SSI-DT79165207

Doctoral Thesis

Massive Object Transportation by Robots

(ロボットによる重量物搬送)

Jaesung Yang

August, 2019

Division of Systems Science and Informatics
Graduate School of Information Science and Technology
Hokkaido University

Doctoral Thesis
submitted to Graduate School of Information Science and Technology,
Hokkaido University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy.

Jaesung Yang

Thesis Committee: Prof. Atsushi Konno
Prof. Masahiko Onosato
Prof. Satoshi Ogasawara

Massive Object Transportation by Robots*

Jaesung Yang

Abstract

Robots have been expected to replace human in performing dull, dirty, dangerous, demeaning, and difficult tasks. Transportation of massive object is one of those tasks, which is hard, dangerous, or impossible in some cases for human to do. In space application, a robot manipulator must manipulate highly massive objects such as satellites and modules of a space structure. Space robotic arms are generally operated by a human operator locally from a space vehicle or remotely from the earth. The main problems in the manipulation of a massive object in space by teleoperating a space robot from the earth are communication delay and the unexpected excessive force generated between the space robot and the massive object. On the other hand, in applications on earth, there are a lot of demands for multi-robot cooperative massive object transportation. If an object on earth is too heavy for multiple robots to lift up, the object must be transported by pushing and pulling. The main problem in the multi-robot cooperative massive object transportation is how to exert effective force on the object to push or pull for robots while avoiding slippage between wheels/feet and the ground.

This thesis discusses two types of massive object transportation: massive object transportation in space by teleoperating space robots, and massive object transportation on earth by pushing/pulling by autonomous humanoid robots.

For the massive object transportation in space by teleoperation, force mitigation of the contact force, and a virtual environment considering the dynamics were proposed to solve the large contact force generation problem and the communication delay problem. Hardware-In-the-Loop-Simulations (HILS) were performed

*Doctoral Thesis, Division of Systems Science and Informatics, Graduate School of Information Science and Technology, Hokkaido University, SSI-DT79165207, August 9, 2019.

with a scenario, in which a space robotic arm was teleoperated to manipulate a massive object of 6,000 kg. The results of the HILS show the validity of the proposed methods.

For the massive object transportation on earth by multiple humanoid robots, a method of generating an optimal robot motion was proposed to effectively exert force on the massive object to push or pull, keeping contacts between the feet of the robots and the ground. In the robot motion generation, frictional force generated at the feet from the ground and reaction force generated at the hands from the object must be estimated in real-time. In order to estimate the frictional and reaction force, a method of attaching virtual masses to the hands and feet of humanoid robots was proposed that improves computational efficiency compared with conventional methods. Dynamic simulations were performed on transportation of various massive objects by two humanoid robots with the proposed method and the conventional PD controller. The results showed that the two humanoid robots could transport a massive object of 200 kg by pushing/pulling without slipping by using the proposed method, while the two humanoid robots could not transport the object by using the conventional PD controller because of the slippage between the feet and the ground.

Keywords: massive object, transportation, teleoperation, humanoid robots, quadratic programming

Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Background

1.1.1 Massive object transportation by robots

Robots have been used to replace human in performing various tasks, and also to perform complicated tasks that can not be done by a human. There are many kinds of robots which can perform human tasks. For instance, cleaning robots perform dirty or dangerous tasks instead of a human [1], [2]. In [3], the cleaning robot is presented which can reconfigure their formation to maximize the cleaning coverage. Cleaning of the building glass is a dangerous task for a human due to the falling by wind or other accidents [4], [5]. [6]. Moreover, industrial robots work instead of a human in factories where there is a shortage of human workers to increase the productivity of factories. Industrial robots applications cover from simple tasks to complex tasks. Generally, manipulators have been used in picking and placing, welding and assembling. There are many kinds of research using manipulators in industrial tasks [7],[8]. An inverse kinematic method for industrial manipulator is proposed in [9]. In [10], autonomous picking and placing task is presented using manipulator with deep learning. In [11], intelligent welding robot programming method is presented using a three-dimensional sensing and point cloud modeling. Dual manipulator performs the assembly tasks by the teaching of human in [12]. In [13], a method of folding assembly for dual manipulators is presented. Wheel robots have been used in transportation. Also, there are many kinds of research using mobile wheel robots in industrial tasks [14], [15], [16]. The multi mobile robots for logistics are autonomously navigated and avoiding the obstacles [17]. Swarm robot control of mobile robots for logisitics are presented in [18].

On the other hand, there are some tasks which are out of human ability, but they can be performed by robots. Subsea operation, surveillance task in the narrow space environment, and massive object transportation are some examples of these tasks that are out of human capabilities. In the subsea operation, a human cannot go to the profound depth of the sea without special vehicles due to the pressure issue. In [19] and [20], the subsea operation is presented using an autonomous remotely operated underwater vehicle (ROV). A well-designed robot, especially for narrow spaces, is appropriate for the surveillance tasks in disaster areas with obstacles. A snake-like robot for surveillance is presented in [21], [22]. In the massive object transportation, transportation cannot be performed by a human due to the heavy weight of the target. The massive object which is too heavy to transport by a human is usually transported by heavy machinery and robots.

