2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 23-27, 2022, Kyoto, Japan

Development of Low-Inertia Backdrivable Arm Focusing on Learning-Based Control

Manabu Nishiura[1], Akira Hatano[1], Kazutoshi Nishii[1], and Yoshihiro Okumatsu[1]

***Abstract*— A robot designed to coexist and work with humans in the same workspace should be able to work at the same speed as humans and have safe contact with humans and with the environment. However, when a robot arm has been given flexibility through mechanisms and controls for the purpose of coexistence, it is difficult for it to perform tasks at the speed and accuracy desired by humans if it is moved simply by using conventional position-based controls. With such an arm, we consider that the use of learning-based control is necessary to achieve both safety and speed. Therefore, we prototyped a low-inertia, high-backdrivability arm as a platform for studying learning-based control and tested two types of learning-based control. This paper describes our design process, in which hardware suitable for learning-based control was developed according to the requirements of the specific task. It also presents the results of our evaluation experiments, in which tasks involving quick movements and motion requiring physical contact with an object were performed using learning-based control.**

I. Introduction

Robots are constantly being introduced into people's lives, contributing to their wealth and well-being. In particular, there is increasing demand for the use of robots in professional nursing care and at home because of the declining birthrate and an aging population, which has become a matter of concern in many developed countries, including Japan. Most robots currently used in such environments are designed for cleaning, sterilizing, and serving tasks or for communication. These types of robots are not typically used in activities that involve physical contact with people or the environment (other than for limited work and in environments such as assembly plants [1]). Actuation is a key factor in this problem. In many industrial applications, humans and robots are separated by barriers to ensure human safety. In a normal environment, however, where humans will use and touch robots without having been trained in robot safety, safety must be ensured in other ways to allow human–robot coexistence. In addition to the matter of safety, it is necessary for a robot to be able to move at sufficiently high speeds (with respect to human movements) without sustaining damage. Recently, collaborative robots have been used to achieve collaborative work in factories. However, the task speeds are quite slow because the environment is limited in terms of the tasks and the people who can coexist with the robot, and the robot needs to be slowed when a person is nearby.

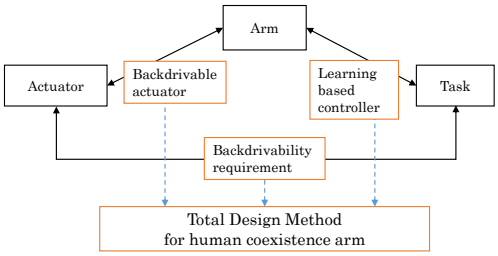


Fig. 1: Overview of the design process.

Therefore, in this study we aimed to develop a manipulator (arm) that achieves both safety and operating speed, representative of a robot designed for human–robot coexistence. We consider that flexibility of the manipulator is necessary to provide safety; we aimed to achieve such flexibility by having both high backdrivability and low inertia. A suitable controller for this flexible arm would also be required. Conventionally, means such as impedance control have been used for controlling arms when physical contact is involved. In recent years, however, learning-based controllers have shown some success in tasks involving many contacts that are difficult to model [2]. Therefore, our objective was to develop a flexible arm, a learning-based controller, and an overall design process that can comprehensively handle both an arm and a controller (Figure 1).

In this study, we considered actuators, arms, and control algorithms that can perform human-equivalent tasks, with the goal of developing an arm that can coexist with humans. We developed a prototype arm with three degrees of freedom (DOFs) and having high backdrivability and low inertia, and we evaluated it with tasks that involve physical contact at speeds higher than those typically used by traditional rigid robots.

The remainder of this paper is organized as follows. Section II discusses related work. In Section III, appropriate levels of backdrivability for flexible arm configurations are discussed. Section IV describes the configuration of a backdrivable, low-inertia, 3-DOF arm. Learning-based control of the flexible arm for tasks involving mechanical contact is described in Section V, and the paper concludes with Section VI.

II. Related Work

Various studies have been conducted to provide flexibility to robotic arms and perform tasks involving mechanical contact. Industrial robots require high rigidity and low backlash,

[1]M. Nishiura, A. Hatano, K. Nishii, and Y. Okumatsu are with the Frontier Research Center, Toyota Motor Corporation, 543 Kirigahora, Nishihirose-cho, Aichi, Japan, e-mail: {manabu_nishiura, akira_hatano_aa, kazutoshi_nishii, yoshihiro_okumatsu}@mail.toyota.co.jp

2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 978-1-6654-7927-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/IROS47612.2022.9981337

and wave gears are often used, as their purpose is to perform accurate positioning. Although this configuration has low backdrivability, the backdrivability can be increased by attaching a torque sensor for each axis [3] [4]. However, torque sensors are expensive and easily damaged, and the response delay of the sensor feedback limits the responsiveness of the arm as a whole. Therefore, a method for estimating the torque from the torsional rigidity of the reducer that relies on an arrangement of encoders on the input and output side rather than a torque sensor is also being actively researched [5]. The series elastic actuator (SEA) is also sometimes used; it has an elastic element attached in series to the actuator. However, it is not suitable for high-speed operation because the responsiveness of the arm is determined by the characteristics of the elastic element [6] [7].

