Master's Thesis of Engineering

# Developing Motion Control of Different Morphology using Deep Reinforcement Learning

## 심층 강화학습을 이용한 사람의 모션을 통한 이형적 캐릭터 제어기 개발

August 2022

Seoul National University
Department of Computer Science and Engineering
Sunwoo Kim

# Developing Motion Control of Different Morphology using Deep Reinforcement Learning

심층 강화학습을 이용한 사람의 모션을 통한 이형적 캐릭터 제어기 개발

지도교수 서 진 욱

이 논문을 공학석사 학위논문으로 제출함

2022년 4월


서울대학교 대학원

컴퓨터공학부

김 선 우


김선우의 공학석사 학위 논문을 인준함

2022년 6월

| | | |
|---|---|---|
| 위 원 장: | 김 명 수 | (인) |
| 부위원장: | 서 진 욱 | (인) |
| 위 원: | 신 영 길 | (인) |

# Abstract

A human motion-based interface fuses operator intuitions with the motor capabilities of robots, enabling adaptable robot operations in dangerous environments. However, the challenge of designing a motion interface for non-humanoid robots, such as quadrupeds or hexapods, is emerged from the different morphology and dynamics of a human controller, leading to an ambiguity of control strategy. We propose a novel control framework that allows human operators to execute various motor skills on a quadrupedal robot by their motion. Our system first retargets the captured human motion into the corresponding robot motion with the operator's intended semantics. The supervised learning and post-processing techniques allow this retargeting skill which is ambiguity-free and suitable for control policy training. To enable a robot to track a given retargeted motion, we then obtain the control policy from reinforcement learning that imitates the given reference motion with designed curriculums. We additionally enhance the system's performance by introducing a set of experts. Finally, we randomize the domain parameters to adapt the physically simulated motor skills to real-world tasks. We demonstrate that a human operator can perform various motor tasks using our system including standing, tilting, manipulating, sitting, walking, and steering on both physically simulated and real quadruped robots. We also analyze the performance of each system component ablation study.

**keywords**: Reinforcement Learning, Computer Graphics, Robotics, Human Robot Interaction

**student number**: 2019-25917

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

A legged robot worker that is secure for entering hazardous environments has long been a pursuit in the robotics field. To achieve this goal, many research teams have developed various approaches to designing autonomous robotic systems from classical model-based control to learning-based controllers. However, current autonomous agents struggle with unexpected dangerous scenarios where the workers lack the information as disasters. To overcome the limitation, the demand for a more flexible control system that is applicable to black-box scenarios has emerged in a few years.

We present a human motion control system that allows operators to flexibly control quadrupedal robots using reflexive motions. In the conventional approach, the human motion control system has been designed with a model-based algorithm. As shown in the work of Ramos and Kim [5], the traditional interface projects human centroidal dynamics to the robot's space. In the contrast, our approach carries great potential to overcome completely novel scenarios by obtaining the novel idea using motion imitation learning [6, 7, 8]. This approach has proved to perform natural motion control on simulated creatures or real robots. Especially, we target the interface to the quadrupedal robots inspired by emerging success of demonstrations [9, 7, 10, 11].

We conduct our motion control systems into two main parts: the motion retargeting module and the motion imitation control policy. The motion retargeting module cap-

tures a live human motion and decodes it into the dynamically plausible robot motion that carries the intention of the input human motion. The imitation policy tracks the retargeted robot motion based on the detected sensor data. We design the motion retargeting module similar to the supervised learning style while designing an imitation control policy with deep reinforcement learning.

There are a few notable challenges to attaining our system and developing a general motion control interface. The first challenge generates retargeting and control difficulties caused by the intention ambiguity from human motion. We mitigate this issue by embracing a hierarchical model with a set of experts for motion retargeting networks and control policies. The second key challenge is the retargeted motion may show the implausible traits. We adjust the motion with a couple of post-processing methods by guaranteeing contact and temporal consistencies of the retargeted motion. Another key challenge is the lack of accessing the future reference trajectory when imitating the target motion. We adopt curriculum learning in the training step which gradually increases the difficulty of multiple tasks.

We demonstrate the execution of various motor tasks seamlessly on both simulated and real quadrupedal robots by our system. We use a consumer-grade motion capture system, Microsoft Kinect [12] to perform such a control interface. We show the individual tasks of the operator controlling an A1 robot to approach the target and manipulate the object with both standing and sitting postures. Also, the operator can tilt the robot's body to reach out to a distant object or avoid incoming objects that possibly damage the robot. Furthermore, we composite those tasks to achieve multiple goals in a novel scenario. We analyze our system by ablation studies of essential components such as consistency corrections, curriculum learning, and domain randomization. Our technical contributions are as following list:

- We develop a novel human motion interface system for a quadrupedal robot that requires minimal information about the task or the model.

- We create an adequate motion retargeting algorithm with contact and temporal consistency corrections.
- We enhance the performance of motion imitation by adopting curriculum learning and a hierarchical formulation of a set of experts.
- We demonstrate that an operator can execute various motor tasks seamlessly on simulated and real robots.

Figure 1.1: Novel control system that allows an operator to control a quadrupedal robot on various tasks.

# Chapter 2

# Related Work

## 2.1 Legged Robot Control

**Legged robot control.** For decades, roboticists have strived for the advance of robust and dynamic robotic systems in both hardware and software of the legged robots. Under these advancements, legged robots can demonstrate diverse, robust, and dynamic motor skills. The legged robots now can traverse challenging terrains or exhibit high agility in their motion. The robot hardware development has allowed agile motor skills along with high stability of quadrupedal robots [13, 14, 15]. Besides, the research on bipedal robot hardware has developed the robustness of locomotion [16, 17, 18]. Conventional software techniques for effective motion controllers have involved numerous manual engineering and domain expertise. The mathematical approaches such as trajectory optimization [19, 20] and model predictive control (MPC) [21, 22, 23, 24], bring the optimization techniques to forging robot motions while relieving the human-powered struggles in controller design process. The development of optimization methods has enabled legged robots to conduct the challenging control goals such as locomoting on a slippery floor [25, 26], traversing the rough terrain [27], recovering from the slip [28] and even keeping the balance on a large ball in a physics simulation [29]. However, the real-legged robot obtains high complexity

dynamics formulation which usually leads the algorithms to either operate with a simplified robot model or design a task-specific controller. Our system allows the robot to learn a wide range of motor skills without any task-specific dynamics modeling.

**Learning-based control.**The control of physically simulated characters using Reinforcement learning (RL) has shown a great performance in sophisticated motor skills such as walking, jumping, cart-wheel, and skating [6, 30, 31, 32]. However, control strategies learned in idealized simulation environments often struggle when transferred to the real world. When this control policy is applied to the real-world environment, the motor controller exhibits infeasible behaviors due to the simulation and real-world difference, which is often referred to as the *sim-to-real gap* or shortly *reality gap*. To address the reality gap, some research groups attempt to solve it by combining the policy with conventional optimization methods such as MPC, allowing it to adjust on the real-robot [33, 34, 35, 36]. Another groups have investigated approaches that leverage real-world data, such as learning on real robots [37, 38, 39], identifying system parameters [9], or adapting policy behaviors [7, 40, 11]. Instead, we employ a Domain Randomization (DR) technique [41, 42, 43, 44, 45, 46, 8], which randomizes domain parameters such as mass, friction, or PD gain during training in simulation to obtain more robust control policies while training only in simulation.

## 2.2 Motion Imitation

Data-driven motion controllers have shown effective motion generation for a wide range of physically plausible motions by leveraging motion capture data. This methods have been developed to obtain interactive motion control [47, 48, 49, 50, 51, 52]. However, the kinematic approach cannot be transferred to real-world directly, since the motions lack dynamics information. On the other hand, physics-based motion controllers [53, 54] let us convey physically feasible motions in simulation, but their control design requires extra manual efforts, such as feature selection and motion pro-

cessing. The recent RL-based formulation [6] provides an automated pipeline for generating motion imitation control policies from simple reward descriptions containing imitation components. This scheme shows novel capability of learning various motions on simulated characters [55, 56, 57, 6, 58, 30, 59, 31, 32, 60, 61], or even on a real quadrupedal robot [7]. We adopt the concept of reinforcement learning with an imitation objective to gain both physically correct motion and interactive control.

## 2.3 Motion-based Control

Controlling robots using human motion provides an intuitive interface by giving the command directly from the human body motion to the robot body. Human motion control schemes liberate the human operator from commonly used control mechanisms such as joysticks, keyboards, or mouses. Furthermore, they allow the operator to convey the user's intention to the robot control much better than conventional control interfaces. Because of these benefits the control schemes obtain, human posture-based control has been widely studied by computer animation researchers and roboticists. [62, 63, 64, 3, 65, 4, 66, 67, 68, 69, 70, 5, 1, 71, 72].