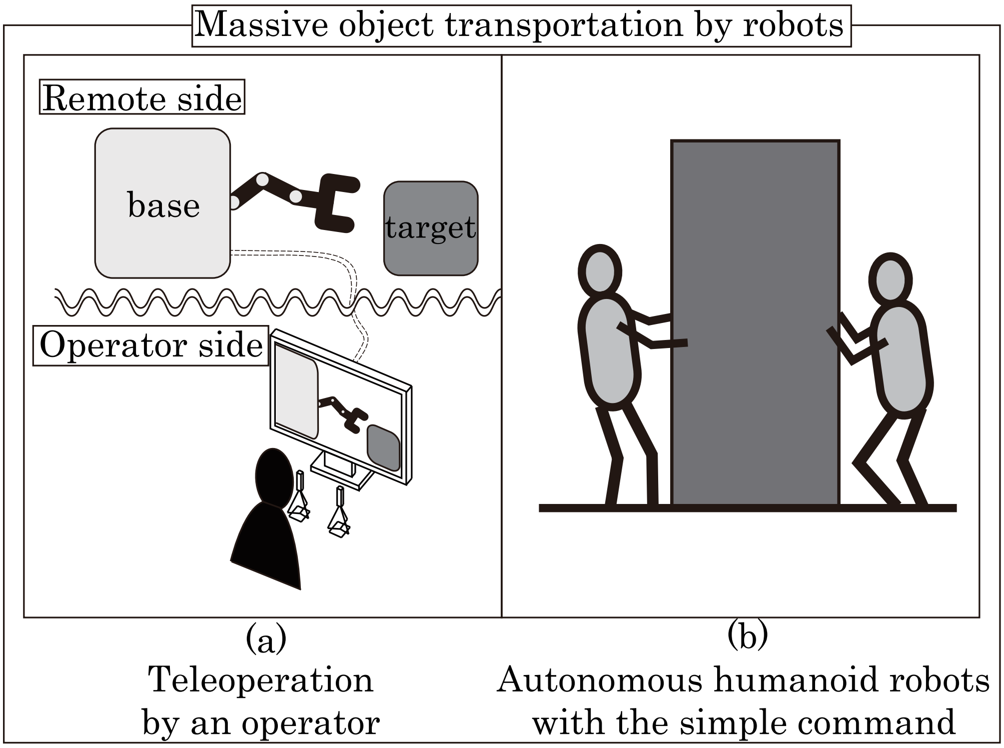


Figure 1.1: Scope of this thesis.

There are many ways to transport a massive object by robots. Teleoperation by an operator and autonomously generating an optimum motion based on a given task can be considered as methods of massive object transportation using robots. In this thesis, two kinds of methods are discussed for massive object transportation by robots. First is the teleoperation by an operator and second is the autonomously generating an optimum motion based on a given task (see Fig. 1.1).

1.1.2 Massive object transportation using manipulator teleoperation by an operator

In the teleoperation by an operator, there are many kinds of researches for the teleoperation of massive object handling using robots. However, these researches are not considered about the long time delay such as from earth to the orbit. In the thesis, massive object transportation using teleoperation by an operator in the long time delay environment. A scenario is assumed that the massive object is transported by using manipulator, and the manipulator is remotely operated by an operator with a proposed teleoperation system in the very long time delay environment.

1.1.3 Massive object transportation using autonomous humanoid robots with the simple command

In the method of the autonomously generating an optimum motion based on a given task, there are many kinds of researches for the massive object transportation using robots. Generally, massive object transportation using these method are performed by automation of heavy machinery and heavy load duty manipulators [23]. However, heavy machines and heavy load duty manipulators are not suitable for transporting massive objects in buildings built for human due to the accessibility issues. There are passages, stairs, ladders, etc. for humans in these buildings. In such a building, it is effective to use humanoid robots due to the accessibility. Therefore, in this thesis, a scenario is assumed that the massive object transportation is performed by humanoid robots in the bulidings built for human. Also, the simple task command such as transport it to the right or left is given to the humanoid robots, and humanoid robots generate appropriate motions autonomously.

1.2 Previous works

In the massive object transportation by robots, teleoperation by an operator and autonomously generating an optimum motion based on a given task can be considered as a method. There are difficulties in each method as mentioned in Section 1. Previous researches about massive object transportation using manipulator teleoperation by an operator and using autonomous humanoid robots with the simple command are presented in this section.

1.2.1 Previous research for massive object transportation using manipulator teleoperation by an operator

Generally, teleoperation using manipulator is performed by master-slave system. The main problems of the massive object teleoperation are the time delay and the unexpected huge force at the robot hand contacts with the massive target. As disccusssed in [24], the time delay between a master system and a slave system decreases the performance of operation.

Directly coupled master-slave teleoperation systems

The master arms and the slave arms are coupled directly in general teleoperation such as [25]. In the case of the large communication delay, operator forced to do "move and wait" strategy. It means that in every operation procedure, operator must wait to see the result of the slave-arm data and continue the operation.

Teleoperation systems using virtual environment

Using a virtual environment, such as model-based teleoperation researches have been performed [26][27][28]. However, an unexpected excessive force generated by a modeling error in the geometric or dynamic properties of the slave arm and the payload could be a problem using a virtual environment, such as model-based teleoperation.

The employment of mixed force and motion commands has been proposed [28][29] to overcome the problem of modeling error in the teleoperation. However, in these teleoperations, the master and slave arms are assumed to have the same sizes. Therefore in massive-payload manipulation, these teleoperation systems cannot be directly used.

Teleoperation systems considered the power scaling

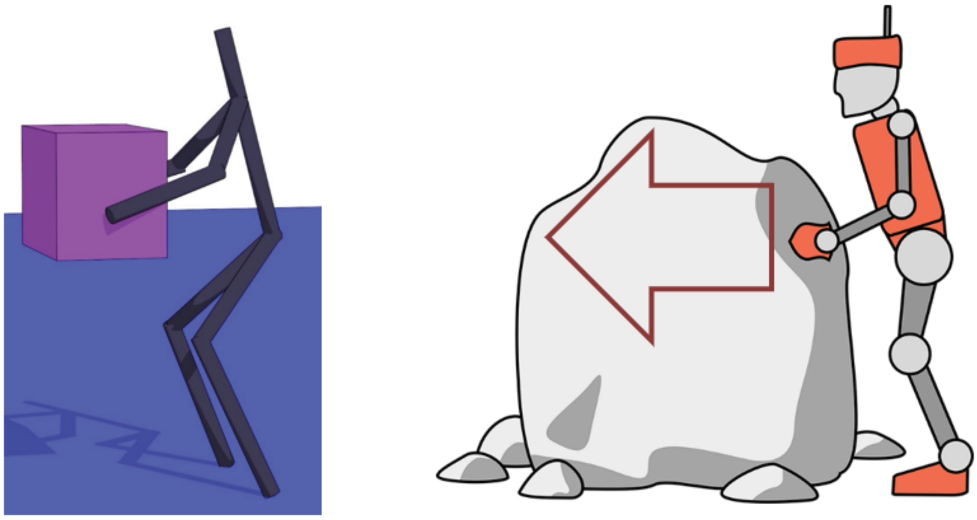
In [30], the writers used a small size master-arm to operate an actual space robot (ETS-VII manipulator). However, the force scaling was not considered, and the contact force was the only feedback to the operator. In [25], the writers considered the power scaling; however, if the communication delay is significant, the method used in [25] may not be useful due to the teleoperation inefficiency.