Because the main cause of the decrease in backdrivability is the reducer, the direct-drive method has also been researched and developed as a single actuator. However, there are almost no practical examples with robot arms, owing to the low torque density of such actuators. In addition, because a direct-drive motor provides insufficient torque density and a wave gear has high friction, there have been studies investigating quasi-direct-drive (QDD) actuators using a planetary reducer with a reduction ratio of approximately 10:1 and a low-speed high-torque motor [8] [9].

Research on the backdrivability of actuators has been conducted for both manipulators and legged robots. The backdrivability required to absorb the impact of landing has been optimized across the actuator and leg structures. In addition, studies have been undertaken to achieve faster running and higher flying [10] [11]. Other studies have been conducted to improve the backdrive performance of reducers [12], and some previous studies used hydraulic pressure [13].

Another common approach is to arrange an actuator at the base of the arm and transmit power to each axis using a power transmission mechanism to reduce the inertia of the arm. A typical method is the wire drive method, and several studies using this technique have been reported [14] [8]. Problems of the wire drive system include its rigidity, which is lower than that of an industrial robot, and the reliability and maintainability of the wire in actual operation. Therefore, a method was proposed in which only the part near the base of the arm is used as a belt mechanism [15].

One study investigated the realization of impedance at the control level; a torque-controlled humanoid was used to perform full-body movements [16]. Using a different approach, another study realized push recovery by absorbing shocks, accomplished by cycling control extremely quickly [17]. Variable stiffness control using soft hardware has been achieved by learning of the mapping of the muscle and joint space for redundant muscle arrangements [18].

However, none of these studies were able to accomplish a task at adequate speed in a human–robot-coexistence environment. To develop an arm that can achieve this, it is necessary to accomplish the following: 1) define the backdrivability required for the task to be performed and realize the appropriate actuators, 2) design and implement the arm with the arm inertia required for coexistence with humans and for task execution, and 3) develop and implement a method of control for a flexible arm with low inertia and high backdrivability.

In this paper, we describe the basic approaches to these problems for two relatively simple tasks (hammering and whiteboard erasing) and show that the hardware can be designed and controlled in practice.

III. Actuator Selection Based on Required Backdrivability

A. Backdrivability

In this section, we discuss the backdrivability required for an arm suitable for machine learning from the assumed task. First, we define backdrivability, which in the context of robotics refers to the ease of moving the actuator from the output side. The backdrive torque T_f $[\mathrm{N\,m}]$ is defined as

$$T_f = T_s + I_a\ddot{\theta} + C_a\dot{\theta} + K_a\theta, \tag{1}$$

where T_s is static friction torque and cogging torque, θ is the output axis angle, K_a is the rigidity, C_a is the viscosity, and I_a is the inertia.

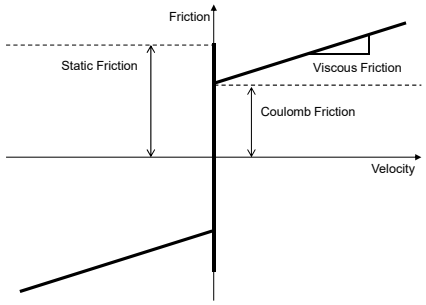


Fig. 2: Simplified model of static and viscous friction.

The model shown in Figure 2 is known as a simplified model of static and viscous friction, which treats the difficulty of turning from the output shaft as the torque due to friction. However, the actual torque of the backdrive is more complex. The moment of inertia is proportional to the acceleration, and the moment of inertia of the motor is proportional to the square of the deceleration ratio when it is turned from the output shaft. Therefore, this term becomes dominant, especially in actions that involve large acceleration, such as the landing in a jump action. In addition, it is known that the magnitude of the backdrive torque depends on the position of the moving rotor because the T_s term contains nonlinear elements such as cogging torque, which varies with the angle of the rotor [19]. Therefore, the term in the backdrive torque equation that has the greatest effect in the arm task operation of interest must accordingly be defined.

B. Basic Strategy for Achieving Optimal Backdrivability

Here we outline how the backdrivability required was determined for the tasks assumed in this study.

The tasks envisioned were hammering and whiteboard erasing. Each of these is explained in detail in Section V. For the hammering task, the goal is not only to have the flexibility necessary for quick movements but also not to break the hardware during the learning process. The whiteboard erasing task requires the arm to push the eraser against the board with the correct amount of force. The backdrivability requirements for the arm were considered for the whiteboard erasing task because whiteboard erasing requires more delicate force control than hammering. The arm must have the flexibility to allow the whiteboard eraser to be placed on the whiteboard and the resolution to detect minute forces. If the arm holding the whiteboard eraser can detect and control the force with which it can move by touching the whiteboard, we assume that the arm can adjust the force subtly, as a person does. Thus, the minimum force needed to initiate motion while pressing against the whiteboard eraser was determined as the required backdrivability of the actuator. The torque was determined from the payload required for the arm, and an appropriate actuator was selected according to its rated torque and static friction. The friction of the arm includes motor friction and joint friction, but joint friction is only that from bearings and can be ignored. Therefore, in this study, only the static friction of the motor was considered.

