prosthesis in free move and object grasping task. The results showed that the algorithm is applicable with good performance.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Key R&D Program of China (2018YFB1307300) and the National Natural Science Foundation of China (91648207 and 61673068).

ORCID iDs
Zhaolong Gao https://orcid.org/0000-0001-8607-2862
Luyao Chen https://orcid.org/0000-0001-6048-9985

References
1. Melchiorri C and Kaneko M. Robot hands. In: Siciliano B and Khatib O (eds) Springer handbook of robotics. Berlin: Springer, 2016, pp. 463–480.
2. Shimoga KB. Robot grasp synthesis algorithms: a survey. Int J Rob Res 1996; 15: 230–266.
3. Miller AT and Allen PK. GraspIt! A versatile simulator for robotic grasping. IEEE Robot Autom Mag 2004; 11: 110–122.
4. Diankov R and Kuffner J. OpenRAVE: a planning architecture for autonomous robotics. Technical Report CMU-RI-TR-08-34, 79. Pittsburgh, PA: Robot Institute, 2008.
5. Geng T, Lee M, and Hüls M. Transferring human grasping synergies to a robot. Mechatronics 2011; 21: 272–284.
6. Ma RR and Dollar AM. On dexterity and dexterous manipulation. In: 2011 15th international conference on advanced robotics (ICAR), Tallinn, Estonia, 20–23 June 2011, pp. 1–7. IEEE.
7. Yousef H, Boukallel M, and Althoefer K. Tactile sensing for dexterous in-hand manipulation in robotics—a review. Sens Actuators A Phys 2011; 167: 171–187.
8. Huang J, Cao Y, Xiong C, et al. An echo state Gaussian process-based nonlinear model predictive control for pneumatic muscle actuators. IEEE Trans Autom Sci Eng 2019; 16: 1071–1084.
9. Ke A, Huang J, Chen L, et al. Fingertip tactile sensor with single sensing element based on FSR and PVDF. IEEE Sens J 2019; 19: 11100–11112.
10. Salisbury K. Issues in human/computer control of dexterous remote hands. IEEE Trans Aerosp Electron Syst 1988; 24: 591–596.
11. Kim HK, Biggs SJ, Schloerb DW, et al. Continuous shared control for stabilizing reaching and grasping with brain-machine interfaces. IEEE Trans Biomed Eng 2006; 53: 1164–1173.
12. Fong T, Thorpe C, and Baur C. Multi-robot remote driving with collaborative control. IEEE Trans Ind Electron 2003; 50: 699–704.
13. Backes PG. Multi-sensor based impedance control for task execution. In: Proceedings 1992 IEEE international conference on robotics and automation, Nice, France, 12–14 May 1992, pp. 1245–1250. IEEE.
14. Michelman P and Allen P. Shared autonomy in a robot hand teleoperation system. In: Proceedings of IEEE/RSJ international conference on intelligent robots and systems.
15. Losey DP, McDonald CG, Battaglia E, et al. A review of intent detection, arbitration, and communication aspects of shared control for physical human–robot interaction. *Appl Mech Rev* 2018; 70(1): 1–19.

16. Cipriani C, Zacone F, Micera S, et al. On the shared control of an EMG-controlled prosthetic hand: analysis of user–prosthesis interaction. *IEEE Trans Robot* 2008; 24: 170–184.

17. Iturrate I, Montesano L, and Minguez J. Shared-control brain-computer interface for a two-dimensional reaching task using EEG error-related potentials. In: 2013 35th annual international conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Osaka, Japan, 3–7 July 2013, pp. 5258–5262. IEEE.

18. Lopes AC, Nunes U, Vaz L, et al. Assisted navigation based on shared-control, using discrete and sparse human-machine interfaces. In: 2010 annual international conference of the IEEE Engineering in Medicine and Biology Society, Buenos Aires, Argentina, 31 August–4 September 2010, pp. 471–474. IEEE.

19. Saen M, Ito K, and Osada K. Action-intention-based grasp control with fine finger-force adjustment using combined optical-mechanical tactile sensor. *IEEE Trans Ind Electron* 2014; 61(8): 4026–4033.

20. Huang J, Ri S, Fukuda T, et al. An Inverted Pendulum System With Inverted Pendulum Systems. *IEEE Trans Ind Electron* 2020; 67: 2030–2041.

21. Lillicrap TP, Hunt JJ, Pritzel A, et al. Continuous control with deep reinforcement learning. In: *4th international conference on learning representation, ICLR 2016—conference track proceeding*, San Juan, Puerto Rico, 2–4 May 2016.

22. Andrychowicz M, Wolski F, Ray A, et al. Hindsight experience replay. In: *31st Conference on Neural Information Processing Systems (NIPS)*, Long Beach, CA, USA, 4–9 December 2017, pp. 5049–5059.

23. Brockman G, Cheung V, Pettersson L, et al. OpenAI Gym. *arXiv preprint arXiv:1606.01540* 2016; 1–4.

24. Todorov E, Erez T, and Tassa Y. MuJoCo: a physics engine for model-based control. In: *2012 IEEE/RSJ international conference on intelligent robots and systems*, Vilamoura, zPortugal, 7–12 October 2012, pp. 5026–5033. IEEE.

25. Reddy S, Dragan A, and Levine S. Shared autonomy via deep reinforcement learning. *arXiv preprint arXiv:1802.01744* 2018. DOI: 10.15607/rss.2018.xiv.005.

26. Nair V and Hinton GE. Rectified linear units improve restricted Boltzmann machines. In: *Proceedings of the 27th international conference on machine learning*, Haifa, Israel, 21–24 June 2010, pp. 807–814. Madison, WI: Omnipress.

27. Paszke A, Gross S, Chintala S, et al. Automatic differentiation in PyTorch. In: *31st Conference on Neural Information Processing Systems (NIPS)*, Long Beach, CA, USA, 4–9 December 2017, pp. 8024–8035.

28. Plappert M, Andrychowicz M, Ray A, et al. Multi-goal reinforcement learning: challenging robotics environments and request for research. *arXiv preprint arXiv:1802.09464* 2018; 1–16.

29. Ismail MAF and Shimada S. ‘Robot’ hand illusion under delayed visual feedback: relationship between the senses of ownership and agency. *PLoS One* 2016; 11: e0159619.