1.2.2 Previous research for massive object transportation using autonomous humanoid robots with the simple command

There are many kinds of researches about the object transportation using the humanoid robots.

The object transportation by one humanoid robot

In the case of the object which is small and light, the object is transported by lifting (see Fig. 1.2 (a)). In [31], the object is lifted by dynamic motion. A method of the object lifting for humanoid robots is proposed without knowing about the information of object in [32]. In [33], a whole body contact motion is proposed for lifting the object. In [34], lifting motion planning is proposed for CoG (Center of Gravity) and humanoid robot hand trajectories. The balancing motion planning of the humanoid robot is proposed in [35] when lifting the object. However, in these researches, the mass of the transported object is not heavy. In order to overcome these problems, the massive object which is too heavy for lifting is transported by pushing in [36] (see Fig. 1.2 (b)). In [36], optimal posture and contact points were studied. However, the cooperative transportation of the massive object and



(a) Transport by lifting (b) Transport by pushing

Figure 1.2: The object transportation by one humanoid robot.

the pulling task and the optimal pushing motion are not considered in [36]. Also, the non-slip condition is not considered in [36], and it means that the feet of the humanoid robot may slip when pushing the massive object.

The cooperative object transportation by humanoid robots

In the case of the object is too large for a humanoid robot, cooperative transportation of humanoid robots is one of the appropriate approaches (see Fig. 1.3). Therefore, many kinds of research have been performed on cooperation among humanoid robots [37]. A modified transpose Jacobian controller is proposed for object transportation by two humanoid robots in [38]. In [39], a symmetry position/force hybrid controller is proposed for cooperative object transportation using multiple humanoid robots. A cooperative stretcher manipulation with locomotion is presented using small-sized humanoid robots in [40]. A cooperative method for transport through floors and stairs by impedance control of humanoid arm and the position control of humanoid legs are presented in [41]. The human and humanoid cooperative tasks are presented in [42], [43]. [44], [45]. However, in these researches, the object is not heavy and transported by lifting them.

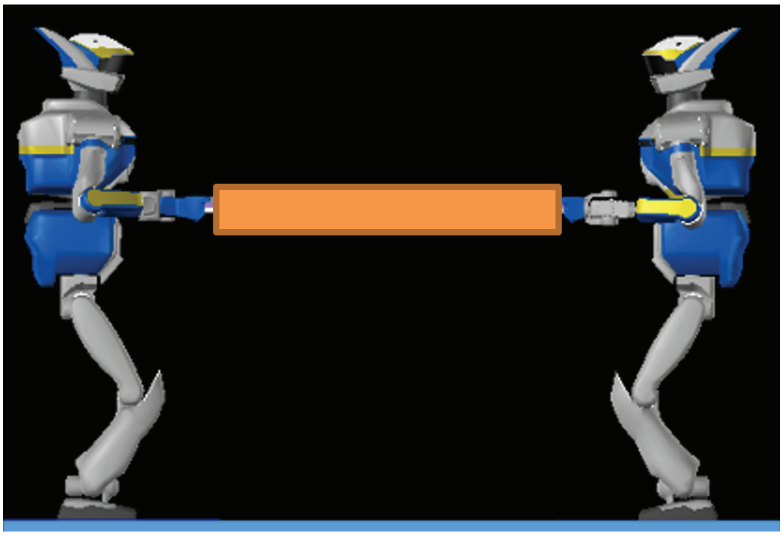


Figure 1.3: The cooperative object transportation by humanoid robots.

The massive object transportation using a cart

Using a cart is a good approach to overcome the output limitation of the humanoid robot when transporting a massive object (see Fig. 1.4). In [46], a human on a wheelchair is transported by humanoid robots. A small humanoid robot transports the massive object using a cart in a cluttered environment with optimizing the trajectory and motions in [47]. However, in order to use the cart, the massive object must be lifted up for the first time. Also, it is difficult to use a cart in the narrow space.

1.3 Research purpose and contribution

1.3.1 Reserch purpose

In this thesis, the massive object transportation by robots are peformed in two methods as stated in Section 1.1. First is the massive object transportation using teleoperation and second is the massive object transportation using two humanoid robots.

In the massive object transportation using teleoperation, we proposed a modified model-based teleoperation method in order to overcome the problem of the

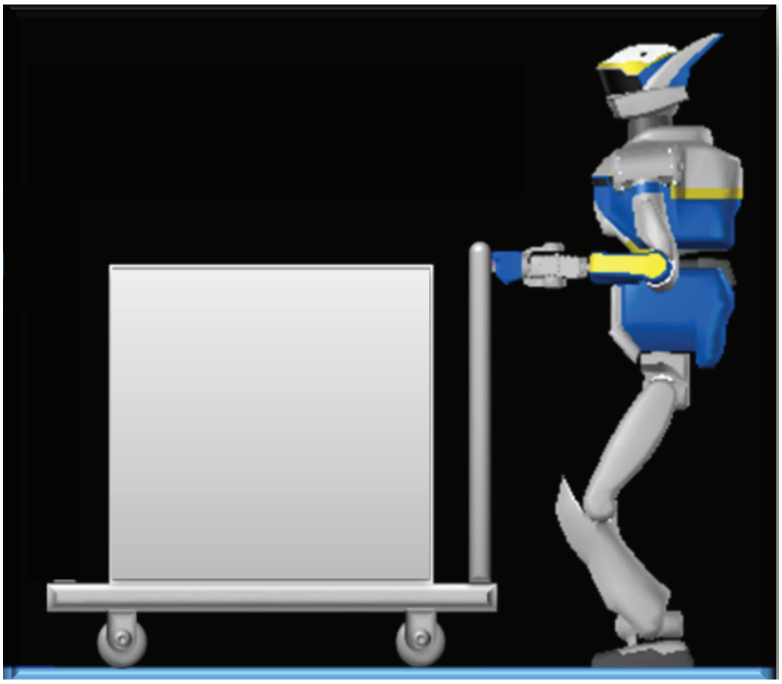


Figure 1.4: The massive object transportation using a cart.

massive object transportation using teleoperation stated in Section 1.2. The proposed model-based teleoperation method is suitable for the massive-payload manipulation considering the size differences between the master and the slave arms.

In the massive object transportation using two humanoid robots, the massive object transportation using two humanoid robots by pushing/pulling without using a cart is proposed in order to overcome the problem of the massive object transportation using humanoid robots stated in Section 1.2.