C. Actuator Selection

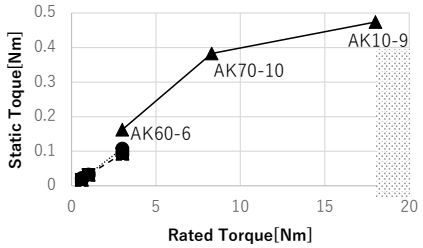
We evaluated motors from T-motor, which offers lines of inner rotor, outer rotor, and quasi-direct-drive actuators for robots. The static friction torque was measured for three motor types from each of three T-motor series: the G series and the GL series of outer rotors, and the AK series of quasi-direct-drive actuators, each with different outer diameters. Table I shows the specifications and measurement results for each motor. Figure 3 shows static friction torque (vertical axis) plotted against rated torque (horizontal axis). The torque and static friction torque were proportional for the G series and GL series, which do not have reduction gears. This can be attributed to the fact that the ratio of the maximum cogging torque to the rated torque is constant because the basic motor design is the same. The AK-series motors have different reduction ratios, and the static friction increased in proportion to the reduction ratio. Moreover, it is believed that the ratio of the static friction torque to rated torque is not linear because the AK70-10 has a reduction ratio of 10:1, which is the largest of the three types.

In the whiteboard erasing task, the force required to start the movement from the pressed state was measured and taken as the minimum resolution. This value was approximately 2 N. The torque converted to the actuator output shaft was 1.14 $\mathrm{N\,m}$ because the arm length was 568 mm. The minimum static friction force was set to 0.4 $\mathrm{N\,m}$, considering the arm length when it was bent in the actual working area. The maximum payload of the arm was set to 4 kg. Therefore, the rated torque of the actuator was 18 $\mathrm{N\,m}$. From these values, the optimal actuator was determined to have a static friction torque of 0.4 $\mathrm{N\,m}$ or less and a rated torque of 18 $\mathrm{N\,m}$ or more, as shown in Figure 3a. As can be seen, none of the actuators measured have values in this region. However, the AK70-10 and AK10-9 are close to this region, and we selected the AK70-10 because we prioritized backdrivability.

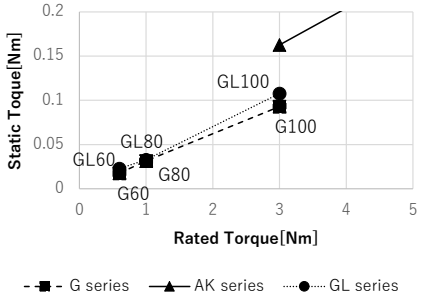
TABLE I: Motor specifications and measurements.

	G60	G80	G100	GL60	GL80
Rerated torque $[\mathrm{N\,m}]$	0.6	1	3	0.6	1
Mass [g]	364	482	954	226	315
Static friction torque (measured) $[\mathrm{mN\,m}]$	18	32	93	23	33

	GL100	AK60-6	AK70-10	AK10-9
Rerated torque $[\mathrm{N\,m}]$	3	3	8.3	18
Mass [g]	698	315	521	820
Static friction torque (measured) $[\mathrm{mN\,m}]$	108	163	383	474



(a) AK series



(b) G series and GL series

Fig. 3: Measured static friction torque plotted against rated torque.

IV. Low-Inertia Backdrivable Arm

Figure 4 shows the appearance of the robotic arm, and Table II lists its specifications. The robot has three DOFs: shoulder yaw, shoulder pitch, and elbow pitch. The motors are positioned near the shoulder axis to reduce the moment of inertia of the links; therefore, the elbow pitch axis is driven by timing belts. The reflected inertia of the motor is proportional to its rotor inertia and the square of the gear ratio. A low gear ratio of 10:1 was selected to reduce this inertia and thereby improve backdrivability.

V. Learning-Based Control Experiments on Prototype 3-DOF Arm

In this section, we describe the application of learning-based control to the prototype 3-DOF arm and show using experimental results from a real robot that it is possible for

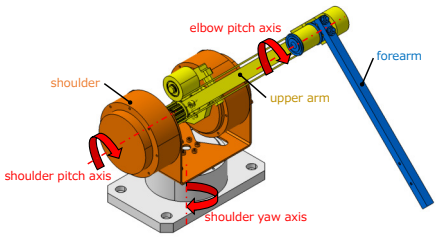


Fig. 4: Appearance of the prototype robot.

TABLE II: Specifications of the prototype robot.

	Upper arm	Forearm	Shoulder
Mass [g]	94	350	1,278
Inertia (link only) [kg mm^2]	366	6,724	4,725
Inertia (with rotor) [kg mm^2]	4,143	10,501	8,502
Length [mm]	223	290	55

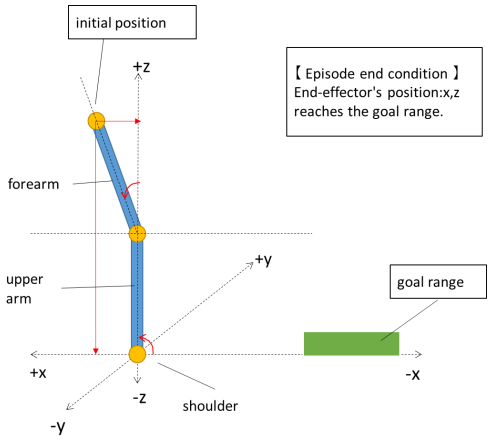


Fig. 5: Setup for experiment with hammering using reinforcement learning.

our robot to acquire contact-rich motion and quick movements. For learning-based control, we chose two methods: unsupervised learning and supervised learning. For each learning method, we wished to confirm that it is possible to search for a controller on our real robot. Two tasks were selected for learning: (1) a hammering motion, representing a quick type of motion, and (2) a rapid whiteboard-erasing motion, representing a contact-rich, rapid type of motion.