**Human motion control for humanoid robots.** Humanoid robots inherit many features of human morphology therefore, they provide a suitable platform for mimicking human motions. Application of such anthropomorphic creatures ranges from human interaction [67] to housekeeping teleoperation [73] or even hazardous disaster rescue [68]. The expansion of application to small-scaled humanoid by motion imitation [70] emphasize the challenges of casting the human posture to a new morphology. Research groups [62, 63, 65] have offered the methods dealing with the morphology differences in the configuration space as joint limits, link length, and degrees of freedom. Timing is an additional factor to consider when expanding the controller from simple posture mapping to more dynamic motions like maintaining a balance. Zheng and Yamane [66] suggest the method of integrating a time-warping objective

to obtain a smoother motion-to-motion mapping control. Specifically, many research groups have dedicated solving the balancing issues by many means such as Linear Inverted Pendulum (LIP) model safety constraints [69] or the balance feedback to the human [5]. Aside from these issues, teleoperation schemes have to overcome the safety challenge to protect the manipulating robots. Choi et al. [72] proposed a shared-latent embedding retargeting algorithm to avoid self-collisions which can potentially cause severe damage. Arduengo et al. [74] demonstrated a technique of the end-effector to switch between stiffness and compliance to convey sounder safety.

**Motion retargeting to non-humanoid characters.** The non-human-like characters have different configurations from humans, therefore, conveying challenges of controlling using human motion. However, several groups have succeeded to overcome this issue, obtaining the semantics of the human postures to characters as animals or alphabet shape creatures [64, 3, 4] with proper motion retargeting skills. The pose-to-pose motion retargeting algorithms have been studied by various approaches as probabilistic pose-to-pose mapping [3], semantic deformation transfer as mapping [64], or a feature selecting method [4]. These mappings show highly interactive posture mapping but are not being extended for robot control in the physics world. In 2D physics simulation, Kim et al. [1] presented the embedding of cyclic motion on the shared latent space, which can control the ostrich character. In this work, we propose a new control framework that allows a user to control a quadrupedal robot with body motions, which allows a different morphology control. We achieve real-time motion control for various tasks, including walking, tilting, manipulation, and sitting, with a minimal amount of information about each task.

# Chapter 3

## Overview

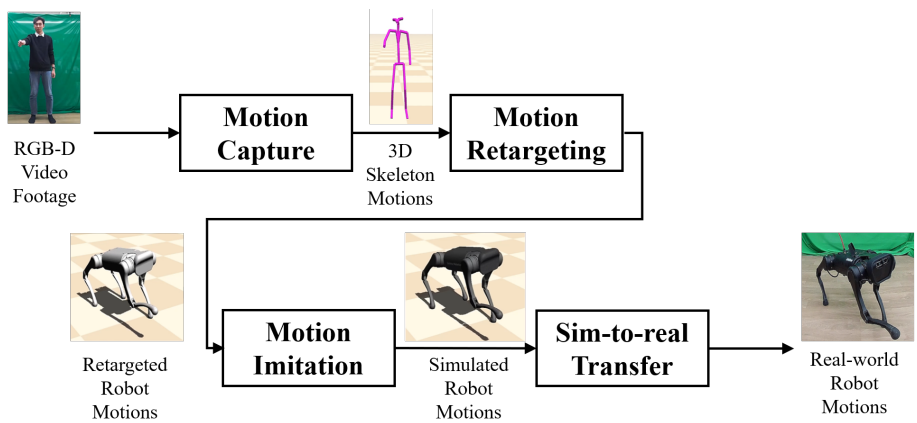



Figure 3.1: Overview diagram of our system. It takes a human motion as inputs and controls the robot via motion retargeting and motion imitation.

Our system demonstrates the control framework for controlling a quadrupedal robot with a human operator's motion. Our framework receives human motion data as input from any motion capture methods. We use Microsoft Azure Kinect [12] in our case which serves simple access to the captured data. Then, the motion retargeting module in our system (Chapter 4) transforms the captured human motion into the corresponding robot motion that is physically plausible and obtains proper semantics. We

adopt a hierarchical structure of learning a set of experts to build mappers while applying post-processing techniques to correct the motion. To generate a control policy that can be adaptable to the physics environment, we learn a control policy that imitates the given retargeted robot motion using deep reinforcement learning (Chapter 5). For more robust and flexible control, we develop expert policies using curriculum learning and combine them as a state machine in operating time with additional transition algorithms. We illustrate the system overview in Figure 3.1.

# Chapter 4

# Motion Retargeting Module

A motion retargeting module converts the captured operator's motion into the corresponding robot motion. Although numerous prior works have demonstrated success in human-to-humanoid motion mapping [62, 63, 65, 66, 67, 70, 68, 69, 5, 73, 72], our goal carries the unique problem to find a mapping function between two very different morphologies. This problem usually obtains complex calculations, leveraging man-crafted engineerings such as contact physics or centroidal dynamics. Even worse, we have to handle additional issues, such as the sparsity of the data and the necessity of interactivity.

We dive into this issue by leveraging the concept of the deep neural network to create a motion retargeting function. Thus, we propose the idea of learning a set of expert networks and applying post-processing to consist of the retargeting module. Traditional techniques [2, 64, 3] commonly retarget the motion by solving optimization. However, they exhibit a slow turnaround time that is unsuitable for interactive applications. Furthermore, they often require task-specific formulation [4], which affect the system to be complicated when handling a wide variety of motions. On the other hand, learning-based approaches [72] demonstrate impressive inference capabilities at interactive rates, but we might suffer the data-hungry problem. Furthermore, they can generate inconsistent or unexpected motions that cause hardware damage to

the robot. We propose a motion retargeting module that first infers robot motion in a supervised learning fashion from a sparse dataset. Then, it corrects the inconsistency of motion using simple optimization by post-processing. We generate a set of experts to manage the various motion as an additional selector. Therefore, our system enables the user to build a fast and robust motion retargeting scheme that is applicable to a wide range of tasks.

## 4.1 Motion Retargeting Network

In this section, we will illustrate how to learn a motion retargeting network for a single task. We develop a motion mapping function $f$ that takes a human pose $\mathbf{q}$ as inputs and then outputs the mapped corresponding robot pose $\mathbf{p}$. A simple pose-to-pose mapping module may produce ambiguous motions when retargeting periodic motions because a single pose does not contain any temporal information. For example, when expressing an in-place marching human motion, the two different phases *swing up* or *swing down* are interpreted as the same pose which must be retargeted to different quadruped poses. Therefore, we construct the retargeting network to take a triplet of the human pose, velocity, and acceleration $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ to retarget those of robots $(\mathbf{p}, \dot{\mathbf{p}}, \ddot{\mathbf{p}})$. We omit the derivatives in some figures and equations for brevity.

**Data preparation.** To prepare a dataset $\mathcal{D}$ for learning the single mapping network, we collect matching pairs of human and robot motion. First, we generate robot motions for sampled tasks. For instance, we generate tilting task motions by giving a random task goal and solving inverse kinematics to obtain robot motion. For locomotion tasks, we generate a set of walking motions with various gait parameters such as body heights, foot clearance heights, and swing angles using a trajectory generator [33]. Note that these motion data are reused for training imitation policy in Section 5.

After generating robot motion datasets, we collect the matching human motion sequences. We ask a human operator to act the "corresponding motions" while showing

the robot motions based on the operator's intuition. We capture the human motion of this act. Then, we manually process the motions to clean up noisy features and fix asynchronous actions. For both robot and human motion, we pair the triplets of the pose and their derivatives with finite differences with $\Delta t = 0.1$.

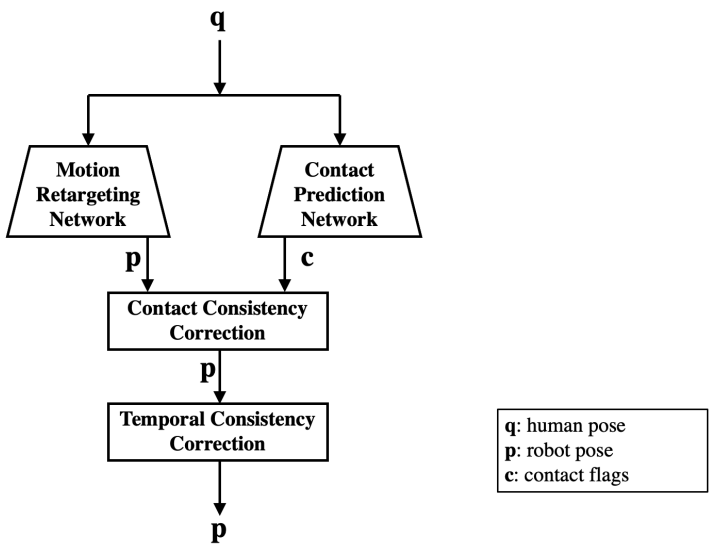



Figure 4.1: The illustration of the motion retargeting module. The motion retargeting networks converts the given human motion $\mathbf{q}$ into the robot motion $\mathbf{p}$. The contact and temporal consistencies are corrected based on the inferred contact flags and previous retargeted motion.