1.3.2 Contribution

The contributions of this thesis are as follows.

Contribution for the massive object transportation using teleoperation

Contributions for the massive object transportation using teleoperation are force scaling and mitigation of the force generated between the end-effector and the massive payload. In the very long time delay teleoperation of the massive object transportation, the unexpected excessive force due to the operator's scaled-up commands are the key points. Therefore, the mitigation of the force generated between the end-effector and the massive payload are proposed. Also, the virtual environment is proposed, which is considered the force between the end-effector and the massive payload. Therefore, an operator transports the massive payload effectively by watching the proposed virtual environment.

Contribution for the massive object transportation using two humanoid robots

1. Motion generation for pushing/pulling massive objects with two humanoid robots using QP solver

As mentioned in Section 1.2, in the previous researches of the massive object transportation by the humanoid robot, the massive object is transported by lifting them. Therefore, the massive object transportation using two humanoid robots by pushing and pulling is one of the contributions. Also, a realtime QP solver is used to generate the optimized transportation motion of the humanoids in realtime. The realtime QP solver selects the setting variable that minimizes the objective function within the constraints in a short time. Therefore, even in the case of a humanoid robot having many degrees of freedom, it can be optimized autonomously by using the realtime QP solver. The joint torques are minimized by the objective function when transporting the massive object, and the non-slip condition is satisfied as a constraint.

2. Proposal of the prediction method of external force applied to foot using virtual mass and foot acceleration

Generally approximated friction cone method is used as a non-slip constraint of the humanoid robot feet in the QP problems (e.g. [48]). However, this method has disadvantages such as the complexity of the equations and the long calculation speeds as the approximate points increases. In this thesis, a wall pushing situation is assumed to estimate the maximum reaction forces when transporting the massive object. The wall pushing situation is reproduced by attaching huge virtual masses to the end of limbs. The reaction force from the ground to the feet of the humanoid robot is calculated simply by using the virtual mass of the foot and the acceleration of the foot. The calculation speed is about 1.64 times faster than the approximated friction cone method.

1.4 Outline of the thesis

The thesis consists of six chapters.

Chapter 1 is an introduction of this thesis why this subject on massive objects transportation by robots is chosen and researched.

In Chapter 2, as a way to transport the massive object, the method for massive object transportation using teleoperation by an operator is presented. Communication delay, the excessive force generated at the hand of the slave arm due to the massive object, and geometric/dynamic modeling error are critical problems of the massive object transportation using teleoperation. A teleoperation system using mixed force and motion commands is proposed to solve those critical problems. Also, the proposed teleoperation system is verified by hardware-in-the-loop simulations.

In Chapter 3, as a way to transport the massive object, the method for massive object transportation by one humanoid robot is presented. As a previous step for the massive object transportation by two humanoids robots, massive object transportation by a humanoid robot is performed. A scenario is assumed that the target is very heavy for a humanoid robot to lift them. Therefore, the target (massive object) is transported by pushing or pulling. An optimized pushing/pulling motion is generated by the realtime QP solver. The joint torque is minimized by the objective function. The maximum reaction force is estimated by the proposed virtual mass method. The pushing and pulling simulations are performed to verify the proposed motion generation method.

In Chapter 4, as an extension of massive object transportation by one humanoid robot, massive object transportation by two humanoid robots is presented. A scenario is assumed that the target is very heavy for two humanoid robots to lift them. Therefore, the target (massive object) is transported by one humanoid robot is pushing, and the other is pulling simultaneously. The massive object (200 kg) is effectively transported using the proposed method in the dynamic simulation.

Also, the dynamic simulation of the same massive object transportation using a conventional PD controller is performed and compared with the proposed method to show the advantage of the proposed method.

In Chapter 5, the simulation of the intermittent pushing/pulling of the massive object (200 kg) by two humanoid robots is performed to transport the massive object further. In order to transport the massive object further, the stepping/recovering posture motion is generated after the pushing/pulling motion without stepping. Also, the use of proposed method (QP) and the minimization of the joint torque objective function are advantages of this research. In order to show these advantages, the simulations of the massive object transportation with QP and PD and QP without minimize joint torque objective function (QP (w/o JT)) are compared with variation of massive object masses.

Chapter 6 summarizes the thesis and suggests future works.

Chapter 2 Hardware-in-the-loop Simulation of Massive-Payload Manipulation on Orbit

2.1 Introduction

As a way to transport the massive object, the method for massive object transportation using teleoperation by an operator is presented in this chapter.

Space robotic manipulators have been operated in satellite orbit injection and construction of the ISS (International Space Station). SRMS (Shuttle Remote Manipulation System) and SSRMS (Space Station Remote Manipulator System) are two examples of these manipulators which have been used by astronauts onsite.

However, for both safety and financial reasons, there would no need to send astronauts to space if such manipulation could be performed by teleoperating the space robot from the earth.

Generally, satellites or modules of space structures have huge mass compared to the robot arm. During the manipulation, excessive force might be generated between the end effector of the robot arm and the massive payload, especially at the start and end of the movement. Therefore, Astronauts must be very careful when operating the robot. For teleoperating the robot arm from the earth during a manipulation task, the communication delay and the unexpected excessive force generated between the space robot and the massive payload are the two key issues.

The communication delay between a master arm and a slave arm is known to dramatically decreases the performance of the teleoperation [24]. The master arms and the slave arms are coupled directly in general teleoperation (e.g. [25]). In the case of the large communication delay, operator forced to do “move and wait” strategy. It means that in every operation procedure, operator must wait to see the result of the slave-arm data and continue the operation. Under communication delay, in order to perform teleoperation efficiently, semi-autonomy could be a solution [49]. Using a virtual environment, such as model-based teleoperation researches have been performed [26][27][28]. However, an unexpected excessive force generated by a modeling error in the geometric or dynamic properties of the slave arm and the payload could be a problem using a virtual environment, such as model-based teleoperation.

The employment of mixed force and motion commands has been proposed [28][29] to overcome the problem of modeling error in the teleoperation. However, in the model-based teleoperation, the master and slave arms are assumed to have the same sizes, and therefore in massive-payload manipulation, this teleoperation system cannot be directly used. In [30], the writers used a small size master-arm to operate an actual space robot (ETS-VII manipulator). However in their work, force scaling was not considered, and the contact force was the only feedback to the operator. In [25], the writers considered the power scaling; however, if the communication delay is significant, the method used in [25] may not be useful due to the teleoperation inefficiency, as mentioned above. In this chapter, we modify the model-based teleoperation for application to massive-payload manipulation considering the size differences between the master and the slave arms.