In tasks that involve large amounts of contact, it is necessary to go beyond position control and enable the arm to acquire force manipulation capabilities to generate trajectories in advance in order to respond flexibly to situational changes. Thus, with the whiteboard erasing task, it is necessary for the arm to learn the manipulation of force in addition to simple positioning. Therefore, we included the value of the torque in commands generated from the learning-based controller to show that it is possible for the arm to learn force operations in place of simple positioning commands.

A. *Unsupervised Learning: Reinforcement Learning*

We conducted an experiment to enable the prototype 3-DOF arm to acquire a hammering motion using reinforcement learning and verify unsupervised learning. In this experiment, we wished to accomplish and analyze the following:

- To acquire a quick movement using only the actual device.
- To acquire a stable trajectory by torque control.

In acquisition learning that involves contact with the environment, it is common to use simulation only for learning the behavior, using the actual robot for learning robustness. This approach is used because too much input during the learning process can lead to hardware failure. However, contact between the robot and environment is unavoidable, even during the learning of robustness using the real robot, because it is difficult to accurately simulate the behavior of a real robot. Our aim with this experiment was to confirm that the hardware could learn without failure even in the early stages of learning with more contact. Therefore, we chose to use the hammering task, which involves more intense contact with the environment, because the training for this task used only real robots.

Position control is suitable when a robot is acquiring a stable trajectory. However, when a task involves contact with the environment or an object, external forces can cause deviation from the target position, resulting in the generation of excessive force by general position control; therefore, torque control is necessary.

Torque control has been used in a few cases when there are concerns about stability associated with contact with the environment or objects, whose characteristics are often unknown. For the arm to move stably under torque control, the load of friction and self-weight, which varies with the posture of the arm, must be taken into account. Conventional arms are strongly affected by friction and deadweight. Therefore, the torque and friction must be estimated and compensated for with high accuracy. There have been studies to improve the accuracy of friction estimation [20] and self-weight compensation according to arm posture [21], but the methods they used have drawbacks, such as the need for adaptation and their computational cost. Such measures are not needed with the prototype 3-DOF arm, however; friction estimation is unnecessary because the effects of motor friction and the weight of the robot are small. Therefore, the learning process was performed using only torque control to confirm that the robot can acquire a stable trajectory even with torque control and without compensating for friction and deadweight.

1) Setup and Overview of the Experiment: We defined the coordinates as shown in Figure 5. The shoulder yaw axis was fixed and controlled such that it did not move in the y-axis direction. To simplify the problem, the end effector did not actually hit a nail. Instead, the end effector was judged to have completed the hammering operation when it reached the target range (in which a nail is assumed to exist) from the initial position. The z-axis velocity of the end effector at the time of its arrival and the motor torque output during the operation were used to evaluate the operation's success and to update the evaluation value.

2) Reinforcement Learning Method: Typical reinforcement learning methods include Q-learning [22], SARSA [23], and Monte Carlo planning [24]. In this experiment, the operation's success is evaluated when the

TABLE III: Reward design.

Case	Reward
Velocity is within the acceptable range	3,000 + Total energy consumed
Velocity is outside the acceptable range	-10 + Total energy consumed
Target range is not reached	$-3{,}000$

hammer reaches the target range. For this reason, we chose the Monte Carlo method, which updates the values in the Q-table all at once after the reward is obtained.

3) States and Actions: The state was defined as the position and velocity of the shoulder pitch axis (Motor 1) and elbow pitch axis (the difference between Motors 1 and 2 multiplied by the reduction ratio of the belt). Each position and velocity range was divided equally into 12 possible values.

The action was defined as any of four possible combinations of outputting or not outputting a constant torque value command to Motors 1 and 2.

4) Reward: We first designed the reward such that the hammer would strike with a constant force. As the experiment focused on the z-axis velocity of the end effector when it reached the target range, we defined the range for the velocity required. When the end effector reached the target range, it was judged to be striking with a certain force if the velocity was within the range. The greater the extent to which these requirements are satisfied, the greater the reward.

We then incorporated the energy consumption of the series of actions into the reward such that the reward increases for lower values of motor output. By taking energy efficiency into account, we expect that energy is to be used only to attain the hammering speed necessary when the end effector reaches the target range. Moreover, we expect the controller to use no unnecessary energy by making contact on the way to achieving the goal.

Thus, a high reward is obtained when the hammering is performed with a constant force and consuming as little energy as possible.

Table III shows the design of the reward. The total energy consumption value is calculated by adding the energy consumption E each time the state changes:

$$E = -(V_1 T_{\mathrm{ref1}} + V_2 T_{\mathrm{ref2}}), \tag{2}$$

where V_1 and V_2 are the velocities and T_{ref1} and T_{ref2} are the torque commands, each for Motors 1 and 2, respectively.

5) Q-table: As we were using the Monte Carlo method, the Q-table was updated when the end effector reached the target range.

