

Learning Form Closure Grasping with a Four-Pin Parallel Gripper

Rui Li , Shimin Liu  and Xiaojie Su *

School of Automation, Chongqing University, Chongqing 400040, China

* Correspondence: suxiaojie@cqu.edu.cn

Abstract: Being able to stably grasp with generalization is one of the distinguished capabilities for building a generic grasping system for robots. In this work, we propose a stable grasping method for four-pin parallel grippers within a reinforcement learning framework. First, a reinforcement learning problem is constructed on the basis of the improved four-pin gripper. Then, the learning policy and the reward function are constructed in consideration of the knowledge of environmental constraint and form closure. Finally, the effectiveness of the designed grasping method is validated in a simulated environment, and the results demonstrate that a safe and stable grasp can be planned for given 2.5D objects.

Keywords: four-pin gripper; form closure; learning to grasp

1. Introduction

Robotic manipulation is still a challenging task for the robotics community [1,2]. In recent years, progress in grasp detection has in some way improved the generalization of robotic grasping systems, enabling a wider application of such systems in pick-and-place tasks. For example, Mahler et al. reported on a picking system that is capable of grasping a diverse range of objects with a rate of more than 300 mean picks per hour [3]. Zeng et al. proposed a similar robotic pick-and-place system to grasp and recognize known and novel objects without the need to train for novel objects [4].

In addition to pick-and-place tasks, there is also a need for post-grasp manipulations [5,6]. Compared with pick-and-place tasks, the latter requires the object to be grasped in a known pose so that further actions can be performed with the object. For example, Paolini et al. proposed a data-driven statistical framework for post-grasp manipulation that enables robots to place, drop and insert [7]. Andrychowicz et al. implemented a reinforcement learning policy that can learn object reorientation on a physical shadow dexterous hand [8]. Cruciani et al. reported an in-hand manipulation benchmark to evaluate the planning and control of such systems [9].

To achieve the aforementioned manipulation tasks, a robotic hand or gripper is a prerequisite, and its design issue should be focused on. The design of robotic hands or grippers has developed immensely over the last few decades [10], evolving in the following aspects: (1) the number of fingers has extended from 2 to 3 [11], 4 [12] or 5 [13], (2) the actuation type has changed from fully actuated [14] to underactuated [15], (3) the material has varied from rigid [16] to soft [17], and (4) the task type has transformed from repetitive tasks to flexible tasks. In general, the developments benefit from the advancements in high-performance MCU, MEMS and materials. However, in real applications, a trade-off should still be considered between stability and flexibility. Four-pin grippers are usually treated as a compromise since they can achieve stable grasping of objects and adapt to changes, which makes them perfect for flexible industrial applications.

To guarantee stable post-grasp manipulation, several grasp quality evaluation concepts have been proposed [18]. Among them, form closure and force closure are commonly used when the object model is explicitly given. Considering the realization of these two concepts, force closure can be achieved by any two-pin gripper or two-fingered robotic hand, and



Citation: Li, R.; Liu, S.; Su, X.

Learning Form Closure Grasping with a Four-Pin Parallel Gripper. *Appl. Sci.* **2023**, *13*, 2506. <https://doi.org/10.3390/app13042506>

Academic Editor: Jose Machado

Received: 19 January 2023

Revised: 10 February 2023

Accepted: 14 February 2023

Published: 15 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

form closure can be achieved by at least seven-pin grippers or seven-fingered robotic hands. Although relocalizing the object to a desired pose using the former concept could be damaged by external disturbance forces caused by the environment, the former still seems to be a sensible solution for robotic hardware, since it is neither easy to plan seven contact points on an object nor feasible to design a seven-pin gripper or a seven-fingered hand. However, the latter concept would be more competitive with regard to confining the form closure problem to a plane and considering force closure in the orthogonal direction, thus enabling 2.5D grasping using four pins or fingers [12].

In previous work, a series of four-pin parallel grippers was designed to achieve 2.5D form closure grasping [5,12,19]. Combined with the form closure grasping algorithm, the grasping points can be planned for regular 2.5D objects in a clean experiment set-up. However, in the real grasping process, modeling of the uncertainty between the object, the robot and the environment is manageable, leading to failure when the geometry of the object or the gripper pins is not identical to the CAD models that are used to construct the environmental constraints.

In this work, our motivation is summarized as follows: (a) form closure grasping has practical value for post-grasp manipulation, (b) the synthetic form closure grasping method suffers from poor efficiency and unmodeled uncertainties, and (c) previous work can benefit from deep learning and reinforcement learning by encoding the state of the object into the autoencoder and the grasping strategy into the policy network.

Therefore we propose a four-pin form closure grasping method within the reinforcement learning framework. This method provides an end-to-end solution to achieve form closure grasps for vision-based robotic grasping systems. The contributions of this work are summarized as follows:

1. An improved parallel four-pin gripper design is presented. The four-pin gripper extends the parameter space of the previous designs, thus enabling more feasible configurations to be achieved.
2. A reinforcement learning (RL)-based robotic grasping scheme is presented. This scheme provides an end-to-end solution for scenarios where a grasp with a stronger closure is preferred.
3. An environmental constraint-based reward function for reinforcement learning is presented. This function provides a continuous score to evaluate the grasp quality rather than a binary value.

To the best of our knowledge, this is the first time that the form closure grasping for a four-pin gripper has been considered in an end-to-end framework with an environmental constraint-based reward function.