**Learning process.** We use a multi-layer perceptron (MLP) as a retargeting network from the given dataset $\mathcal{D}$ (Figure 4.1). Our MLP consists of three leaky ReLU layers and one final hyperbolic tangent layer to make the output value between [-1, 1]. We then shift and scale the outputs using the robot joint limit vector to finalize the joint angle of the robot. We define our loss function as follows:

$$L_{map} = w_{ori}L_{ori} + w_{jnt}L_{jnt} + w_{dx}L_{dx} + w_{ddx}L_{ddx}. \tag{4.1}$$

We bypass the function arguments $\mathbf{p}$, $\dot{\mathbf{p}}$, $\ddot{\mathbf{p}}$, $\bar{\mathbf{p}}$, $\bar{\dot{\mathbf{p}}}$, and $\bar{\ddot{\mathbf{p}}}$ for brevity, where the former

three are the outputs from the networks and the latter three are the target values from the datasets. The orientation loss $L_{ori} = d(\mathbf{p}^{root}, \bar{\mathbf{p}}^{root})$ is designed to compare the root orietation $\mathbf{p}^{root}$ in quaternion and its target value $\bar{\mathbf{p}}^{root}$ via a quaternion distance function $d$. The joint angle loss $L_{jnt} = \|\mathbf{p}^{jnt} - \bar{\mathbf{p}}^{jnt}\|^2$ matches the joint angles $\mathbf{p}^{jnt}$ and their target joint angles $\bar{\mathbf{p}}^{jnt}$. These two loss term mainly drives the loss function. The rest two end-effector terms are auxiliary that drives the end-effectors to settle for target positions which are expressed as $L_{dx} = \|\dot{\mathbf{x}} - \bar{\dot{\mathbf{x}}}\|^2$ and $L_{ddx} = \|\ddot{\mathbf{x}} - \bar{\ddot{\mathbf{x}}}\|^2$. They compare the end effector velocities $\dot{\mathbf{x}}$ and accelerations $\ddot{\mathbf{x}}$ against their target values, $\bar{\dot{\mathbf{x}}}$ and $\bar{\ddot{\mathbf{x}}}$, respectively. Note that these these values can be derived from $\dot{\mathbf{p}}$ and $\ddot{\mathbf{p}}$. As mentioned above, we set the weights $w_{ori}$, $w_{jnt}$, $w_{dx}$, and $w_{ddx}$ as $0.3$, $1$, $0.001$, and $0.001$ for all the experiments, respectively to reflect our intentions to each term.

## 4.2 Post-processing for Consistency

The problem with the learned retargeting function is that they often yield physically inconsistent motions in practice. This inconsistency slows the learning of a control policy and degrades the final motion control quality. To overcome this issue, we clean up the motion at the post-processing stage to maintain contact and temporal consistency.

**Contact consistency correction.** Motion without contact consistency shows the foot skating that is crucial to obtaining physical plausibility. The contact consistency violation destabilizes the robot balancing, hence leading to learning failure. To solve this issue, we estimate four-dimensional contact flag vector $\mathbf{c}_t$. Then undesirable motions are fixed when the flags are supposed to be in contact phases.

One simple possible approach to estimate $\mathbf{c}_t$ is to compare the robot's foot heights to a certain threshold height. However, we found that this approach yields undesirable discontinuous motions when manipulating a foot near the ground level. Therefore, we learn an auxiliary network called the"contact prediction network" that predicts smooth contact probabilities directly from human motion input. We train this contact

consistent network from the same training data using the following loss function:

$$L_{cp} = \|\bar{\mathbf{c}}_t(\bar{\mathbf{q}}, \bar{\dot{\mathbf{q}}}) - \mathbf{c}_t(\mathbf{q}, \dot{\mathbf{q}})\|^2, \tag{4.2}$$

here, the contact probability $\bar{\mathbf{c}}_t$ is continuously estimated from the foot height and velocity. We define the probability as 1.0 if the height is 0cm and the velocity is $0.0$cm/s, while 0.0 if the height is above 2cm and the velocity is more than $60.0$cm/s. This function both considers the motion and the height as a probability so that the flag reflects the movement features better. We apply inverse kinematics when the contact probability is greater than 0.5 to correct the contact feet to the previous frame's positions.

**Temporal consistency correction.** We also design a procedure to guarantee the temporal consistency of the retargeted motions over continuous frames not to be in a dangerous motion caused by abrupt movements. Since deploying the proposed system in the real world is critical in our research, we need to stable these sudden moves. To this end, we clip the joint angles concerning the manually set velocity limits, which are set to $120^\circ/s$.

## 4.3 A Set of Experts for Multi-task Support

In our experiments, obtaining an accurate motion mapping when the human operator tries to demonstrate multiple tasks which are close to each other. To manage this issue, we propose building a hierarchy of the network learning that manages a set of expert networks [75, 76, 77, 78]. We primary learn each different motion retargeting networks for three robot states, *stand*, *walk*, and *sit* (Figure 4.2). Each network can handle multiple tasks within the state. For instance, *manipulation-at-stand*, *tilting-at-stand* or doing both on *stand* state.

We query k-Nearest neighbors (kNN) over the human input motion to identify the state expert associated with the closest data set. We switch the corresponding expert only when the transition signal motion is detected. We discovered that this results

in an accurate mapping function while reducing the time and resources for manual engineerings, such as hyperparameter tuning and data curation.

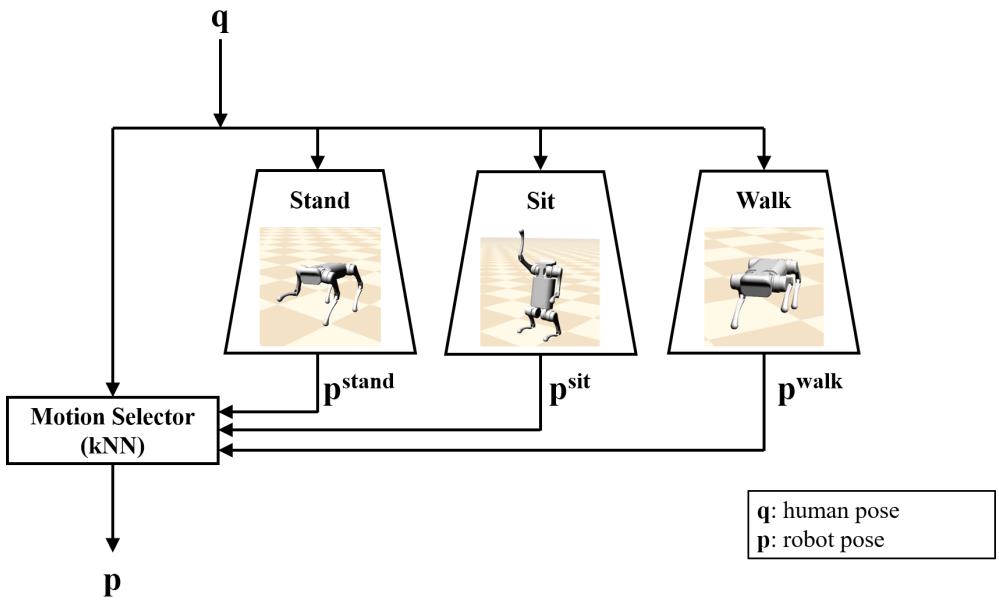



Figure 4.2: We learn a set of expert motion retargeting networks for better controllability in multi-task scenarios.

# Chapter 5

# Motion Imitation Module

The second step of our system is to develop a control policy to imitate the given reference motion generated from the first module. We employ the reinforcement learning with the imitation reward framework of Peng et al. [6] that enables natural and diverse motions in physics simulation, which has also been applied to a real quadrupedal robot [7].

Our ambitious goal aims to track a wide range of motions on real robots so that it differs from other quadrupedal research. Furthermore, to achieve our goal, we face additional unique challenges, especially in the learning process. The first issue to address is dealing with noisier references in imitation learning. Since we capture the input from live human movements, the retargeted reference motion involves noises that are not observed in other works. Thus, our controller must obtain the capability to imitate a wide range of motion with spatial and temporal noises because a human cannot reproduce the exact same motion. The second difficulty comes that our controller can't access the "future" reference motions. Clearly, the absence of future information frequently set a control policy to be conservative states rather than actively tracking the reference motion.

To overcome the problems and maximize the performance of motion imitation, we present the following techniques. First, our system provides a hierarchical model of

control that learns three expert controllers for robot states *stand*, *sit*, and *walk*. The transition between states is manually designed. Second, to obtain an effective expert controller, we introduce curriculum learning which is organized over difficulties and tasks. Our novel ideas demonstrate a practical controller that is capable of handling a wide range of motion data on simulation and real robots.

## 5.1 Background: Reinforcement Learning

Our problem is formulated as Partially Observable Markov Decision Processes (PoMDP) to employ the reinforcement learning [79]. On an each time step, an agent observes an observation $\mathbf{o}_t \sim \mathcal{O}(\mathbf{s}_t)$ emitted from the current state $\mathbf{s}_t$ and takes an action $\mathbf{a}_t \sim \pi(\mathbf{a}_t|\mathbf{o}_t)$ from its policy $\pi$. This results in the trajectory of the states and actions $\tau = \{(\mathbf{s}_0, \mathbf{a}_0), (\mathbf{s}_1, \mathbf{a}_1), \cdots (\mathbf{s}_T, \mathbf{a}_T)\}$ where $T$ is the episode length. Our goal is to find the optimal policy that maximizes the expected return:

$$J(\pi) = E_{\tau \sim p(\tau|\pi)}[\sum_{t=0}^{T-1} \gamma^t r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1})], \tag{5.1}$$

where $p(\tau|\pi)$ is a probability of the given trajectory $\tau$.