The values in the Q-table corresponding to the state and action from the beginning to the end of the operation were updated using the following two formulas:

$$Q(s,a) \leftarrow Q(s,a) + \alpha(G - Q(s,a)), \tag{3}$$

$$G = r_{t+1} + \gamma_{t+2} + \cdots + \gamma^{T-1} r_T. \tag{4}$$

$Q(s,a)$ represents the value of action a in state s.
α denotes the reflection rate of the learning results. We set α to 0.8 when the current reward was larger than the previous reward in the episode, and to 0.05 when the current reward was smaller. Therefore, the behavior is easily reflected when a large reward is obtained.
rn $(n = t+1, \ldots, T)$ represents the reward at each step.
γ is the discount rate, which was set to 0.99.

6) Selection of Action: To select an action, a value for ϵ was first calculated using

$$\epsilon = \frac{0.5}{nk + 1.0}, \tag{5}$$

where n is the number of the episode, and the value of the coefficient k was set to 0.25. Following the ϵ-greedy algorithm, this value of ϵ was then compared with a random value. If the value of $epsilon$ was larger, an action was selected randomly, and if the value of $epsilon$ was smaller, an action was selected from the Q-table.

7) Evaluation Experiment: The learning sequence begins after the end effector is moved to the initial position shown in Figure 5 by position control. An episode ends when the position of the end effector reaches the target range or when the number of steps exceeds 3,000 and the target range has not been reached. The Q-table was updated after each episode was completed. When the velocity remained within the acceptable range for 100 consecutive episodes, the arm returned to the initial position to repeat the learning process, and the learning was judged to be successful.

We used two settings for the range of acceptable velocities in the experiment:

- $[-4.25, -3.75]$ rad/s
- $[-3.75, -3.25]$ rad/s

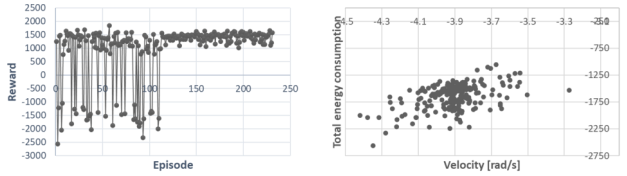
For this evaluation phase, a weight of 500 g was attached to the tip of the end effector.

8) Results: Learning Process: Figure 6a shows the change in reward as the episode count increased when the range of acceptable velocities was set to $[-4.25, -3.75]$ rad/s. In this condition, learning stabilized and motion acquisition was achieved when the episode count exceeded 110. Figure 6b shows the relationship between velocity and energy consumption for this condition. The velocities converged to the slow end of the acceptance range ($[-4.0, -3.75]$ rad/s) as the learning progressed, indicating that the system can learn actions that consume less energy.

When the range of acceptable velocities was set to $[-3.75, -3.25]$ rad/s, more time was needed for the learning to converge (~ 600 episodes),

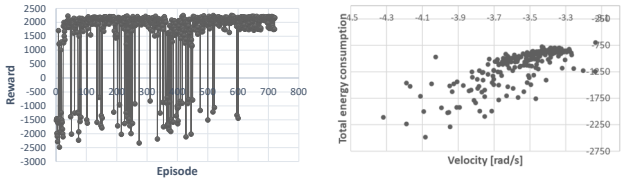
as shown in Figure 6c. However, the controller succeeded in learning the hammering motion with low energy consumption, as shown in Figure 6d.

9) Results: Acquired Behaviors: Figure 7 shows the trajectory of the motion acquired via learning. The horizontal and vertical axes represent the x-coordinate position [m] and z-coordinate position [m], respectively. The outer colored points represent the position of the end effector and the torque of Motor 2, whereas the inner colored points represent



(a) Reward for each passing episode ($[-4.25, -3.75]$ rad/s)

(b) Relationship between velocity and energy ($[-4.25, -3.75]$ rad/s)



(c) Reward for each passing episode ($[-3.75, -3.25]$ rad/s)

(d) Relationship between velocity and energy ($[-3.75, -3.25]$ rad/s)

Fig. 6: Results for reinforcement learning of hammering motion.

the elbow joint axis position and the torque of Motor 1. The figure shows the trajectory from the initial position with the end effector at $(x, z) = (0.1, 0.5)$ until the end effector reached the target position, $(x, z) = (-0.5, 0.05)$.

In the acquired motion, torque was generated for acceleration at the start of the movement. During the downward motion, however, little torque was generated. Just before reaching the target range, only Motor 2 generates torque to adjust the velocity. The results confirmed that the controller can acquire motion using the weight of the arm and the weight attached to the end effector.

More than 10,000 episodes of learning were performed in this experiment, including preliminary trials. No hardware failures occurred that might have halted the experiment. The arm was disassembled and examined to confirm the absence of abnormalities in the individual components. These findings confirmed that the arm could be used in acquiring motions involving physical contact through hardware-based learning alone and that it has the necessary durability to be used in this way. To the best of our knowledge, no robot has learned while undergoing 10,000 collisions; this comparison indicates the superiority of the design of the prototype arm.

In addition, we obtained a stable and successful trajectory for a quick movement using torque control alone and without compensating for friction or deadweight. Our findings confirm that our prototype, which is a low-inertia, high-backdrivability 3-DOF arm, can effectively execute quick movements with torque control.