The remainder of this paper is organized as follows. In Section 2, the grasping problem is defined. In Section 3, the learning policy is described. Specifically, the environmental constraint-based reward function is discussed. In Section 4, experiments to verify the effectiveness of the proposed method are presented, while Section 5 contains the conclusions and plans for future work.

2. Problem Formulation

In this section, the gripper and the closure grasps are introduced. The formulations for expressing the form closure grasp as a reinforcement learning problem are also presented.

2.1. The Four-Pin Gripper

To achieve a planar form closure for an object, at least four contact points around the object should be provided. A four-pin gripper is a perfect fit to achieve form closure with a planar object. In the real world, although there are no planar objects, the four-pin gripper is still practical, since there are many column-like objects whose cross-sections are identical. With a four-pin gripper, we can still achieve form closure in the plane and, at the same time, force closure in the orthogonal direction along the column axis.

In industrial applications, the pneumatically driven four-pin gripper has demonstrated its value through its compact design and ease of control. Furthermore, to facilitate more flexible grasping configurations, motor-driven four-pin grippers have been suggested. The goal of this idea is to achieve the highest number of possible grasping configurations with the lowest number of actuators.

In [12], two versions of four-pin grippers were proposed, and the configuration space was greatly extended through the version upgrade. In this work, we present an improved design of the four-pin gripper on the basis of the CASIA V2 gripper, as shown in Figure 1. The gripper consists of four DC motors controlling four pins in a coupled manner: (1) the pins are grouped into left palm pins and right palm pins, (2) one palm motor controls the distance between two pins in one group with a gear mechanism, and (3) two center motors control the distance and relative pose between two groups of pins with a rack mechanism. The improvements of this design can be summarized as follows:

- The transmission of racks is replaced by belts to eliminate the backlash caused by racks.
- The connecting parts of the fingers are reinforced to resist any deformation during power grasping.

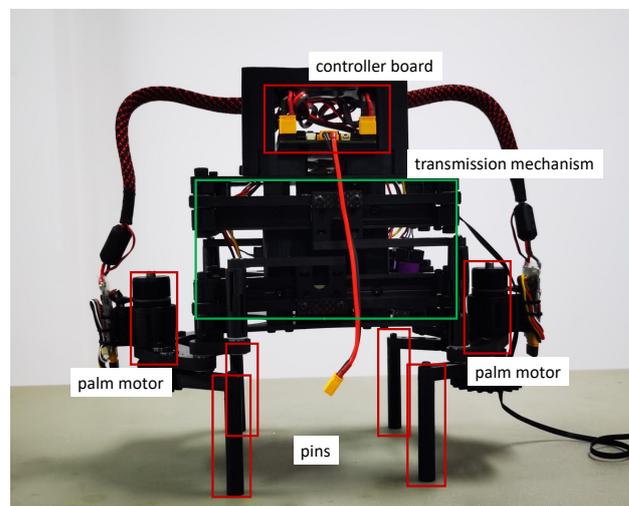


Figure 1. Illustration of the four-pin gripper.

2.2. Form Closure Grasps

Force closure and form closure are two concepts for describing the status of an object being constrained and therefore immobilized by force or form, or more precisely considering the forces or not. As mentioned in Section 1, we prefer form closure since it may provide “stronger” closure than force closure. A form closure grasp is a type of grasping strategy that aims to achieve form closure by robotic hands or grippers.

We may express the planar form closure grasp for the four-pin gripper with the following notations. Let $G \in \mathbb{R}^{n \times k}$ be the grasp matrix of the robotic gripper, where $n = 3$ is the number of dimensions for the planar case and $k = 4$ is the number of gripper pins (and the number of contact points on the object). For each contact point, if there always exists a non-negative coefficient $x_i, i = 1, 2, 3, 4$ such that $\sum_{i=1}^k G_i x_i = -f_{\text{ext}}$ holds for all $f_{\text{ext}} \in \mathbb{R}^n$, where G_i is the i th column of G , and f_{ext} is the external force and torque applied on the object, then the gripper configuration corresponding to G is defined as a form closure grasp.

In practice, it is not possible to check if a given G corresponds to a form closure with the above equation. Trinkle et al. proposed a first-order form closure test by constructing the above equation as a linear program (LP) [20]. However, it still remains a challenge to construct a G that corresponds to form closure (rather than just determining whether it is one or not). Among many of the related works, the environmental constraint-based method receives attention [12,19] since it not only extracts the form closure configurations but also provides

a grasp quality score to evaluate how “strong” this form closure is regarding breaking. In this work, we utilize the environmental constraint-based method to find the optimal form closure grasps. This method is used to construct the reward function as a priori knowledge.

2.3. The Learning Pipeline

Reinforcement learning has been used to build universal grasping systems in many recent papers. However, most of these works have focused on the generalization of the types of objects to be grasped, rather than ensuring a closure grasp that confines the pose of the object during the whole manipulation process. Deep convolutional neural networks have been leveraged to find the best edges on the object that are suitable for force closure grasps or power grasps.

In this work, we aim to implement a deep reinforcement learning method to construct a grasping system that is capable of grasping the object to a closed grasp status. Our focus is not only on grasping novel objects but also on grasping them firmly. While there are some similar works [21–23] in this direction, they only offered a sparse reward function, thus making it inefficient to learn such knowledge in the robotic grasping system.

To address this issue, we focus on improving the reward function. Utilizing our knowledge of form closure, we define a grasp quality score function to evaluate how far a grasp is from a good closure grasp.