## 5.2 Formulation of Motion Imitation

We formulate the problem of imitating the given reference motion as PoMDP.

**Reference Motions.** To provide the reference motions for imitation learning, we take the robot trajectories that are generated for training a retargeting function in the previous section. We also injected a noise vector into reference motions to improve the robustness of the control policy.

**Observation.** We set the observation vector $\mathbf{o}_t = [\mathbf{z}_{t-3:t}, \mathbf{a}_{t-3:t-1}, \bar{\mathbf{p}}_{t-3:t}]$ consisting of three components: robot sensor data, previous actions, and reference poses, with their corresponding histories. Each robot sensor data $\mathbf{z}_t$ represents a 16 dimensional

vector from 12 joint motor encoders and 4 IMU orientation readings in quaternions. A history vector of previous actions $\mathbf{a}_{t-3:t-1}$ is also stored to make the problem more Markovian in the real world. The policy also takes the previous reference poses $\bar{\mathbf{p}}_{t-3:t}$. Please note that we do not have *future* reference motions due to the nature of our problem therefore, it is more difficult to solve tracking tasks.

**Action.** The action $\mathbf{a}_t$ defines as the PD target for the twelve joint motors of a robot. We apply the Butterworth low-pass filter with the cut-off frequency at 5Hz to actions to generate smoother motions.

**Reward function.** We design reward functions for the learning to encourage the agent to imitate the given reference motion while adapting to the physics environment of simulation:

$$r_t = w^{main} r_t^p \cdot r_t^e \cdot r_t^{rp} \cdot r_t^{ro} \cdot r_t^{sp} + w^{acc} r_t^{acc}, \tag{5.2}$$

which embraces the multiplicative form to drive all the component to obtain certain level of reward by previous works [30, 31]. The term $r_t^p$ refers to a joint imitation reward:

$$r_t^p = \exp\left(s_p \sum_j \|\bar{\mathbf{p}}_t^j - \mathbf{p}_t^j\|^2\right), \tag{5.3}$$

where $\bar{\mathbf{p}}$ and $\mathbf{p}$ are the target and sensor-read joint angles. The end-effector reward $r_t^e$ induces the robot to track the end-effector of the reference:

$$r_t^e = \exp\left(s_e \sum_e \|\bar{\mathbf{x}}_t^e - \mathbf{x}_t^e\|^2\right), \tag{5.4}$$

where $\bar{\mathbf{x}}_t^e$ and $\mathbf{x}_t^e$ denotes the target and current end-effector positions in 3D cartesian coordinate relative to the root position. In the same way, the root position reward $r_t^{rp}$ and the root orientation reward $r_t^{ro}$ drives the robot to minimize the differences in root

position and orientation:

$$
\begin{aligned}
r_t^{rp} &= \exp(s_{rp}\|\bar{\mathbf{x}}_t^{\text{root}} - \mathbf{x}_t^{\text{root}}\|^2) \\
r_t^{ro} &= \exp(s_{ro} d(\bar{\mathbf{p}}_t^{\text{root}}, \mathbf{p}_t^{\text{root}})^2)
\end{aligned}
\tag{5.5}
$$

by comparing the current root position $\mathbf{x}^{\text{root}}$ and orientation and $\mathbf{p}^{\text{root}}$ with respect to their target values, $\bar{\mathbf{x}}_t^{\text{root}}$ and $\bar{\mathbf{p}}_t^{\text{root}}$. Finally, we penalize the deviation from support polygon

$$
r_t^{sp} = \exp\left(s_{sp} d_{sp}(\mathbf{x}^{\text{root}}, \mathbf{p}^{\text{root}}, \mathbf{p})^2\right), \tag{5.6}
$$

where $d_{sp}$ is the minimal distance to the support polygon. We only calculate $d_{sp}$ when the robot is required to make at least three contacts: otherwise, $d_{sp}$ is defined as zero so that $r_t^{sp} = 1$. In auxiliary term, we penalize excessive motions with the acceleration penalty term:

$$
r_t^{acc} = \exp\left(s_{acc} \sum_j \|\ddot{\mathbf{p}}_t\|^2\right). \tag{5.7}
$$

To reflect our intention to emphasize the main mimicking term, we set weight terms $w^{main} = 0.9, w^{acc} = 0.1$ for all learning stage. The scaling coefficients are set to $s_p = 1.0$, $s_e = 20.0$, $s_{rp} = 20.0$, $s_{ro} = 5.0$ and $s_{sp} = 10.0$ respectively.

**Early termination.** As proved in many works s [6, 80, 57, 81], the early termination during policy training accelerates the learning speed incredibly. We trigger the early termination when the robot trunk touches the ground and self-collision happens.

**Learning process.** We utilize Proximal Policy Optimization(PPO) [82] to optimize our control policies. Each policy is consist of feedforward networks of two hidden layers with $256$ ReLu neurons. Our PPO formulation has a clipping range of $0.2$, learning rate of $0.00005$, the discount factor is $\gamma = 0.95$, and the GAE parameter is $\lambda = 0.95$. The minibatch size is 128 for the policy and value network. The max gradient norm is set to $0.5$.

## 5.3 Curriculum Learning over Tasks and Difficulties

While the policy learned from the above formulation works skillfully for a single motion task, our aim of learning a versatile policy seemingly for multiple tasks remains a challenge. We observe that naive learning will result in a policy that generates conservative movement that is stuck in a steady position to avoid body falls while not trying to track the target motion. To manage this problem, we train an expert policy for the given motion state with a curriculum to expand the range of motion and the number of tasks.

We divide the curriculum in to two parts: difficulties and the number of tasks. Therefore, we sort all the robot reference motions based on two criteria: a task type as a primary and difficulty of the task as a secondary. For instance, we train a *stand* expert policy by training on the *tilting-at-stand* task first and expanding the task set by adding the *manipulation-at-stand* task. For each task, we gradually increase the difficulty by manipulating the range of reference motions. Likewise, we train a *walking* expert by expanding the curriculum from the *walking forward* to the *turning left/right*, with increasing turning rates.

## 5.4 Hierarchical Control with States

Our system is designed to perform motions with multiple tasks including *tilting*, *manipulation*, and *locomotion* over three different robot states, *stand*, *sit*, and *walk*. The states and tasks yields different combinations such as *tilting-at-stand* or *manipulation-at-sit*. This leads us to develop three expert policies for each robot state instead of a monolithic policy. We produce a special transition controller which is called when the motion selector of the motion retargeting module detects transitions. There are multiple methods to build a transition controller such as model-based control or reinforcement learning. However, we reuse the existing motion imitation framework as a controller. The transition takes 1 to 3 seconds depending on the tasks and the robot's

state. During transitions, the robot performs the predefined policy while ignoring human motions.

## 5.5 Domain Randomization

The dynamics of the simulation and the real world contain a certain gap which decreases the performance of simulation learned control policies that are deployed on a real robot. We bring the Domain Randomization method [41], which randomizes dynamics parameters during the training of the policy in simulation to obtain more robust control in the real world. The randomized parameters and their ranges are identified in Table 5.1. We also applied the curriculum to these parameters similar to the method mentioned in section C to facilitate the learning. Detailed procedures are well mentioned in previous works [6, 8].

| **Parameters** | **Range** | **Unit** |
|---|---|---|
| Link Mass | [0.75, 1.25] X default | kg |
| Ground Friction Coefficients | [0.5, 1.5] | 1 |
| Proportional Gain | [0.7, 1.3] X default | N/rad |
| Derivative Gain | [0.7, 1.3] X default | N·s/rad |
| Communication Delay | [0, 0.016] | sec |
| Ground Slope | [0, 0.14] | rad |

Table 5.1: Domain Randomization Parameters

# Chapter 6

# Results and Analysis

We design the analysis process to evaluate our framework from three perspectives. First, we demonstrate our result of the system's ability to perform a set of tasks in simulation and real environments. Second, we conduct ablation studies that evaluate the effectiveness of our system components. Finally, we compare our system with similar motion-based interfaces in a qualitative manner.

## 6.1 Experimental Setup

Our system is tested on an A1 quadrupedal robot [83], which has twelve actuated degrees of freedom for all legs(three for each leg) and six under-actuated degrees of freedom for the root. We utilize 76 to 522 matching data pairs for training the retargeting module varying by the tasks. We train each control policy using 1.2 billion samples in the RaiSim [84] physics simulator which serves a stable environment. Our experiment was conducted on a stable desktop computer with Intel 16 core 3.60GHz i9-9900K CPU and GeForce RTX 2070 SUPER GPU. We capture an operator's motion using a Kinect [12].

The simulation demonstrations are performed in interactive response. On the other hand, we conduct real-world experiments in two modes. The first mode is the live mode

that controls a real robot interactively in an end-to-end fashion which is in fact the same way as simulation demos. The second mode is the replay mode that controls the robot with the prerecorded human motion trajectories. The reason to build the replay mode is that fluctuation control delays could harm a real robot and an operator. We do not feed the future trajectory information even in the replay mode to grant the same condition with the live mode. We annotate the experiment modes in the manuscript and supplemental videos to clarify the demonstration.