Moreover, a hardware-in-the-loop simulation (HILS) is employed to validate the modified model-based teleoperation. In [50], several HIL simulators for space robots were surveyed. In order to simulate the capture of space targets (for example, satellites), HIL simulators for space robots have been developed [52][53][54]. However, to reproduce the space target motion, conventional serial robots are used in most HIL simulators for space robots. Generally, since serial robots do not move very quickly, the response delay is relatively large. HILS instability may be caused due to the large response delay of the motion table (the robot arm used to reproduce the space target motion) [55]. In this chapter, in order to validate the model-based teleoperation by simulating massive-payload manipulation, delay time compensation [55] is applied to a parallel-robot-based HIL simulator [56].

The proposal for a modified version of model-based teleoperation using mixed force and motion commands and the parallel-robot-based HILS of the massive payload manipulation are the main contributions in this chapter. In the proposed method, there are force scale-up for the command of the operator. Therefore, the mitigation of the excessive force at the remote site due to the operator ’ s scaled-up commands in a serious communication delay would be the key point. By performing HIL simulations, the proposed method which mitigate the generated force at the remote side was verified.

2.2 Hardware-in-the-loop (HIL) space robot simulator

2.2.1 Overview of space robot teleoperation system

In Fig. 2.1, a space robot teleoperation system is shown. In the space center on earth, the master arms are manipulated by an operator. This task is performed by observing the virtual environment and predicting the movement of the space robot on orbit. A communication delay occurs during the transformation of the control inputs from the master arms and the position data of the virtual arm to the slave arm on orbit. When the slave arms contact with the payload or the environment, the slave arms on orbit follow the motion commands generated by the master arms on earth, and the contact force is locally controlled using force sensor data. However, due to the communication delay, the slave arms' position data arrive at the space center on earth with a lag. Hence, at the moment of teleoperation, the positions of the slave arms are unknown on earth. Simulated motions of the slave arms (virtual arms) are displayed for the human operator to overcome the communication delay problem. As well as the motion command generated by the master arms, the simulated motion is also transferred to the slave arm controller (see Fig. 2.1). In order to reduce the error between the slave arm and the virtual arm, the position of the slave arm is modified by the motion of the virtual arms. In order to display the movement of the slave arm on the virtual environment, which is superimposed on the movement of the virtual arm, the joint angle data of the slave arm are transferred to the virtual environment. The system can be divided into the slave system, the master system, and the hybrid motion simulator.

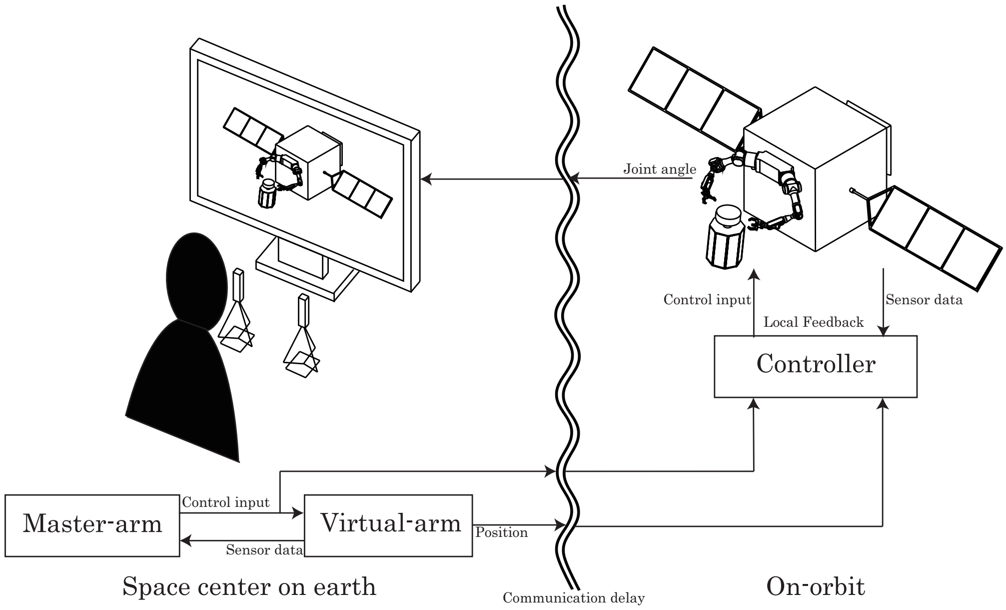


Figure 2.1: A space robot teleoperation system.

2.2.2 HIL simulator

The HIL simulator [56] is composed of three main parts (6 axis force sensor, X-Y motion table, HEXA-type motion table). For acquiring the external force, it consists of a 6 axis force sensor, for reproducing the large translational movement of the payload an X-Y motion table is employed, and for reproducing 6-DOF guide movement, a HEXA-type [57] motion table is utilized. At the boundary of the hardware and the software simulation, the force sensor is mounted. A force and torque acting on the payload are measured by the force/torque (F/T) sensor mounted in the HIL simulation. By solving a dynamic equation with the measured force/torque data, the relative position and orientation are calculated concerning the payload of interest. Eventually, the calculated relative position and orientation are recognized on the hardware side by a servomechanism in real time (see Fig. 2.2). In [55], the response delay time compensation method is employed to the hybrid simulator.