The whole process of our proposed method is illustrated in Figure 2. We hoped to implement an end-to-end solution, and therefore we used the image *IM* as our input. The output was a set of manipulator and gripper actions $\{\tau_1, \tau_2, \dots, \tau_6, p_1, p_2, q, r\}$, assuming we were using a six-axis manipulator.

Owing to the high dimensionality of the image space, we believe that the mapping from the input image to the robot action can be learned through the learning framework.

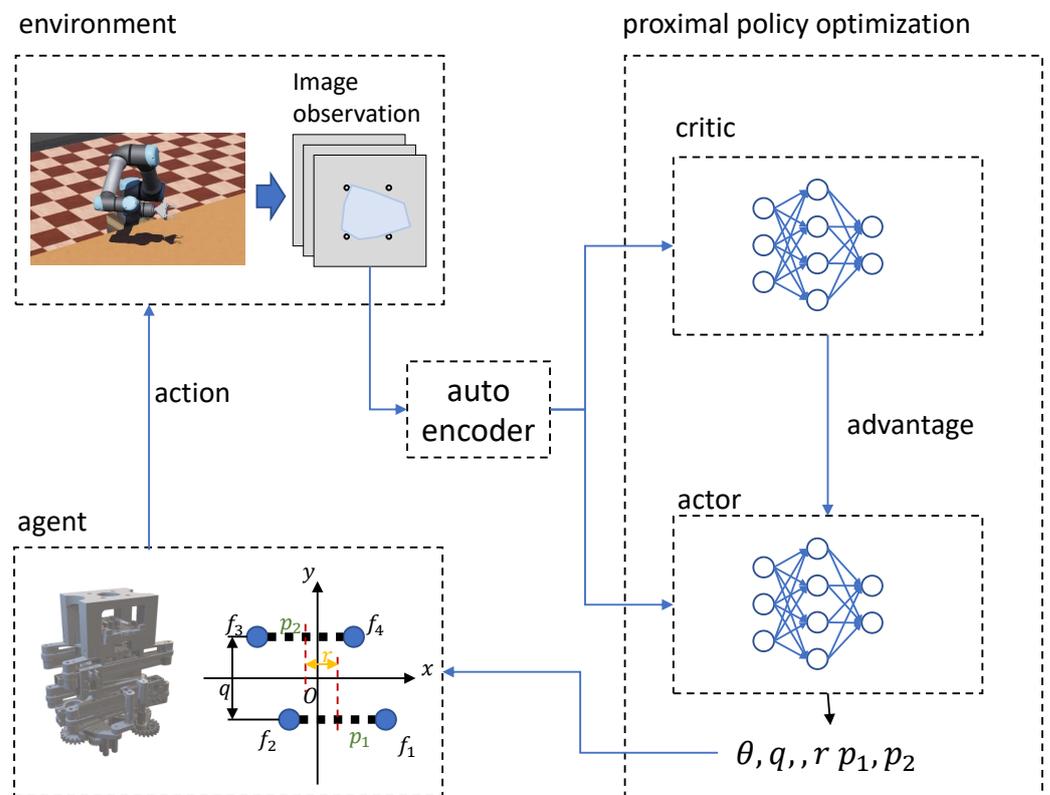


Figure 2. Full pipeline for training the four-pin gripper for form closure.

3. Method

In this section, we introduce the learning policy used in our grasping process.

3.1. Agent and Environment of the Learning Framework

To adapt to the reinforcement learning policy, we regarded the four-pin gripper as the agent and the interaction between the gripper and the object as the environment.

The gripper used in this work is depicted in Figure 3. Let $\{G\}$ be a fixed frame attached to the gripper, $\{W\}$ be the world frame fixed in the workspace and $\{B\}$ be a fixed frame attached to the center of mass (CoM) of the object.

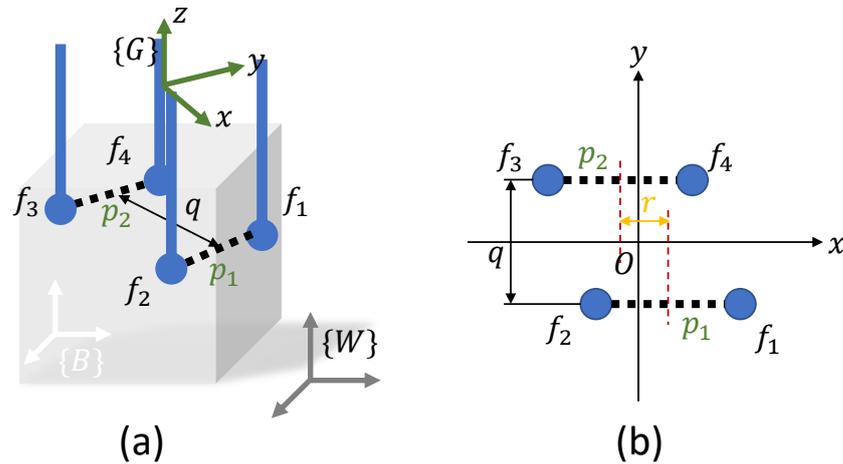


Figure 3. Illustration of the four-pin gripper and the grasping model. (a) The gripper pins and the object to grasp. (b) The 2D view of the gripper parameters p_1, p_2, q and r .