## 6.2 Motion Performance

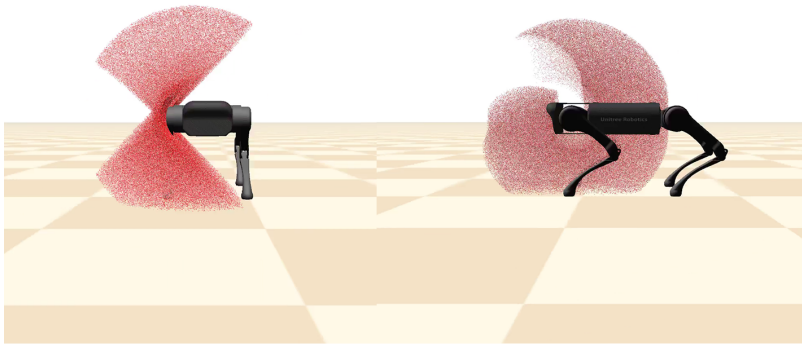
**Individual tasks.** First, we demonstrate a diverse set of motor tasks on the individual state for A1 using human motion control. In the *stand* state, a robot manipulates its end-effector while simultaneously tilting its body. The range of tilting varies from $-40^\circ$ to $40^\circ$ for all $x, y, z$ axes, which is larger than the tilting range of the manufacturer's controller: $-20^\circ$ to $20^\circ$ for pitch and roll and $-28^\circ$ to $28^\circ$ for yaw. We show that simultaneous manipulation and tilting provide a broader workspace. The manipulation space with tilting delivers approximately $2.7$ times larger touching space volume than manipulation without tilting. We generate point clouds as in Figure 6.1 to compare the working space volume. In the *sit* state, a robot is allowed to use both arms so that it can reach higher targets. The capability of tilting while balancing in the *sit* state is $30^\circ$, $15^\circ$, and $7^\circ$ in pitch, roll, and yaw axes, respectively. Our controller enables a robot to walk at the speed of $0.0$m/s to $0.97$ m/s with the maximum turning rate of $15^\circ$/second. The quality of motion shows lesser performance than other controllers since our controller should prepare for the abrupt change of motion at any moment. We illustrate the motion tasks in Figure 6.2.

**Composite tasks.** Our control system enables a user to smoothly switch between tasks by building a hierarchical structure. As illustrated in Figure 6.3 top, we conduct a simulation experiment with the following sequence: (1) tilting the body to tell a salute, (2)

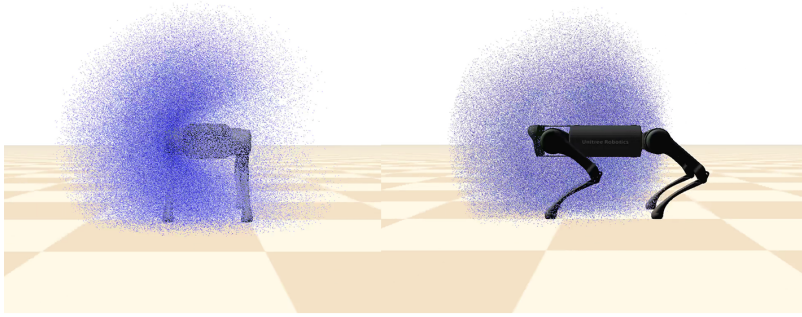
locomoting forward to reach a target of 3m, (3) weaving a thrown orange-colored ball by crouching, (4) manipulating the target and (5) touching another target high in the air. In the same way, we set up the replay mode demonstration to execute the following tasks: (1) smashing a tennis ball located at $0.42$m height, (2) sitting to hit a bone hanging high at $0.8$m height, and (3) dodging a thrown tennis ball (Figure 6.3 middle). The robot must sit to achieve (2) since the target bone is too high to reach while *standing*. These scenarios present the seamless transition capability of our control system. We conduct the final demo that controls a real robot to push the box to the target position(X-mark on the floor) in real-time. The challenge of this task comes from the fact that the target box has located a distance from the initial position. To achieve the goal, we control the robot by repeating the following control tasks: (1) walk near the box and (2) push the box toward the target (Figure 6.3 bottom).

**Control responsiveness.** To conduct the real-time control, responsiveness issue emerges to be the most important criteria, which is even more critical for a quadrupedal robot with a floating base. Our control system consists of two main stages: *motion reconstruction* and *control inference*. In the *motion reconstruction* stage, the system infers human 3D poses from the Kinect. The important technical components of our system is included in the *control inference* stage. However, our system responsiveness bottleneck mainly occurs at the *motion reconstruction* stage which affects for our $30$ Hz control loop. The main control loop component, entire inference task, consumes less than $0.01$ second, which is fast enough. We stabilize the control frequency from *motion reconstruction* stage by skipping Kinect reading when the delay is significant and reusing the existing human motions from the previous frame.

**Different mapping styles.** Support of different styles of mapping for the same task shows the flexibility of our system. As shown in Figure 6.4, we generate two different retargeting functions with the human motion of (1) in-place marching and (2) cyclic hand gestures. Both mapping styles demonstrate successful marching motions in the simulation.

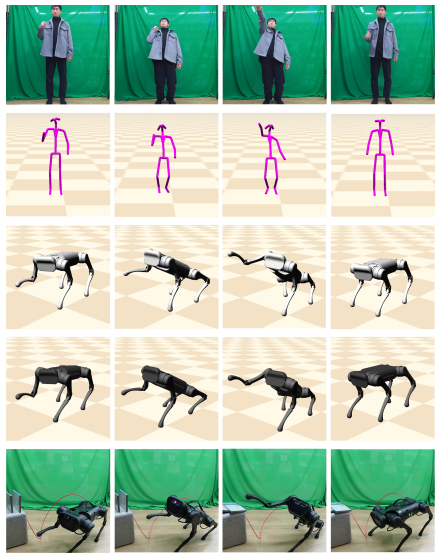


(a) Only manipulation

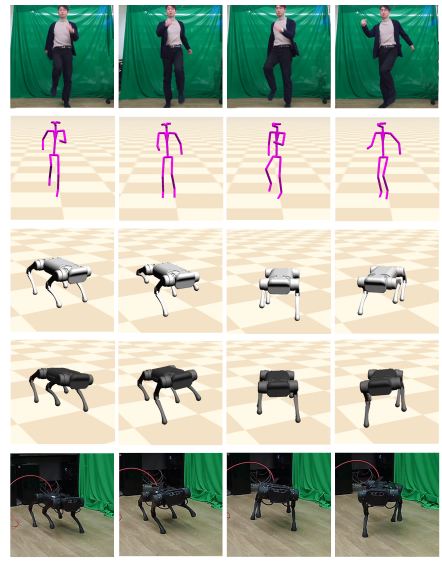


(b) Tilting and manipulation

Figure 6.1: Point clouds that illustrate the workspace of the right front leg when performing only manipulation(red) and manipulation with tilting(bottom) during *standing*. The robot can reach approximately 2.7 larger volumes by simultaneously tilting its body.



(a) Tilting and Manipulation



(b) Walking

Figure 6.2: Individual task motion of the (**a**) *tilting and manipulation* and (**b**) *walking* tasks. From the top row, we illustrate human video footage, human skeleton, retargeted robot motion, simulated motion, and real robot motion, at the corresponding time frames.

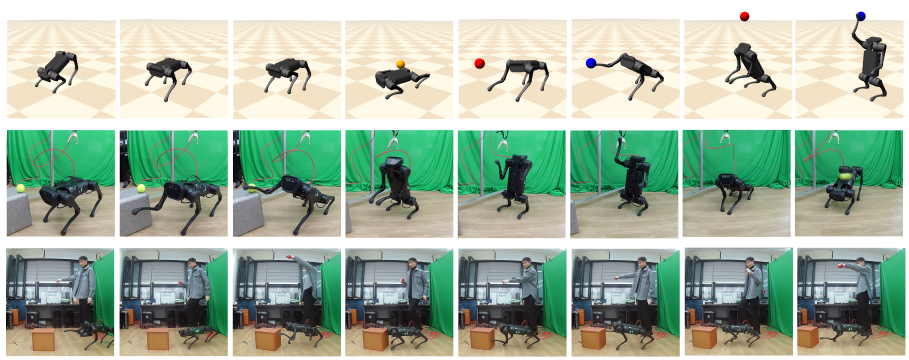
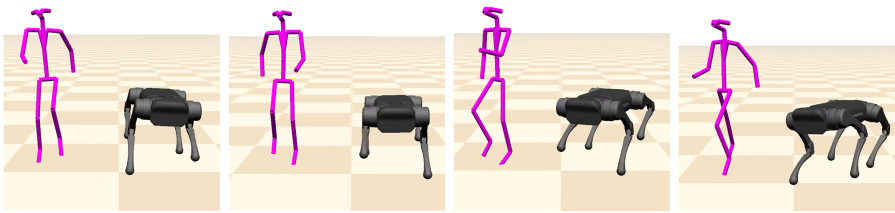
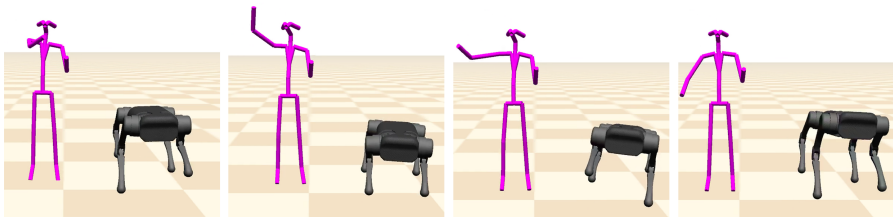


Figure 6.3: Composite tasks in simulation (**top**), real-world (**middle** (replay mode) and **bottom** (live mode)).