B. Self-Supervised Learning

We conducted a second experiment, this one involving erasing a whiteboard at the same speed as a human by using self-supervised learning, which is a subcategory of supervised learning algorithms. In this experiment, we wished to accomplish and analyze the following:

- To learn a task that involves a large amount of contact.
- To learn a rapid motion.
- To search for a controller in the torque space.

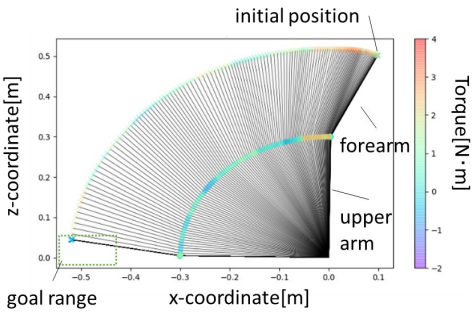


Fig. 7: Trajectory of acquired hammering motion.

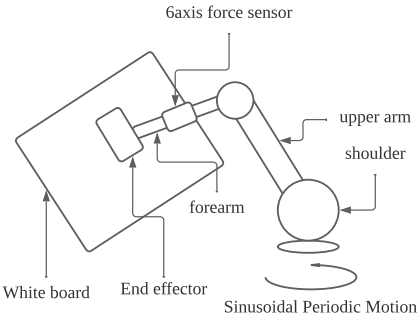


Fig. 8: Setup for experiment with whiteboard erasing using self-supervised-learning dynamics controller.

The task of erasing a whiteboard at the same speed as a human not only involves a large amount of friction between the whiteboard and the eraser, which is difficult to model, but also requires the eraser to apply a constant force to the whiteboard while moving rapidly. Therefore, force control is required rather than simple position control. In order to learn such force control from human demonstrations on actual machines, hardware is needed that will not break during the learning process or when the resulting controller is used. In a test of learning-based control, the hardware is less likely to break if it is able to tolerate forces. Low inertia and high backdrivability are suitable hardware characteristics because they enable the hardware to tolerate the forces directed by a control algorithm. This experiment was designed to confirm that learning in the force dimension is possible using our prototype arm, which has high backdrivability.

1) Setup and Overview of the Experiment: The setup for the experiment is shown in Figure 8. As the end effector, we used a passive 1-DOF joint with an attached eraser. We divided the forearm link into two segments and attached a six-axis force sensor between them.

The sensor was a PFS020YA500U6 [25] by the Leptrino Corporation, from their series of medium- and medium-heavy-load six-axis sensors. The rated capacity of the translational force and moment are 500 N and 0.5 N m, respectively, and the allowable overload is $\pm 200\%$. The yaw axis at the root of the 3-DOF arm was maintained in sinusoidal periodic motion (0.33 Hz), and only the pitch axes of the shoulder and the elbow at the root were controlled to exert the appropriate amount of force on the whiteboard to erase the words written there.

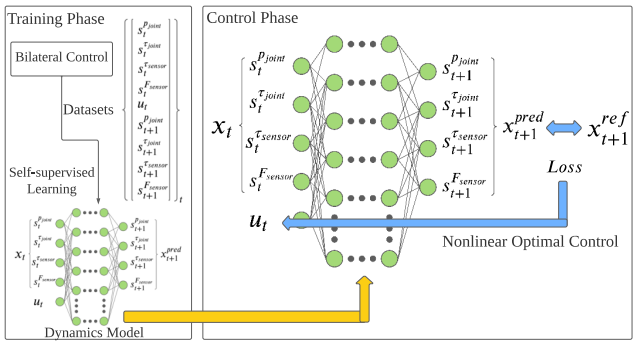


Fig. 9: Overview of the self-supervised-learning dynamics controller.

2) Learning Dynamics via Self-Supervised Learning: A neural network was used to learn the dynamics of the robot and the task, and the control input was determined by nonlinear optimal control [26] [27]. Figure 9 presents an overview of the method used in this study. First, the prototype 3-DOF arm was operated under symmetric bilateral control to collect data as it performed a task. Then, for the task of erasing the whiteboard, we used the collected data to train a dynamics model. This dynamics model includes the model of the prototype 3-DOF arm and is represented by the state equation $x_{t+1} = f(x_t, u_t|W_f)$, where x_t, u_t, and W_f represent the state, control input, and weight of the model network, respectively.

The loss functions of the target and current states are calculated using the learned model, and the control input is determined through optimization using the error back-propagation method. For the current state x_t, the predicted state is x^{pred} and the target state is x^{ref} when the control input u_t is given. The objective function J to make the predicted state approach the target state is expressed as

$$J(x^{\mathrm{ref}}_{t+1}, x^{\mathrm{pred}}_{t+1}) = \frac{1}{2}||x^{\mathrm{ref}}_t - x^{\mathrm{pred}}_t||^2. \tag{6}$$

The calculation of the control input u^{opt}_t by optimization using back-propagation is given by

$$\begin{aligned} u^{\mathrm{opt}}_t &= \arg\min J(x^{\mathrm{ref}}_{t+1}, x^{\mathrm{pred}}_{t+1}) \\ \text{s.t. } u_{\min} &\le u_t \le u_{\max} \\ x^{\mathrm{pred}}_{t+1} &= f(x_t, u_t|W_f) \end{aligned} . \tag{7}$$