The positions of the four pins on the gripper are represented by the coordinates of the end points of the pins as $f_i = [x_i, y_i, z_i]^T, i = 1, 2, 3, 4$. We then define p_1 as the distance between f_1 and f_2 , p_2 as the distance between f_3 and f_4 , q as the distance between $\overline{f_1f_2}$ and $\overline{f_3f_4}$ and r as the distance between the middle lines of $\overline{f_1f_2}$ and $\overline{f_3f_4}$. In this way, we obtained a bidirectional mapping between the Cartesian space of the pins and the parameter space of the gripper:

$$\Pi : \{f_1, f_2, f_3, f_4\} \leftrightarrow \{p_1, p_2, q, r\}. \tag{1}$$

Meanwhile, the configuration c_{obj} of a 3D rigid body can be described by the element in $SE(3) = \mathbb{R}^3 \times SO$. Such an element can be parameterized by six numbers $(x, y, z, \theta, \phi, \gamma)$, where $(x, y, z) \in \mathbb{R}^3$ quantifies the position of the CoM of $\{B\}$ relative to $\{W\}$ and (θ, ϕ, γ) represents the Euler angles of the object pose by defining a mapping $g : (\theta, \phi, \gamma) \in \mathbb{R}^3 \rightarrow \mathbb{R}^{3 \times 3}$.

Without loss of generality, we assumed that most of the objects had finite stable states $S_i, i = 1, 2, \dots, Q$ at which they could stay still even after applying a disturbance wrench, where Q is the number of stable states that is dependent on the geometry and mass distribution of the object. We may express the configuration of an object as follows:

$$\begin{aligned} c_{obj}(S_i) &= c_{obj}(x, y, z^{S_i}, \theta^{S_i}, \phi^{S_i}, \gamma) \\ &= c_{obj}(x, y, \gamma | S_i), i = 1, 2, \dots, Q \end{aligned} \tag{2}$$

where z^{S_i}, θ^{S_i} and ϕ^{S_i} are invariable values determined by S_i . Using Equation (2), we may confine the number of parameters to describe the pose of a 3D object from 6 to 3, when the stable state is given. Following this approach, we may construct the environmental constraints for the four-pin grasping problem, which will be discussed in detail in Section 3.3.1.

3.2. Learning Policy Selection

Proximal policy optimization (PPO) [24] is one of the reinforcement learning policies that has achieved encouraging results in recent years. This policy utilizes the stochastic gradient descent (SGD) algorithm to find the optimal solution to the objective function. The clipping mechanism and mini batch trick are introduced to improve the effectiveness of the update process.

The idea of PPO is to learn a good policy from a good start and then improve the policy step by step with the clipping mechanism, which limits the maximum and minimum of the input to a set range. This can be seen as a conservative policy, but with better a priori knowledge, this policy can achieve good performance since it never runs too far away from the current good results.

To construct the loss function of the actor, the following equation was adopted:

$$L(s, a, \theta_o, \theta_n) = \min(rA^{\pi_{\theta_n}}(s, a), \text{clip}(r, 1 - \epsilon, 1 + \epsilon)A^{\pi_{\theta_n}}(s, a)), \quad (3)$$

where

$$r = \frac{\pi_{\theta_o}(a | s)}{\pi_{\theta_n}(a | s)} \quad (4)$$

is the important weight.

In the above equations, s and a represent the state and action of the agent, respectively, θ_o and θ_n represent the old and new policies, respectively, and $A^{\pi_{\theta_n}}$ is the advantage function measuring how much a certain action a is a good or bad decision, given a certain state s using the new policy θ_n .

In our case, we knew adequate information about the robotic grasping system (i.e., the model of the manipulator and the gripper, the environmental constraint-based algorithm for achieving form closure grasps and the image processing algorithms for extracting the contours of the objects). This means that we had good a priori knowledge of the learning problem. What we could benefit from thanks to the reinforcement learning framework is that it offers an end-to-end solution to integrate all of our knowledge into the system and the input image. Algorithm 1 shows the method used to train the PPO network. The original steps that required many computational resources could be replaced in this framework.

Algorithm 1 Training policy of the actor and critic networks.

Require: total episodes n_{total} , initial policy parameters θ_0

Ensure: trained models

- 1: Build actor;
 - 2: Build critic;
 - 3: **if** previous_run_exists **then**
 - 4: Load previous run weights W and episode count n ;
 - 5: **else**
 - 6: Initialize run weights W and episode count $n = 0$;
 - 7: **end if**
 - 8: **while** $n < n_{\text{total}}$ **do**
 - 9: Collect buffer of observations, actions and rewards by running policy $\pi_k = \pi(\theta_k)$ in the environment;
 - 10: Compute rewards \hat{R}_t using critic and reward function R_t ;
 - 11: Compute advantage estimates as $\hat{A}_t = \hat{R}_t - R_t$;
 - 12: Update actor using observations, advantages and actions;
 - 13: Update critic using observations and rewards;
 - 14: **end while**
-

3.3. Reward Function Design

As mentioned in the previous sections, to make the proposed method learn effectively, we constructed our reward function on the basis of a grasp quality score (GQS) function. Different from the existing works, the GQS function in this paper is continuous so that our agent will always learn something from every grasping attempt instead of obtaining a reward of zero in most cases, especially in the initial steps.

In this paper, the reward function is constructed as follows:

$$R = \begin{cases} f_{\text{GQS}}(IM, p_1, p_2, q, r) & \forall f_i \in \partial O \\ -\sum_{i=1}^4 l_i & \exists f_i \notin \partial O \end{cases} \quad (5)$$

where ∂O is the contour of the object and $\forall f_i \in \partial O$ indicates that all the pins of the gripper touch the surface of the object, while l_i is the minimal distance between the pins (that is not on the surface of the object) and the surface of the object. The GQS function takes the image IM and the gripper parameters as input and always outputs a positive value depending on the closure grasp that the contact points achieve. If a form closure grasp is achieved, then $f_{\text{GQS}} > 1$. If not, then $f_{\text{GQS}} \in [0, 1]$. When one or more of the gripper pins does not touch the surface of the object, there is no way to achieve any form closure grasp. However, we still offer a negative value to the learning agent to make it understand that the pins should always find their positions on the object.