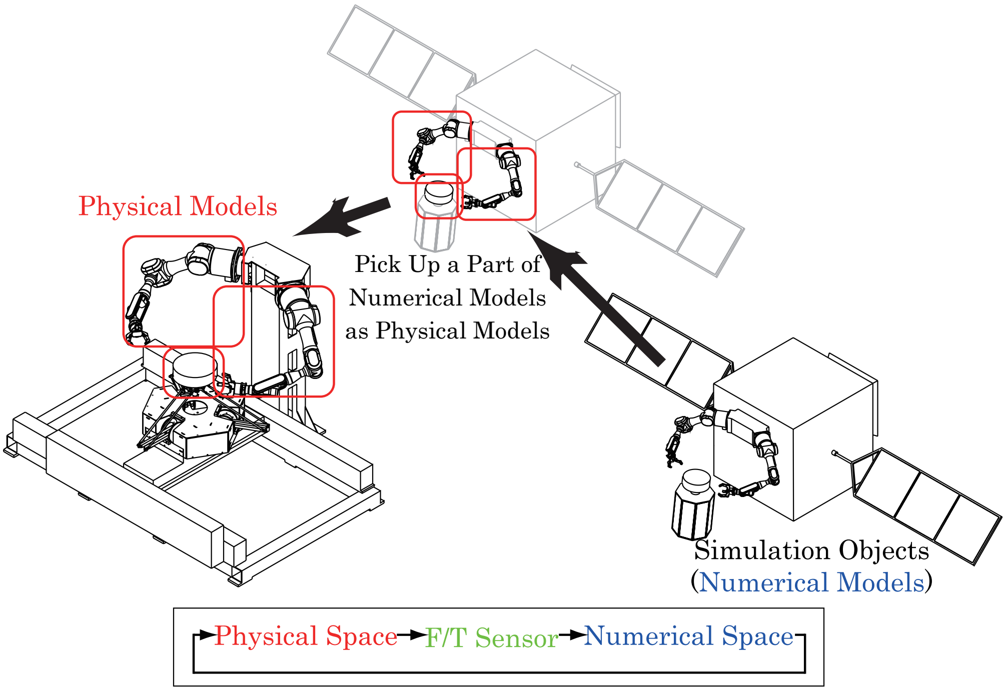


Figure 2.2: The hybrid simulation.

2.2.3 Slave system

Two 7-DOF manipulators with a serial link mechanism and a camera head system for monitoring the working environment form the slave system (see Fig. 2.3). By panning and tilting, the camera head can detect the workspace. For grasping the various payloads, a wrist camera, a 6-axis force sensor and a BarrettHand™ (Barrett Technology [58], LLC) are installed on each end effector. The force generated at the wrist of the slave arm is acquired by The 6-axis force sensor. To grip a payload, the BarrettHand can open and close fingers. To obtain detailed information about the environment with two different fields of view angle, the end effector camera is utilized in addition to the camera head.

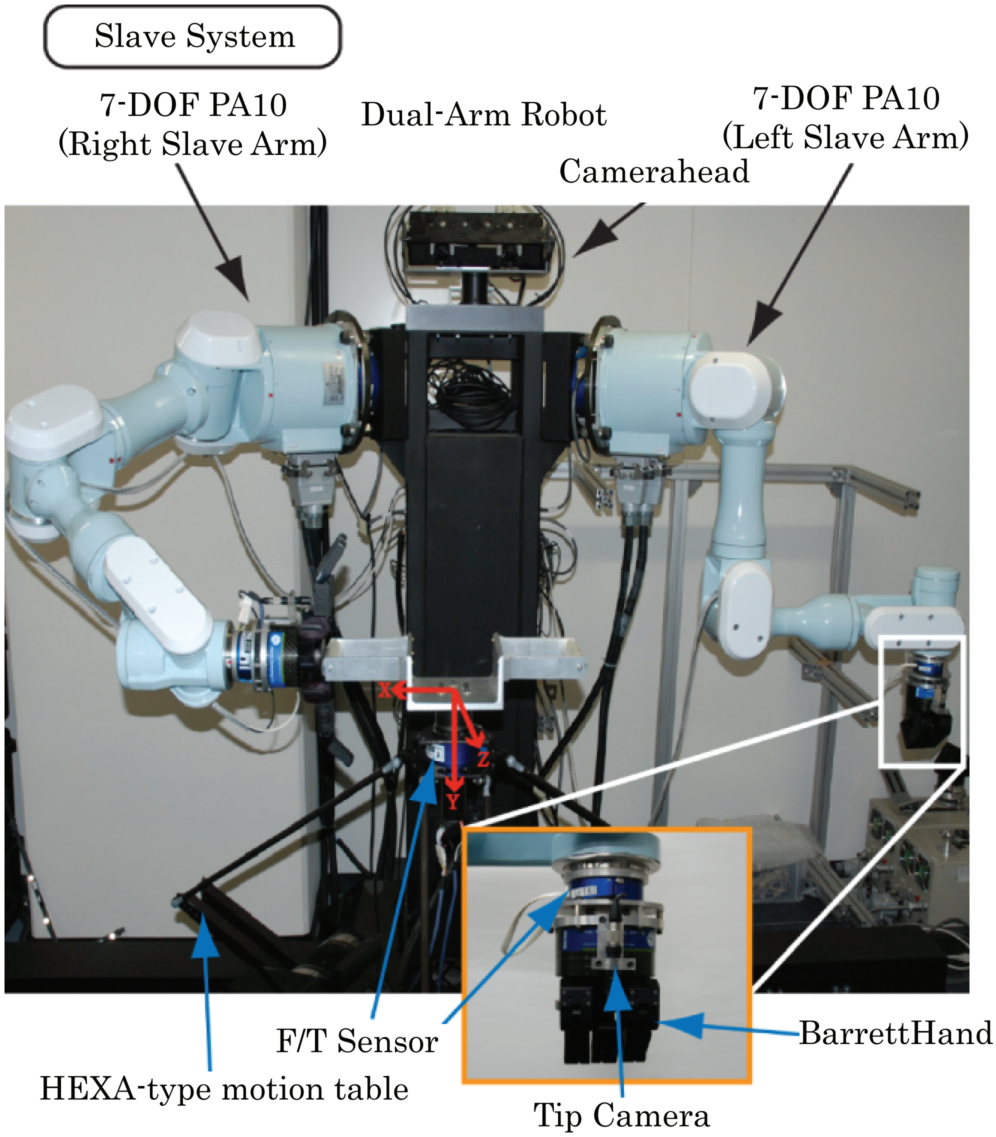


Figure 2.3: The slave system.

2.2.4 Master system

Two 6-DOF haptic interfaces with a parallel link mechanism compose the master system (see Fig. 2.4) [59]. A NANO 5/4-A (Bl Autotec) force sensor is used in the 6-DOF haptic interface. In this chapter, the 6-DOF haptic interface is used in the joystick mode (the haptic device does not move and does not display forces in the joystick mode). Finally, the virtual environment is represented on a monitor.