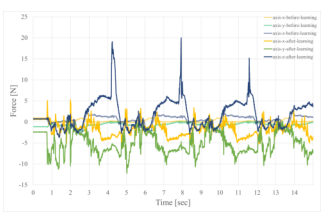
Because the gradient can be approximated by error back-propagation using

$$g_u = \frac{\partial J(x^{\mathrm{ref}}_{t+1}, x^{\mathrm{pred}}_{t+1})}{\partial u_t}, \tag{8}$$

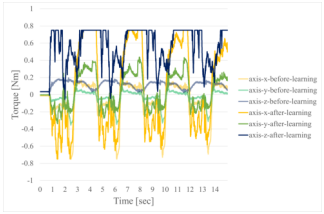
the control input is updated using

$$\begin{aligned} \delta u_t &= -\epsilon_u \frac{g_u}{||g_u||}, \\ u_t &\leftarrow u_t + \delta u_t \end{aligned} \tag{9}$$

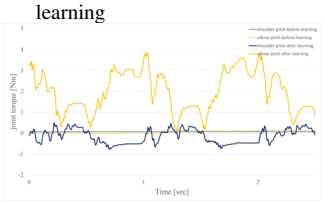
where g_u is the gradient of J for u_t, and ϵ is the constant learning rate for updating u_t.



(a) Sensor force before and after learning



(b) Sensor torque before and after learning



(c) Joint torque before and after learning

Fig. 10: Results of experiment with whiteboard erasing using self-supervised-learning dynamics controller.

3) Design of Controller: By operating the prototype 3-DOF arm with bilateral control, we found that letters written on the whiteboard could be erased when the force sensor received a force in the vertical direction toward the whiteboard. We also confirmed that there was reaction torque on the six-axis force sensor in the direction of the reaction force from the whiteboard. In addition, the third axis of the motor exerted a torque in the direction that pushed the eraser against the whiteboard during bilateral control.

We included the joint angle $s^{p_{\mathrm{joint}}}_t$, joint-exerted torque $s^{\tau_{\mathrm{joint}}}_t$, sensor torque $s^{\tau_{\mathrm{sensor}}}_t$, and sensor force $s^{F_{\mathrm{sensor}}}_t$ in x_t, and the target joint angle $u^{p_{\mathrm{joint}}}_t$ and target joint torque $u^{\tau_{\mathrm{joint}}}_t$ in u_t to move the eraser while exerting force against the whiteboard. The dimensionality of $s^{p_{\mathrm{joint}}}_t$, $s^{\tau_{\mathrm{joint}}}_t$, and $s^{\tau_{\mathrm{sensor}}}_t$ is 3, that of $s^{F_{\mathrm{sensor}}}_t$ is 1 (the reaction direction alone), and that of $u^{p_{\mathrm{joint}}}_t$ and $u^{\tau_{\mathrm{joint}}}_t$ is 2 (shoulder pitch and elbow pitch joints). We set x_{ref} to $(0.0, 0.0, 0.0, 0.0, -0.2, 1.0, 0.0, 0.0, 0.5, -4.0)$; this value was chosen by referring to the sensor value when the prototype robot wiped the whiteboard by bilateral control. For the dynamics model, we used LSTM [28], with 128 as the number of units. The control input was sent to the motors with a period of 3 ms. The calculation of optimal control input using back-propagation took 30 ms, and therefore we updated the target command with a period of 30 ms.

4) Results of Evaluation Experiment Using Learned Controller: The value of the force sensor received from the whiteboard was compared with that in the command to maintain the posture when the first axis was moved. It was confirmed to increase by a factor of approximately five when the whiteboard was erased using the learned model (Figure 10).

The controller could be examined without causing breakage by experimenting with the control input resulting from the search for the torque as the command value in the learning-based controller.

In this phase of the experiment, the yaw axis of the root was controlled while continuing to move at the same speed

as a human. It was confirmed that it is possible to acquire control of the force when the whiteboard was wiped at the same speed as a human.

VI. Conclusions and Future Work

In this study, we determined the backdrivability requirements for two selected tasks, selected actuators based on the requirements, fabricated a prototype of a 3-DOF arm, and achieved a task involving quick movements and physical contact using machine-learning-based control to perform human-equivalent tasks in a human–robot-coexistence environment.

We demonstrated an overall process for designing a prototype arm that allows human–robot coexistence, in which the hardware design requirements are derived from the task and from the hardware used when the task is performed in a more flexible (less limited) manner.

In future studies, it will be useful to generalize this design process to accommodate general-purpose environments and less restricted tasks. With regard to hardware, it will be necessary to derive hardware design requirements from generalized tasks and to include factors in addition to the static friction in order to develop arms having multiple degrees of freedom, low inertia, and high backdrivability. The development of actuators that can provide both the backdrivability and the torque required for the payload is another important aspect for future consideration.

References

[1] S. Robla-Gómez, V. M. Becerra, J. R. Llata, E. González-Sarabia, C. Torre-Ferrero, and J. Pér ez Oria, "Working together: A review on safe human-robot collaboration in industrial environments," *IEEE Access*, vol. 5, pp. 26 754–26 773, 2017.

[2] S. Levine, N. Wagener, and P. Abbeel, "Learning contact-rich manipulation skills with guided policy search," *CoRR*, vol. abs/1501.05611, 2015. [Online]. Available: http://arxiv.org/abs/1501.05611

[3] T. M. Corporation, "Toyota unveils third generation humanoid robot t-hr3," https://global.toyota/en/newsroom/corporate/19841525.html, accessed on:Feb.15,2022[Online].