3.3.1. Grasping Point Planning

When a subject interacts with an object, they form several environmental constraints in their configuration space. These constraints are usually detected and considered in obstacle avoidance algorithms, but they may help to design specific manipulation strategies in various cases. The utilization of environmental constraints is inspired by humans and can be observed in daily lives. For instance, when one tries to grasp a cup, he or she may rotate the cup on the table, moving the handle toward him or her and then grasping the handle.

In previous works, we found that several environmental constraints exist in four-pin grasping tasks. To visualize the constraints, we formulated the problem as follows.

Following the notations defined in Section 2, we defined the gripper parameter q as a function of the object pose c_{obj} and the rest of the gripper parameters $\{p_1, p_2, r\}$ as

$$q(x, \gamma | S_i) = \left\{ \min q | c_{\text{obj}} = (x, y^*, \gamma), (p_1, p_2, r) = (p_1^*, p_2^*, r^*) \right\} \quad (6)$$

where the notations with $*$ mean that these values are adjusted to obtain the minimum of q .

As shown in Figure 4, and following Equation (2), we depicted a typical environmental constraint region in the configuration space of objects, where we used a four-pin gripper to grasp an object with a square cross-section. In this figure, the parameters to describe the pose of the object x and γ were selected as the axes to show how the changes in these values would affect the value of the gripper parameter q .

There were many peaks and valleys in this region, the latter of which in fact corresponded to form closure grasp configurations according to [5]. One may also find that different valleys had different depths, which indicates how hard it is to break the form closure grasp if enough external forces are applied.

The environmental constraints provide a powerful toolkit to not only find the form closure configurations but also evaluate the quality of these candidates. In this work, we integrated the environmental constraints with the GQS function. Given a set of gripper parameters and a certain object configuration, if these values corresponded to a global minimum (“the deepest valley”), then we assigned a large score to the reward function. If the values corresponded to a sub-minimum, then we assigned a smaller positive score to

the reward function. Otherwise, the GQS would not apply, and a negative value would be assigned based on the sum of the distances between the pins and the object’s surface.

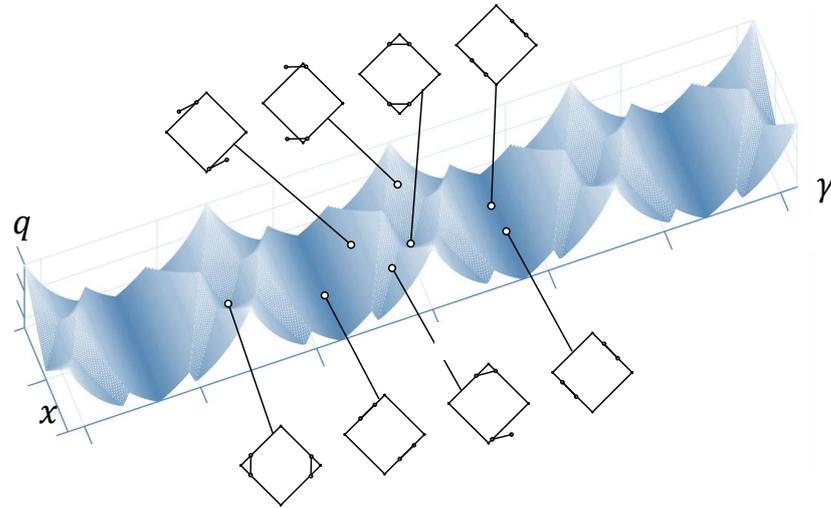


Figure 4. Illustration of the constraint region formed when the four-pin gripper grasps an object with a square cross-section.

3.3.2. Continuous Reward Construction

As mentioned in the above section, we aimed to construct a continuous reward. When there are any gripper pins that do not touch the surface of the object ($\exists f_i \notin \partial O$), the continuity of the function is guaranteed by the continuity of the distance function. When all of the pins touch the surface of the object, this does not always hold, since the valleys are separately distributed in the configuration space. To achieve continuity for $\forall f_i \in \partial O$, we constructed a function $\Gamma : (x, \gamma, q) \in \mathbb{R}^3 \rightarrow \mathbb{R}$ that mapped the local minima to a value in the range of $(0, \infty)$, where the global minimum always corresponds to the largest value. The rest of the local minima were mapped to a value in the range of $(0, 1]$ based on the ratio of $\frac{q_{sub}}{q^*}$.

The GQSs of three example objects are shown in Figure 5. As we expected, when all the gripper pins touched the surface of the object, it received a positive reward. Otherwise, the reward function returned a negative reward based on the distances between the pins and the surface of the object.