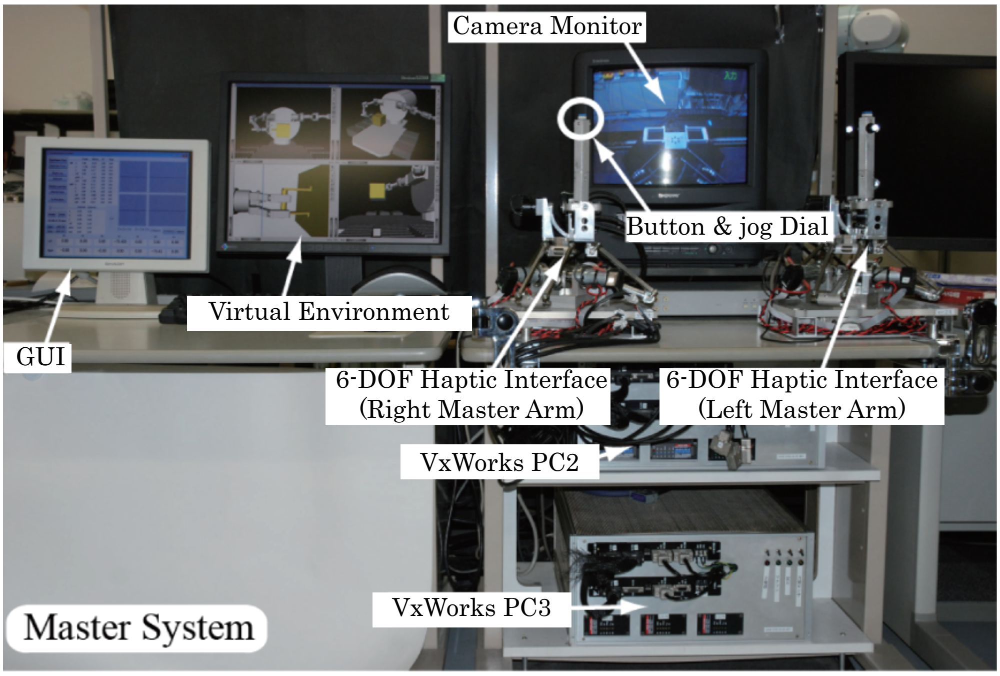


Figure 2.4: The master system.

2.3 Virtual environment and control law development

2.3.1 Operation scenario

Conveying a massive payload by teleoperating a space robot is assumed as the given task. A mounted grip on the HEXA-type motion table (Fig. 2.3) is supposed to be a part of the massive payload. An F/T sensor mounted between the grip and the HEXA-type motion table measures the force applied to the grip. The measured force and torque are regarded as the external force applied to the massive virtual payload. By solving a dynamic equation, the massive payload motion is calculated, and the motion is generated by the HEXA-type motion table mounted on the X-Y table.

In the operation scenario, to manipulate a massive payload, a human is assumed to teleoperate a space robot arm (slave arm) from the earth.

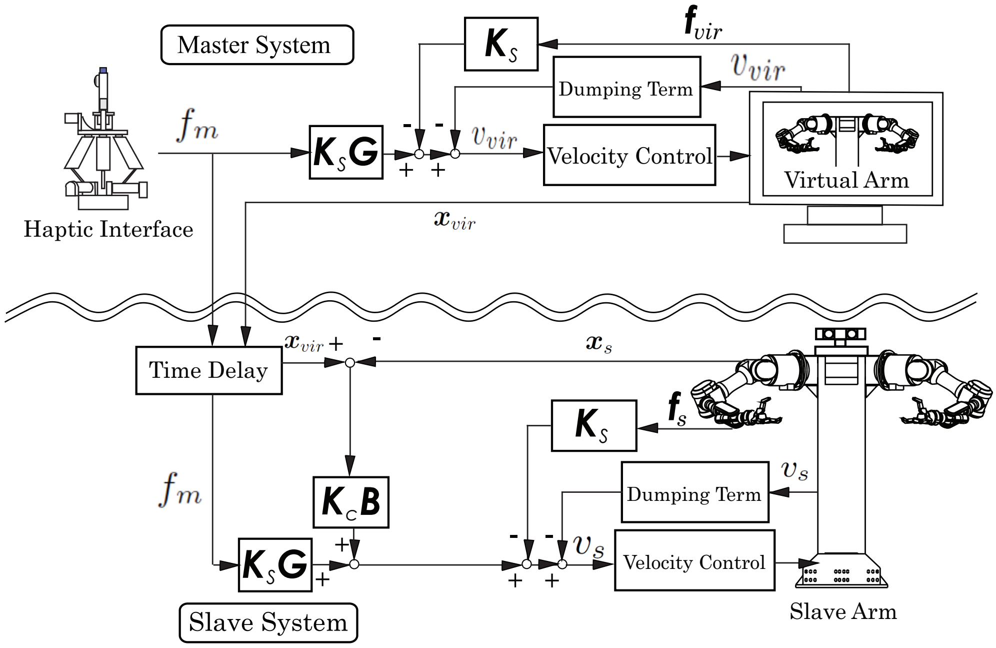


Figure 2.5: System block diagram.

2.3.2 Control law

In Fig. 2.4, the haptic device is utilized as a master arm in the joystick mode. An F/T sensor measures the force applied by the human operator to the stylus of the master arm, and then the measured data is transformed into a velocity command of the slave arm. In Fig. 2.5, a block diagram of the teleoperation system is demonstrated.

The $\boldsymbol{x}_i = [\boldsymbol{r}_i^T \quad \boldsymbol{\phi}_i^T]^T$ and $\boldsymbol{v}_i = [\dot{\boldsymbol{r}}_i^T \quad \boldsymbol{\omega}_i^T]^T$ ($i = m$, vir, s, or t) are the six dimensional position and velocity vectors, respectively. Subscripts m, vir, s, and t, represent the master arm, virtual arm, slave arm, and massive payload, respectively. $\boldsymbol{\phi}_i$ and $\boldsymbol{\omega}_i$ implicate a set of three variables for orientation (e.g., Euler angles) and the angular velocity vector, respectively. The derivative of $\boldsymbol{x}_i$ is transformed into $\boldsymbol{v}_i$ by the matrix $\boldsymbol{B}(\boldsymbol{\phi}_i)$ as $\boldsymbol{v}_i = \boldsymbol{B}(\boldsymbol{\phi}_i)\dot{\boldsymbol{x}}_i$.