[4] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppe, A. Albu-Schaeffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald, and G. Hirzinger, "The kuka-dlr lightweight robot arm - a new reference platform for robotics research and manufacturing," in *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*, 2010, pp. 1–8.

[5] S. Xu, M. Yokoyama, and T. Shimono, "Back-drivability improvement of geared system based on disturbance observer and load-side disturbance observer," *IEEJ Journal of Industry Applications*, vol. 9, no. 5, pp. 475–485, 2020.

[6] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. Hoepflinger, "Anymal - a highly mobile and dynamic quadrupedal robot," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 38–44.

[7] R. Robotics, "Sawyer black edition," https://www.rethinkrobotics.com/sawyer, accessed on:Feb.15,2022[Online].

[8] H. Song, Y.-S. Kim, J. Yoon, S.-H. Yun, J. Seo, and Y.-J. Kim, "Development of low-inertia high-stiffness manipulator lims2 for high-speed manipulation of foldable objects," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4145–4151.

[9] D. V. Gealy, S. McKinley, B. Yi, P. Wu, P. R. Downey, G. Balke, A. Zhao, M. Guo, R. Thomasson, A. Sinclair, P. Cuellar, Z. McCarthy, and P. Abbeel, "Quasi-direct drive for low-cost compliant robotic manipulation," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 437–443.

[10] P. M. Wensing, A. Wang, S. Seok, D. Otten, J. Lang, and S. Kim, "Proprioceptive actuator design in the mit cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 509–522, 2017.

[11] K. Kojima, Y. Kojio, T. Ishikawa, F. Sugai, Y. Kakiuchi, K. Okada, and M. Inaba, "A robot design method for weight saving aimed at dynamic motions: Design of humanoid jaxon3-p and realization of jump motions," in *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, 2019, pp. 586–593.

[12] H. Matsuki, K. Nagano, and Y. Fujimoto, "Bilateral drive gear —a highly backdrivable reduction gearbox for robotic actuators," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 6, pp. 2661–2673, 2019.

[13] FESTO, "Bionic learning network," https://www.festo.com/us/en/e/about-festo/research-and-development/bionic-learning-network-id_31842, accessed on:Feb.15,2022[Online].

[14] T. Lens and O. von Stryk, "Investigation of safety in human-robot-interaction for a series elastic, tendon-driven robot arm," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 4309–4314.

[15] G. Gorjup, C.-M. Chang, G. Gao, L. Gerez, A. Dwivedi, R. Yu, P. Jarvis, and M. Liarokapis, "The aroa platform: An autonomous robotic assistant with a reconfigurable torso system and dexterous manipulation capabilities," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 4103–4110.

[16] A. Dietrich, *Whole-Body Impedance Control of Wheeled Humanoid Robots*. Springer International Publishing Switzerlaand, 2016.

[17] J. Urata, Y. Nakanishi, K. Okada, and M. Inaba, "Design of high torque and high speed leg module for high power humanoid," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 4497–4502.

[18] K. Kawaharazuka, S. Makino, K. Tsuzuki, M. Onitsuka, Y. Nagamatsu, K. Shinjo, T. Makabe, Y. Asano, K. Okada, K. Kawasaki, and M. Inaba, "Component modularized design of musculoskeletal humanoid platform musashi to investigate learning control systems," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 7300–7307.

[19] S. Buechner, V. Schreiber, A. Amthor, C. Ament, and M. Eichhorn, "Nonlinear modeling and identification of a dc-motor with friction and cogging," in *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*, 2013, pp. 3621–3627.

[20] R. Szczepanski, T. Tarczewski, L. J. Niewiara, and D. Stojic, "Identification of mechanical parameters in servo-drive system," in *2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC)*. IEEE, 2021, pp. 566–573.

[21] S. Bembli, N. K. Haddad, and S. Belghith, "Adaptive sliding mode control with gravity compensation: Application to an upper-limb exoskeleton system," in *MATEC Web of Conferences*, vol. 261. EDP Sciences, 2019, p. 06001.

[22] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3, pp. 279–292, 1992.

[23] G. A. Rummery and M. Niranjan, *On-line Q-learning using connectionist systems*. Citeseer, 1994, vol. 37.

[24] L. Kocsis and C. Szepesvári, "Bandit based monte-carlo planning," in *European conference on machine learning*. Springer, 2006, pp. 282–293.

[25] I. Leptrino, "Leptrino force sensor pfs series," https://www.leptrino.co.jp/product/6axis-force-sensor, accessed on:Feb.15,2022[Online].

[26] K. Kawaharazuka, K. Tsuzuki, M. Onitsuka, Y. Asano, K. Okada, K. Kawasaki, and M. Inaba, "Object recognition, dynamic contact simulation, detection, and control of the flexible musculoskeletal hand using a recurrent neural network with parametric bias," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4580–4587, 2020.

[27] T. Murooka, K. Okada, and M. Inaba, "Diabolo orientation stabilization by learning predictive model for unstable unknown-dynamics juggling manipulation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 9174–9181.

[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, p. 1735–1780, nov 1997.