(A) Retarget human walking to robot walking



(B) Retarget human arm gesture to robot walking

Figure 6.4: Different styles of mapping for *walking*. The **top** shows a mapping with *in-place marching* and the **bottom** shows a mapping with *hand gestures*.

**Semantic mapping with manual features.** Similarly, we can manually tune a mapping not from selecting direct motion but by selecting features for retargeting. For instance, mapping for the *walking* task with explicit notions can be extracted to the feature parameters such as gait height, swing angle or gait patterns. Although this explicit mapping offers slightly better motion quality, this requires domain-specific knowledge of the task.

## 6.3 Analysis

**Contact and temporal consistency.** The importance of *contact consistency* and *temporal consistency* corrections are evaluated through an ablation study. We conduct the test environment to track 10 seconds of noisy trajectories perturbed from the ground truth robot motions by 5120 test episodes. We compare ablations based on the success time ratio, which is the ratio of the termination time to the maximum episode length. As shown in Figure 6.5, both components are critical for the system. While contact

consistency correction shows more importance for *standing* and *sitting* motions, temporal consistency seems vital for *walking* motions.

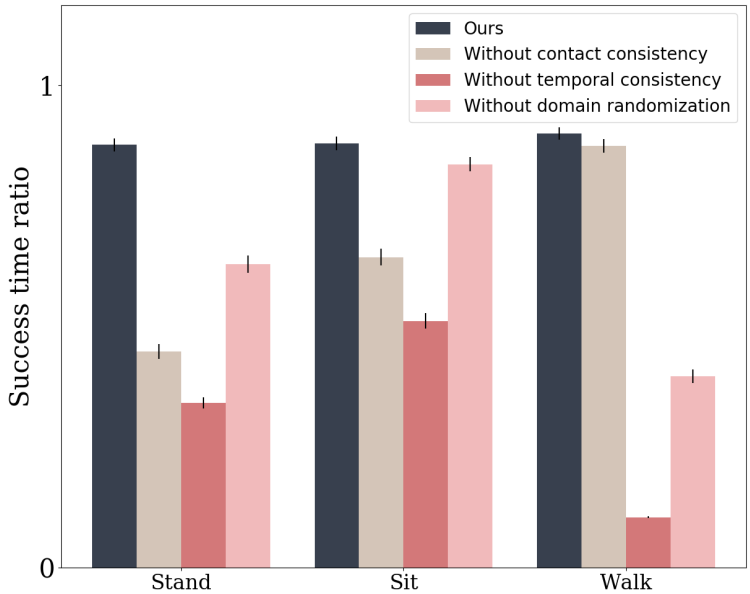



Figure 6.5: Average success time ratio, which is the ratio of the termination time to the maximum episode duration. We conduct an ablation study with contact consistency, temporal consistency, and domain randomization to evaluate their effectiveness.

**Curriculum learning.** The presence of curriculum learning is essential for obtaining the best motion control performance. The absences of the curriculum learning show the tendency of survival motion until the last frame while showing conservative behaviors. This leads us to compare the quality of motions to show the ablation as in Figure 6.6. They illustrate the conservative behaviors of the policies without curriculum, which put all the feet on the ground and do not attempt to reach the target.

**Domain adaptation.** Our system is able to overcome the sim-to-real gap by introducing Domain randomization (DR). We evaluate its effectiveness by measuring the success time ratio in the same way as the consistency ablation study with the presence of randomized dynamics. Figure 6.5 illustrates that DR is essential for our system. In

addition, we conduct the sim-to-real experiment, where the policy without DR cannot complete the given motion and leads to failure.

(a) Reference Manipulation motion

(b) Agent learned with curriculum

(c) Agent learned without curriculum

Figure 6.6: Snapshots of the robot control to show the effectiveness of curriculum learning. The physically simulated agent tries to mimic the motion of reference (**Top**). While the policy trained with curriculum successfully mimic the reference (**Middle**), the policy without curriculum is stuck in local optimum (**Bottom**).

**Importance of future reference.** We observe that our controller shows a motion quality that is not as good as we expected. We hypothesize that the poor tracking performance is caused by our real-time motion tracking without information about the future trajectory. We train an additional policy that takes the additional future information to track the motion to verify our hypothesis. We compare the motion quality with our original agent. The experiment shows that the policy with future information generates more stable motions.

| **Criteria** | **(A)** | **(B)** | **(C)** | **(D)** | **Ours** |
|---|---|---|---|---|---|
| **Mapping dimension** | 2D | 3D | 3D | 3D | 3D |
| **Real-Time** | Y | N | N | Y | Y |
| **Dynamics** | Y | N | Y | N | Y |
| **Mapping Flexibility** | N | N | N | Y | Y |
| **Sim2Real** | N | N | N | N | Y |

Table 6.1: Comparison with previous human to non-humanoid control methods (A) Kim et al. [1], (B)Dontcheva et al. [2], (C)Yamane et al. [3] (D)Seol et al. [4]

## 6.4 Comparison to Other Methods

Our method presents the novelty compared to the previous human to non-humanoid control methods. As illustrated in Table 6.1, our method enables various aspects that previous methods can't accomplish. Kim et al. [1] showed the motion mapping that corresponds to the dynamical systems of two different morphologies, but it is limited to 2D cyclic motions. Dontcheva et al. [2] suggested the concept of detecting the human gesture to explore the matched motion pair of characters. This method carries the limitation that a lack of dynamics. Yamane et al. [3] showed the success in mapping human motions to non-humanoid characters with natural movements. However, this approach obtains heavy calculation therefore, not aiming to get real-time control. Seol et al. [4] showed a flexible retargeting scheme that contains both agility and semantics by feature mapping but is limited to kinematic animations.

Our framework shows strength over flexible mapping and real-time control compared to the presented methods. Thus, our framework supports a wide range of tasks without domain knowledge of task-specific dynamics due to the motion retargeting schemes. We also perform robust control in a real robot by adopting motion imitation learning with domain randomization.

# Chapter 7

# Conclusion And Future Work

We presented a novel human motion control system that enables a user to control quadrupedal robots using body motion. Our system mainly consists of two parts: a motion retargeting module and a motion imitation policy. The motion retargeting module converts the captured human motion into robot motion with preserving semantics via supervised learning manner and post-processing techniques. Then, we train a motion imitation control policy that tracks the given retargeted motion while adapting to the physics environment using deep reinforcement learning. We further improve the control performance by leveraging the concept of a set of experts and curriculum learning. We demonstrate the evaluation of the motion control system on both simulation and the real world by conducting various tasks, including standing, tilting, sitting, manipulating, walking, or their combinations.

Our system shows a few limitations. First, we observe that the significant delay of the motion capture processing system, Kinect, shows $0.01$s to $0.06$s latency. This particularly for a real robot, prevents us from conducting more real-world experiments in the live mode by affecting the control frequency loop. While we mitigate this issue by randomizing control frequency during training, some raised issues still remain. Moreover, the instability of the Kinect estimation system occasionally observes unexpected operator motions that lead to control failure. We suggest that a motion capture system

with better stability and higher rates will be more suitable for real-time motion control applications.

Another limitation comes from the lack of future trajectory that severly degrades the motion quality. Our control policies track the versatile reference motions with frequent changes in motion that often resulting in conservative policies with not good motion quality. We suggest to improve this issue by referencing the research that predict user intentions from history and leverage them in the control policies.

We conduct our experiments that the user and the robot are in the same space. However, we can expand this setting to achieve the higher goal of developing robotic workers in dangerous environments. In the future study, we suggest combining the proposed system with virtual reality devices to deliver a more immersive control scheme. Our proposed extensions raise new research questions: which robot sensory information is essential for users to operate the robot with proper intention or how to deal with increased delay.

# Bibliography


[1] N. H. Kim, Z. Xie, and M. Panne, "Learning to correspond dynamical systems," in *Learning for Dynamics and Control*, pp. 105–117, PMLR, 2020.

[2] M. Dontcheva, G. Yngve, and Z. Popović, "Layered acting for character animation," *ACM Trans. Graph.*, vol. 22, p. 409–416, jul 2003.

[3] K. Yamane, Y. Ariki, and J. Hodgins, "Animating non-humanoid characters with human motion data," in *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 169–178, 2010.

[4] Y. Seol, C. O'Sullivan, and J. Lee, "Creature features: online motion puppetry for non-human characters," in *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 213–221, 2013.

[5] J. Ramos and S. Kim, "Dynamic locomotion synchronization of bipedal robot and human operator via bilateral feedback teleoperation," *Science Robotics*, vol. 4, no. 35, p. eaav4282, 2019.