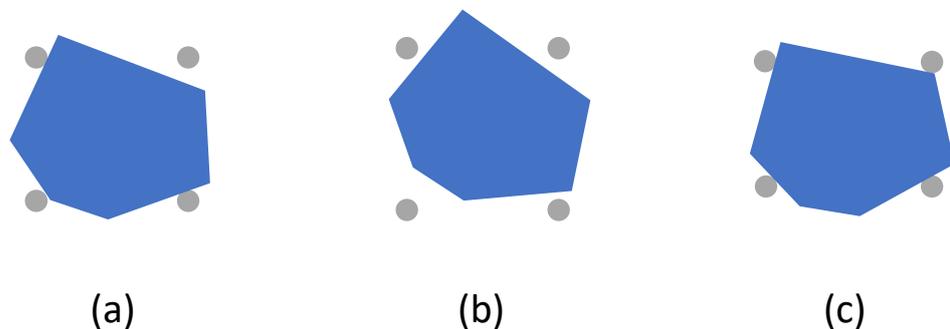


Figure 5. Example of rewards from the GQS function for different cases. (a) Some of the pins are on the surface of the object. Reward: -0.017 . (b) None of the pins are on the surface of the object. Reward: -0.048 . (c) All of the pins are on the surface of the object, but form closure is not achieved. Reward: 0.25 .

4. Experimental Validation and Discussion

4.1. Experimental Set-Up

In this section, we first validate the design of the gripper and then test the proposed method in a robot simulation environment.

To validate the design of the gripper, we fixed the gripper onto the table, manually set the object to be grasped to a form closure state and applied a random disturbance force to see if the object would escape from the form closure state.

To test the proposed method, we used Webots as our simulator due to its good performance, friendly interface and easy-to-use scripting system. With the help of URDF2Webots (<https://github.com/cyberbotics/urdf2webots> accessed on 13 February 2023), we could easily import the CAD model of the gripper illustrated in Figure 3 to the Webots simulator with minor revisions to the coordinate frames. A UR5e robotic manipulator was used in our experiment to drive the gripper from the initial pose to the target pose. To always maintain a clear view of the object without any occlusion, we attached an in-hand camera at the bottom of the gripper so that the object could always be fully observed during the grasping process.

To train the agent, we used a computing server with an Intel i7-12700F CPU, NVIDIA RTX3080 GPU and 32 GB of memory. The Webots simulator was run in fast mode, and rendering was off during the training process. We leveraged the Deepbots [25] framework to implement our environment as well as the learning algorithm due to its convenience in integrating the simulator and the RL algorithm. The robot-supervisor scheme was applied to provide full control capability during the training process.

4.2. The Prototype Experiment

We 3D printed eight objects to validate the effectiveness of the proposed gripper. As shown in Figure 6, all the objects shared a column-like structure. In this experiment, the objects would be grasped along the column axis, and the 2.5D form closure would be achieved. For each object, a random disturbance force in a certain range would be applied for 5 s. If the object stayed stable after the disturbance, then we believe that the design of the gripper achieved the goal of forming stable grasps.

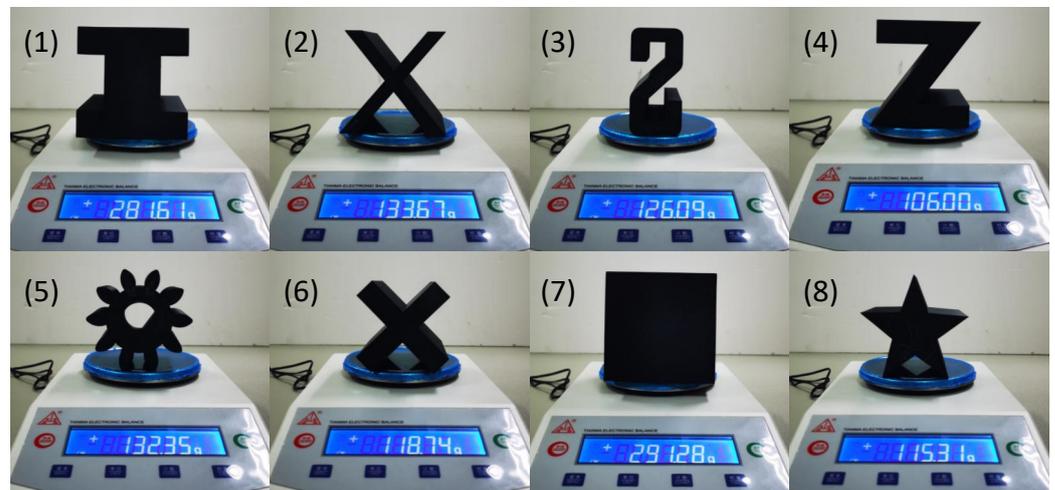


Figure 6. Eight objects were 3D printed to validate the effectiveness of the proposed gripper. (1) H shape, (2) X shape, (3) S shape, (4) Z shape, (5) gear, (6) cross, (7) cube and (8) star.

The results are shown in Table 1. All the objects could be grasped by the proposed gripper with a success rate of 95.0%. When a disturbance in the range of 0.0–1.0 N was applied, the gear and Z-shaped objects failed to maintain a form closure grasp, since the former had curved surfaces and the latter had limited high-score form closure configurations. When a disturbance in the range of 1.0–5.0 N was applied, the star and S-shaped objects could be grasped with a success rate of 95.0% for a similar reason.

Table 1. The success rate of the proposed four-pin gripper.

Object Name	Experiments [No. Successful Trials/No. Total Trials]		
	No Disturbance	1.0 N Disturbance	5.0 N Disturbance
H shape	20/20	20/20	20/20
X shape	20/20	20/20	20/20
S shape	20/20	20/20	19/20
Z shape	20/20	19/20	19/20
Gear	19/20	19/20	18/20
Cross	20/20	20/20	20/20
Cube	20/20	20/20	20/20
Star	20/20	20/20	19/20

4.3. The End-to-End Grasping Simulation

In this simulation experiment, we trained the gripper to grasp 1000 randomly generated objects in a form closure manner using the proposed method.