[6] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Trans. Graph.*, vol. 37, pp. 143:1–143:14, July 2018.

[7] X. B. Peng, E. Coumans, T. Zhang, T.-W. E. Lee, J. Tan, and S. Levine, "Learning agile robotic locomotion skills by imitating animals," in *Robotics: Science and Systems*, 07 2020.

[8] Z. Li, X. Cheng, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Reinforcement learning for robust parameterized locomotion control of bipedal robots," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–7, 2021.

[9] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019.

[10] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science Robotics*, vol. 5, no. 47, 2020.

[11] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," *arXiv preprint arXiv:2107.04034*, 2021.

[12] "Azure kinect dk – develop ai models: Microsoft azure," 2018.

[13] M. Raibert, K. Blankespoor, G. Nelson, and R. Playter, "Bigdog, the rough-terrain quadruped robot," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 10822–10825, 2008.

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

[15] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, "Mit cheetah 3: Design and control of a robust, dynamic quadruped robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2245–2252, 2018.

[16] R. Bischoff and V. Graefe, "Demonstrating the humanoid robot hermes at an exhibition: A long-term dependability test," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems; Workshop on Robots at Exhibitions*, Lausanne, Switzerland, 2002.

[17] Z. Xie, G. Berseth, P. Clary, J. Hurst, and M. van de Panne, "Feedback control for cassie with deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1241–1246, IEEE, 2018.

[18] K. Kim, P. Spieler, E.-S. Lupu, A. Ramezani, and S.-J. Chung, "A bipedal walking robot that can fly, slackline, and skateboard," *Science Robotics*, vol. 6, no. 59, p. eabf8136, 2021.

[19] M. H. Raibert, "Hopping in legged systems — modeling and simulation for the two-dimensional one-legged case," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-14, no. 3, pp. 451–463, 1984.

[20] H. Geyer, A. Seyfarth, and R. Blickhan, "Positive force feedback in bouncing gaits?," *Proceedings. Biological sciences / The Royal Society*, vol. 270, pp. 2173–83, 11 2003.

[21] T. Horvat, K. Melo, and A. J. Ijspeert, "Model predictive control based framework for com control of a quadruped robot," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3372–3378, 2017.

[22] C. Gehring, S. Coros, M. Hutter, M. Bloesch, M. A. Hoepflinger, and R. Siegwart, "Control of dynamic gaits for a quadrupedal robot," in *2013 IEEE International Conference on Robotics and Automation*, pp. 3287–3292, 2013.

[23] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the mit cheetah 3 through convex model-predictive control," in *2018*

*IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–9, 2018.

[24] Y. Ding, A. Pandala, C. Li, Y.-H. Shin, and H.-W. Park, "Representation-free model predictive control for dynamic motions in quadrupeds," *IEEE Transactions on Robotics*, vol. 37, no. 4, pp. 1154–1171, 2021.

[25] F. Jenelten, J. Hwangbo, F. Tresoldi, C. D. Bellicoso, and M. Hutter, "Dynamic locomotion on slippery ground," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4170–4176, 2019.

[26] J. Carius, R. Ranftl, V. Koltun, and M. Hutter, "Trajectory optimization for legged robots with slipping motions," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 3013–3020, 2019.

[27] M. Focchi, R. Orsolino, M. Camurri, V. Barasuol, C. Mastalli, D. G. Caldwell, and C. Semini, *Heuristic Planning for Rough Terrain Locomotion in Presence of External Disturbances and Variable Perception Quality*, pp. 165–209. Cham: Springer International Publishing, 2020.

[28] M. Focchi, V. Barasuol, M. Frigerio, D. G. Caldwell, and C. Semini, *Slip Detection and Recovery for Quadruped Robots*, pp. 185–199. Cham: Springer International Publishing, 2018.

[29] C. Yang, B. Zhang, J. Zeng, A. Agrawal, and K. Sreenath, "Dynamic legged manipulation of a ball through multi-contact optimization," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7513–7520, 2020.

[30] S. Lee, M. Park, K. Lee, and J. Lee, "Scalable muscle-actuated human simulation and control," *ACM Trans. Graph.*, vol. 38, jul 2019.

[31] S. Park, H. Ryu, S. Lee, S. Lee, and J. Lee, "Learning predict-and-simulate policies from unorganized human motion data," *ACM Trans. Graph.*, vol. 38, no. 6, 2019.

[32] R. Yu, H. Park, and J. Lee, "Figure skating simulation from video," in *Computer graphics forum*, vol. 38, pp. 225–234, Wiley Online Library, 2019.

[33] A. Iscen, K. Caluwaerts, J. Tan, T. Zhang, E. Coumans, V. Sindhwani, and V. Vanhoucke, "Policies modulating trajectory generators," in *2nd Annual Conference on Robot Learning, CoRL 2018*, pp. 916–926, 2018.

[34] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke, "Sim-to-real: Learning agile locomotion for quadruped robots," *arXiv preprint arXiv:1804.10332*, 2018.

[35] T. Li, J. Won, S. Ha, and A. Rai, "Model-based motion imitation for agile, diverse and generalizable quadupedal locomotion," *arXiv preprint arXiv:2109.13362*, 2021.

[36] Z. Xie, X. Da, B. Babich, A. Garg, and M. van de Panne, "Glide: Generalizable quadrupedal locomotion in diverse environments with a centroidal model," *CoRR*, vol. abs/2104.09771, 2021.

[37] T. Haarnoja, S. Ha, A. Zhou, J. Tan, G. Tucker, and S. Levine, "Learning to walk via deep reinforcement learning," *arXiv preprint arXiv:1812.11103*, 2018.

[38] S. Ha, P. Xu, Z. Tan, S. Levine, and J. Tan, "Learning to walk in the real world with minimal human effort," *arXiv preprint arXiv:2002.08550*, 2020.

[39] L. Smith, J. C. Kew, X. B. Peng, S. Ha, J. Tan, and S. Levine, "Legged robots that keep on learning: Fine-tuning locomotion policies in the real world," 2021.

[40] W. Yu, J. Tan, Y. Bai, E. Coumans, and S. Ha, "Learning fast adaptation with meta strategy optimization," 2020.

[41] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018.

[42] W. Yu, C. K. Liu, and G. Turk, "Policy transfer with strategy optimization," in *International Conference on Learning Representations*, 2019.

[43] OpenAI, M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba, "Learning dexterous in-hand manipulation," 2019.

[44] Q. Vuong, S. Vikram, H. Su, S. Gao, and H. I. Christensen, "How to pick the domain randomization parameters for sim-to-real transfer of reinforcement learning policies?," 2019.

[45] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation policies from data," 2019.

[46] N. Ruiz, S. Schulter, and M. Chandraker, "Learning to simulate," 2019.

[47] D. Holden, T. Komura, and J. Saito, "Phase-functioned neural networks for character control," *ACM Trans. Graph.*, vol. 36, jul 2017.

[48] K. Bergamin, S. Clavet, D. Holden, and J. R. Forbes, "Drecon: Data-driven responsive control of physics-based characters," *ACM Trans. Graph.*, vol. 38, Nov. 2019.

[49] S. Starke, H. Zhang, T. Komura, and J. Saito, "Neural state machine for character-scene interactions," *ACM Trans. Graph.*, vol. 38, nov 2019.

[50] D. Holden, O. Kanoun, M. Perepichka, and T. Popa, "Learned motion matching," *ACM Trans. Graph.*, vol. 39, jul 2020.

[51] S. Starke, Y. Zhao, T. Komura, and K. Zaman, "Local motion phases for learning multi-contact character movements," *ACM Trans. Graph.*, vol. 39, jul 2020.

[52] S. Starke, Y. Zhao, F. Zinno, and T. Komura, "Neural animation layering for synthesizing martial arts movements," *ACM Trans. Graph.*, vol. 40, jul 2021.

[53] L. Liu and J. Hodgins, "Learning to schedule control fragments for physics-based characters using deep q-learning," *ACM Trans. Graph.*, vol. 36, jun 2017.

[54] L. Liu and J. Hodgins, "Learning basketball dribbling skills using trajectory optimization and deep reinforcement learning," *ACM Trans. Graph.*, vol. 37, jul 2018.

[55] N. Heess, D. TB, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, Z. Wang, S. M. A. Eslami, M. Riedmiller, and D. Silver, "Emergence of locomotion behaviours in rich environments," 2017.

[56] J. Won, J. Park, K. Kim, and J. Lee, "How to train your dragon: Example-guided control of flapping flight," *ACM Trans. Graph.*, vol. 36, nov 2017.

[57] J. Won, J. Park, and J. Lee, "Aerobatics control of flying creatures via self-regulated learning," *ACM Trans. Graph.*, vol. 37, dec 2018.

[58] A. Clegg, W. Yu, J. Tan, C. K. Liu, and G. Turk, "Learning to dress: Synthesizing human dressing motion via deep reinforcement learning," *ACM Trans. Graph.*, vol. 37, dec 2018.

[59] S. Min, J. Won, S. Lee, J. Park, and J. Lee, "Softcon: Simulation and control of soft-bodied animals with biomimetic actuators," *ACM Trans. Graph.*, vol. 38, no. 6, 2019.