We set the parameters for the autoencoder and the PPO as follows. (1) For the autoencoder, we set the input image size to $256 \times 256 \times 3$ and the latent size to 128. We used a convolutional variational-type autoencoder. The size of the dataset was 120,000, which indicates that there were 120,000 randomly generated top-view images of objects extracted by the in-hand camera. (2) For the PPO, we set the buffer size to 512, batch size to 64 and layer size to 256. We used two hidden layers in the actor and critic networks. The loss clipping was set to 0.2, and the step for each episode was set to 1. The average training time was approximately 30 h, and the model size was approximately 110 M.

We first trained an autoencoder which aimed to extract the object pose from the raw image. To speed up the training process, we scaled the image size down to $256 \times 256 \times 3$. Since we were in a simulated environment, the autoencoder showed good results in recognizing the poses of the objects.

We then trained the reinforcement learning model using the PPO policy. The model was trained over 1000 epochs, and we observed that the training results converged. Using the latest trained weight, we observed that the manipulator could grasp an object using the four-pin gripper after the training process, similar to the prototype experiment.

We compared the experimental results with our previous work [12], in which a synthetic method was adopted to grasp the object as shown in Table 2. Since we performed the experiment in a simulation environment, we extended the number of test objects to 100. We followed the same experimental process in the previous work to find the success rate and also the average planning time of the proposed grasping method. We observed that the proposed method in this paper achieved a success rate of 94.4% without a disturbance and 84.8% with a 5 N disturbance, which was lower than the results in [12]. This was because the end-to-end framework brought uncertainty in various processes (e.g., the autoencoder bringing in errors during the pose estimation of the object and the actor network bringing in errors during the control of the gripper). However, there was an improvement in adaptation to the number of objects, and the average planning time was greatly reduced. A better result may be expected if the resolution of the image is increased.

Table 2. Comparison of the methods.

Methods	No. Objects	Success Rate		Avg. Planning Time [s]
		No Disturbance	5.0 N Disturbance	
[12]	12	57/60 (95.0%)	54/60 (90.0%)	12.0
This work	100	472/500 (94.4%)	424/500 (84.8%)	3.1

5. Conclusions and Future Work

Form closure grasping has practical value for post-grasp manipulation. However, the synthetic form closure grasping method suffers from poor efficiency and unmodeled uncertainties. In this work, we extended our previous work from the aspects of the hardware and method: (a) we introduce an improved design of a four-pin gripper which was optimized for immobilizing objects relative to the gripper by form closure, and (b) we used the proximal policy optimization (PPO) algorithm to train our gripper to grasp any given object in the way of forming a closure by form. With this work, our previous work can benefit from deep learning and reinforcement learning by encoding the state of the object into the autoencoder and the grasping strategy into the policy network.

Since we have adequate a priori knowledge regarding the whole grasping process, this algorithm is good at training a satisfactory strategy with slight adjustment to a good result at the initial steps. Since the environmental constraint-based grasping quality score function requires many computational resources, the trained model is expected to encode this knowledge in the neural network, thus speeding up the grasping process.

In the experiment section, we implemented a real-world experiment regarding the effectiveness of the gripper design and a comparison between the proposed method in a simulation and our previous work [12] from the aspects of the success rate and average planning time. The comparison result showed that the proposed method had the advantage of improving the efficiency of the previous work. We will try to transfer the learning results from the simulation to a real-world set-up using transfer learning policies in our future work. Our future work will also focus on developing a complete grasping system that could fully utilize the four-pin gripper for dexterous manipulation tasks.

Author Contributions: Conceptualization, R.L.; methodology, R.L.; software, R.L.; validation, R.L.; formal analysis, R.L.; investigation, R.L. and S.L.; resources, R.L. and X.S.; data curation, R.L. and S.L.; writing—original draft preparation, R.L.; writing—review and editing, R.L.; visualization, R.L.; supervision, X.S.; project administration, X.S.; funding acquisition, X.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was financially supported by the National Key R&D Program of China, Project Number 2022YFE0107300, by the National Natural Science Foundation of China (Grant No. 62003059), by the Key-Area Research and Development Program of Guangdong Province (Grant No. 2020B0909020001), by the China Postdoctoral Science Foundation (Grant No. 2020M673136) and by the Chongqing Postdoctoral Research Project Special Grant (XmT2020123).

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank Xiaoyu Ma for his help with the formal analysis and resources in this paper.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

2.5D	2.5 dimension
GQS	Grasp quality score
PPO	Proximal policy optimization

References

1. Zhang, H.; Tang, J.; Sun, S.; Lan, X. Robotic Grasping from Classical to Modern: A Survey. *arXiv* **2022**, arXiv:2202.03631.
2. Li, R.; Qiao, H. A Survey of Methods and Strategies for High-Precision Robotic Grasping and Assembly Tasks—Some New Trends. *IEEE/ASME Trans. Mechatron.* **2019**, *24*, 2718–2732. [[CrossRef](#)]
3. Mahler, J.; Matl, M.; Satish, V.; Danielczuk, M.; DeRose, B.; McKinley, S.; Goldberg, K. Learning ambidextrous robot grasping policies. *Sci. Robot.* **2019**, *4*, eaau4984. [[CrossRef](#)] [[PubMed](#)]