[60] Y.-S. Luo, J. H. Soeseno, T. P.-C. Chen, and W.-C. Chen, "Carl: Controllable agent with reinforcement learning for quadruped locomotion," *ACM Trans. Graph.*, vol. 39, jul 2020.

[61] S. Lee, S. Lee, Y. Lee, and J. Lee, "Learning a family of motor skills from a single motion clip," *ACM Trans. Graph.*, vol. 40, no. 4, 2021.

[62] A. Safonova, N. Pollard, and J. K. Hodgins, "Optimizing human motion for the control of a humanoid robot," *Proc. Applied Mathematics and Applications of Mathematics*, vol. 78, pp. 18–55, 2003.

[63] W. Suleiman, E. Yoshida, F. Kanehiro, J.-P. Laumond, and A. Monin, "On human motion imitation by humanoid robot," in *2008 IEEE International Conference on Robotics and Automation*, pp. 2697–2704, 2008.

[64] I. Baran, D. Vlasic, E. Grinspun, and J. Popović, "Semantic deformation transfer," *ACM Trans. Graph.*, vol. 28, jul 2009.

[65] S. Albrecht, K. Ramirez-Amaro, F. Ruiz-Ugalde, D. Weikersdorfer, M. Leibold, M. Ulbrich, and M. Beetz, "Imitating human reaching motions using physically inspired optimization principles," in *2011 11th IEEE-RAS International Conference on Humanoid Robots*, pp. 602–607, IEEE, 2011.

[66] Y. Zheng and K. Yamane, "Adapting human motions to humanoid robots through time warping based on a general motion feasibility index," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6281–6288, 2015.

[67] J. P. Whitney, T. Chen, J. Mars, and J. K. Hodgins, "A hybrid hydrostatic transmission and human-safe haptic telepresence robot," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 690–695, 2016.

[68] J. Ramos and S. Kim, "Improving humanoid posture teleoperation by dynamic synchronization through operator motion anticipation," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5350–5356, 2017.

[69] Y. Ishiguro, K. Kojima, F. Sugai, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba, "Bipedal oriented whole body master-slave system for dynamic secured lo-

comotion with lip safety constraints," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 376–382, 2017.

[70] J. Koenemann and M. Bennewitz, "Whole-body imitation of human motions with a nao humanoid," in *2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 425–425, 2012.

[71] S. Choi, M. Pan, and J. Kim, "Nonparametric motion retargeting for humanoid robots on shared latent space," *Proceedings of Robotics: Science and Systems (R: SS)*, 2020.

[72] S. Choi, M. J. Song, H. Ahn, and J. Kim, "Self-supervised motion retargeting with safety guarantee," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8097–8103, 2021.

[73] M. Bajracharya, J. Borders, D. Helmick, T. Kollar, M. Laskey, J. Leichty, J. Ma, U. Nagarajan, A. Ochiai, J. Petersen, K. Shankar, K. Stone, and Y. Takaoka, "A mobile manipulation system for one-shot teaching of complex tasks in homes," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11039–11045, 2020.

[74] M. Arduengo, A. Arduengo, A. Colomé, J. Lobo-Prat, and C. Torras, "A robot teleoperation framework for human motion transfer," 2019.

[75] C. Yang, K. Yuan, Q. Zhu, W. Yu, and Z. Li, "Multi-expert learning of adaptive legged locomotion," *Science Robotics*, vol. 5, no. 49, 2020.

[76] J. Won, D. Gopinath, and J. Hodgins, "A scalable approach to control diverse behaviors for physically simulated characters," *ACM Trans. Graph.*, vol. 39, jul 2020.

[77] X. B. Peng, M. Chang, G. Zhang, P. Abbeel, and S. Levine, "MCP: learning composable hierarchical control with multiplicative compositional policies," *CoRR*, vol. abs/1905.09808, 2019.

[78] K. Frans, J. Ho, X. Chen, P. Abbeel, and J. Schulman, "Meta learning shared hierarchies," *CoRR*, vol. abs/1710.09767, 2017.

[79] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, (Cambridge, MA, USA), p. 1057–1063, MIT Press, 1999.

[80] X. B. Peng, A. Kanazawa, J. Malik, P. Abbeel, and S. Levine, "Sfv: Reinforcement learning of physical skills from videos," *ACM Trans. Graph.*, vol. 37, Nov. 2018.

[81] W. Yu, G. Turk, and C. K. Liu, "Learning symmetric and low-energy locomotion," *ACM Transactions on Graphics*, vol. 37, p. 1–12, Aug 2018.

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

[83] "A1," 2020.

[84] J. Hwangbo, J. Lee, and M. Hutter, "Per-contact iteration method for solving contact dynamics," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 895–902, 2018.

# 초 록

사람의 모션을 이용한 로봇 컨트롤 인터페이스는 사용자의 직관과 로봇의 모터 능력을 합하여 위험한 환경에서 로봇의 유연한 작동을 만들어낸다. 하지만, 휴머노이드 외의 사족보행 로봇이나 육족보행 로봇을 위한 모션 인터페이스를 디자인 하는 것은 쉬운일이 아니다. 이것은 사람과 로봇 사이의 형태 차이로 오는 다이나믹스 차이와 제어 전략이 크게 차이나기 때문이다. 우리는 사람 사용자가 움직임을 통하여 사족보행 로봇에서 부드럽게 여러 과제를 수행할 수 있게끔 하는 새로운 모션 제어 시스템을 제안한다. 우리는 우선 캡쳐한 사람의 모션을 상응하는 로봇의 모션으로 리타겟 시킨다. 이때 상응하는 로봇의 모션은 유저가 의도한 의미를 내포하게 되며, 우리는 이를 지도학습 방법과 후처리 기술을 이용하여 가능케 하였다. 그 뒤 우리는 모션을 모사하는 학습을 커리큘럼 학습과 병행하여 주어진 리타겟된 참조 모션을 따라가는 제어 정책을 생성하였다. 우리는 ”전문가 집단”을 학습함으로 모션 리타게팅 모듈과 모션 모사 모듈의 성능을 크게 증가시켰다. 결과에서 볼 수 있듯, 우리의 시스템을 이용하여 사용자가 사족보행 로봇의 서있기, 앉기, 기울이기, 팔 뻗기, 걷기, 돌기와 같은 다양한 모터 과제들을 시뮬레이션 환경과 현실에서 둘 다 수행할 수 있었다. 우리는 연구의 성능을 평가하기 위하여 다양한 분석을 하였으며, 특히 우리 시스템의 각각의 요소들의 중요성을 보여줄 수 있는 실험들을 진행하였다.

**주요어**: 심층 강화학습, 컴퓨터 애니메이션, 컴퓨터 그래픽스, 로보틱스, 휴먼-로봇 상호작용

**학번**: 2019-25917

# 감사의 글

재미있는 연구를 하고 싶다는 막연한 생각을 가졌던 2019년의 제가 운동연구실로 합류하여 연구를 한지 3년이 지났습니다. 생소한 분야를 더 깊게 배우려다 보니 과연 내가 논문을 쓸 수 있을까 걱정도 했지만 이제희교수님의 지도 덕분에 잘 헤쳐올 수 있던것 같습니다. 교수님께 대학원 지도를 받아 영광이었습니다.

이제희 교수님 못지 않게 지도를 해주신 하세훈 교수님께도 감사인사를 드리고자합니다. 좋은 논문 아이디어를 같이 고민하고 연구를 지도해주셔서 세상이 놀랄만한 결과를 만들었던것 같습니다. 먼 미국에서 매주 빠지지 않고 회의를 통해 지도해주신 점에 감사드립니다.

갑작스레 지도교수님이 변경되었음에도 저희를 따뜻하게 맞아주신 서진욱교수님께도 감사드립니다. 덕분에 마지막 학기에 석사 논문을 잘 준비할 수 있었습니다.

제가 성격이 살갑지 않음에도 편하게 연구실에서 지낼수 있게 해준 연구실 사람들 모두에게도 감사합니다. 수환이형, 세희누나, 승환이형, 재동이형, 민석이형, 정남이형, 세영이누나, 선민누나, 필식이, 용우, 민효형, 지예덕분에 연구실에서 즐거운 나날을 보낼 수 있었고 힘든 일도 이겨나갈 수 있었습니다. 마지막으로, 연구실이 잘 돌아가게끔 수고해주시는 승아쌤께도 감사드립니다.

거의 10년째 본가에서 나와서 살고 있음에도 무한한 신뢰와 사랑을 보내준 가족을 빼 놓을수 없을 것 같습니다. 아버지, 어머니, 건우까지 정말 사랑합니다. 항상 저의 선택을 존중해주시고 응원해주셔서 난관들을 해쳐나갈 수 있었습니다.

10년째 항상 같이 있어준 내 친구 류찬형, 힘들 때 푸념도 들어주고 즐거운 시간을 만들어 나를 웃게해주어서 정말 고맙습니다. 석사 논문 및 취업을 준비할 때 옆에서 함께 있어준 한별이에게도 고맙다는 인사 전합니다.

많은 사람들에게 과분한 사랑과 응원을 받았습니다. 다 언급드리지 못했지만 3년동안 도움을 주신 모든분들께 감사드립니다.