4. Zeng, A.; Song, S.; Yu, K.T.; Donlon, E.; Hogan, F.R.; Bauza, M.; Ma, D.; Taylor, O.; Liu, M.; Romo, E.; et al. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. *Int. J. Robot. Res.* **2019**, *41*, 690–705. [[CrossRef](#)]
5. Ou, Z.; Qiao, H. Analysis of Stable Grasping for One-Parameter Four-Pin Gripper. In *Intelligent Robotics and Applications*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 630–639. . 68. [[CrossRef](#)]
6. Su, J.; Liu, C.; Meng, Y. Immobilizing Caging Grasps of Convex Polyhedrons with a Four-Pin Gripper. *IEEE Robot. Autom. Lett.* **2021**, *6*, 7683–7690. [[CrossRef](#)]
7. Paolini, R.; Rodriguez, A.; Srinivasa, S.S.; Mason, M.T. A data-driven statistical framework for post-grasp manipulation. *Int. J. Robot. Res.* **2014**, *33*, 600–615. [[CrossRef](#)]
8. Andrychowicz, M.; Baker, B.; Chociej, M.; Józefowicz, R.; McGrew, B.; Pachocki, J.; Petron, A.; Plappert, M.; Powell, G.; Ray, A.; et al. Learning dexterous in-hand manipulation. *Int. J. Robot. Res.* **2019**, *39*, 3–20. [[CrossRef](#)]
9. Cruciani, S.; Sundaralingam, B.; Hang, K.; Kumar, V.; Hermans, T.; Kragic, D. Benchmarking In-Hand Manipulation. *IEEE Robot. Autom. Lett.* **2020**, *5*, 588–595. [[CrossRef](#)]
10. Piazza, C.; Grioli, G.; Catalano, M.; Bicchi, A. A Century of Robotic Hands. *Annu. Rev. Control. Robot. Auton. Syst.* **2019**, *2*, 1–32. [[CrossRef](#)]
11. Hasan, M.R.; Vepa, R.; Shaheed, H.; Huijberts, H. Modelling and Control of the Barrett Hand for Grasping. In Proceedings of the 2013 UKSim 15th International Conference on Computer Modelling and Simulation, Cambridge, UK, 10–12 April 2013; IEEE: New York, NY, USA, 2013. [[CrossRef](#)]
12. Li, R.; Cao, Y.; Bing, Z.; Qiao, H. An Improved Four-Pin Gripper for Robust 2.5-D Form-Closure Grasp. *IEEE/ASME Trans. Mechatronics* **2022**, 1–12. . [[CrossRef](#)]
13. Kim, Y.J.; Yoon, J.; Sim, Y.W. Fluid Lubricated Dexterous Finger Mechanism for Human-Like Impact Absorbing Capability. *IEEE Robot. Autom. Lett.* **2019**, *4*, 3971–3978. [[CrossRef](#)]
14. Liu, H.; Wu, K.; Meusel, P.; Seitz, N.; Hirzinger, G.; Jin, M.; Liu, Y.; Fan, S.; Lan, T.; Chen, Z. Multisensory five-finger dexterous hand: The DLR/HIT Hand II. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; IEEE: New York, NY, USA, 2008. [[CrossRef](#)]
15. Ma, R.; Dollar, A. Yale OpenHand Project: Optimizing Open-Source Hand Designs for Ease of Fabrication and Adoption. *IEEE Robot. Autom. Mag.* **2017**, *24*, 32–40. [[CrossRef](#)]
16. Tuffield, P.; Elias, H. The Shadow robot mimics human actions. *Ind. Robot. Int. J.* **2003**, *30*, 56–60. [[CrossRef](#)]
17. Puhlmann, S.; Harris, J.; Brock, O. RBO Hand 3: A Platform for Soft Dexterous Manipulation. *IEEE Trans. Robot.* **2022**, *38*, 3434–3449. [[CrossRef](#)]
18. Murray, R.M.; Li, Z.; Sastry, S.S. *A Mathematical Introduction to Robotic Manipulation*; CRC Press: Boca Raton, FL, USA, 2017. [[CrossRef](#)]
19. Li, X.; Qian, Y.; Li, R.; Niu, X.; Qiao, H. Robust form-closure grasp planning for 4-pin gripper using learning-based Attractive Region in Environment. *Neurocomputing* **2020**, *384*, 268–281. [[CrossRef](#)]
20. Trinkle, J. A Quantitative Test For Form Closure Grasps. In Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Raleigh, NC, USA, 7–10 July 1992. [[CrossRef](#)]
21. Elahibakhsh, A.; Ahmadabadi, M.; Sharifi, F.; Araabi, B. Distributed form closure for convex planar objects through reinforcement learning with local information. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sendai, Japan, 28 September–2 October 2004; IEEE: New York, NY, USA, 2004. [[CrossRef](#)]
22. Jiang, Y.; Moseson, S.; Saxena, A. Efficient grasping from RGBD images: Learning using a new rectangle representation. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; IEEE: New York, NY, USA, 2011. [[CrossRef](#)]
23. Jiang, Y.; Amend, J.R.; Lipson, H.; Saxena, A. Learning hardware agnostic grasps for a universal jamming gripper. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; IEEE: New York, NY, USA, 2012. [[CrossRef](#)]
24. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms, 2017. *arXiv* **2017**, arXiv:1707.06347.
25. Kirtas, M.; Tsampazis, K.; Passalis, N.; Tefas, A. Deepbots: A Webots-Based Deep Reinforcement Learning Framework for Robotics. In *IFIP Advances in Information and Communication Technology*; Springer International Publishing: Cham, Switzerland, 2020; pp. 64–75. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.