Geometric and dynamic modeling errors are two severe problems of the teleoperation with a communication delay. Since the operator on earth would not be able to monitor the remote site in real time, if substantial discrepancies occur between the real environments at the remote site and their models, it may not be possible to catch/contact a payload with the slave arm via teleoperation. An unexpected excessive force can be generated between the slave arm and the payload due to the significant discrepancies between the dynamic properties of the slave arm/payload and those of the models. In [29, 28], model-based teleoperation using mixed force and motion commands has been proposed to overcome those problems. In this chapter, the model-based teleoperation [29, 28] is modified to manipulate a massive payload. The scaling modification between the master arm and the slave arm is the main contribution of this chapter. In the original model-based teleoperation using mixed force and motion commands [29, 28], the force and motion in both arms (master and slave arms) were assumed to be of the same scale. However, to manipulate massive payload by a large-sized slave arm in the manipulation task, we need to scale up the force/motion command given by a human operator. In the following sections, we present the modified model-based teleoperation using mixed force and motion commands.

Control law of slave arm

The velocity command to the slave arm,$\boldsymbol{v}_{s,c}(t)$, is given as follows:

$$\boldsymbol{v}_{s,r}(t) = \boldsymbol{K}_s \boldsymbol{G} \boldsymbol{f}_m(t), \tag{2.1}$$

$$\begin{aligned}\boldsymbol{v}_{s,c}(t) = \boldsymbol{v}_{s,r}(t - t_{delay}) - \boldsymbol{K}_s \boldsymbol{f}_s(t) + \boldsymbol{K}_c \boldsymbol{B}(\boldsymbol{x}_{vir}(t - t_{delay}) - \boldsymbol{x}_s(t)) \\ -\boldsymbol{D}(\boldsymbol{v}_s(t - \Delta t) - \boldsymbol{v}_s(t - 2\Delta t)),\end{aligned} \tag{2.2}$$

where $\boldsymbol{f}_m$ is the reference force commanded by the human operator using the master arm, $\boldsymbol{f}_s$ is the actual force generated at the slave arm, $\boldsymbol{G}$ is a diagonal scaling matrix between the force commanded by the master arm and the force the slave arm is expected to generate. The force command $\boldsymbol{f}_m$ given by the human operator must be scaled up for the slave arm to enable the slave arm to manipulate the massive payload. The matrix $\boldsymbol{G}$ scales up the reference force. Substituting (2.1) into (2.2), the following equation is reached:

$$\begin{aligned}\boldsymbol{v}_{s,c}(t) = \boldsymbol{K}_s(\boldsymbol{G}\boldsymbol{f}_m(t - t_{delay}) - \boldsymbol{f}_s(t)) + \boldsymbol{K}_c \boldsymbol{B}(\boldsymbol{x}_{vir}(t - t_{delay}) - \boldsymbol{x}_s(t)) \\ -\boldsymbol{D}(\boldsymbol{v}_s(t - \Delta t) - \boldsymbol{v}_s(t - 2\Delta t)),\end{aligned} \tag{2.3}$$

$\boldsymbol{v}_{s,r}$ is the reference velocity given by (2.1). $\boldsymbol{x}_{vir}$ is the position vector of the slave arm. They are transferred from the earth and arrive at the space robot t_{delay} later. Hence, $\boldsymbol{f}_m$ and $\boldsymbol{x}_{vir}$ used in (2.3) are t_{delay} ahead of the local (space) time. $\boldsymbol{x}_s$ is the position vector of the slave arm, $\boldsymbol{K}_c$ is a gain matrix for the restraint force, $\boldsymbol{f}_s$ is a force vector measured by the F/T sensor of the slave arm, and $\boldsymbol{K}_s$ is a diagonal force feedback gain matrix. $\boldsymbol{D}$ is a diagonal damping gain matrix. Δt is the sampling period of the controller. The original model-based teleoperation using mixed force and motion commands [29, 28] shows some unstabilities in HILS. To solve this, the damping term (the last term of the right-hand side of (2.2)) is added in our method.

Virtual arm

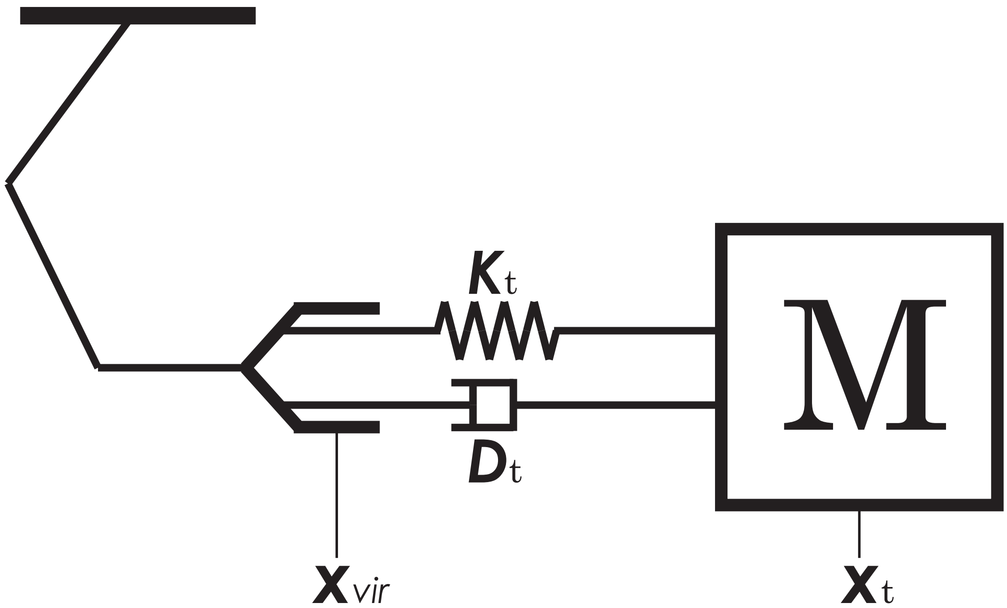


Figure 2.6: Virtual force.

In order to overcome communication delay (e.g., [28]), or unexpected happenings (e.g., [60, 61]), model-based virtual robots between the real robot and the master arm have been proposed. In order to compensate for the modeling error in dynamic properties as well as to overcome the communication delay, the virtual arm is implemented. As shown in the second term of the right-hand side of (2.2), the difference in the positions between the virtual slave arm and the real slave arm is feedback to the real slave arm control. This term helps decrease the positioning error made by the modeling error in the dynamic properties of the slave arm. The reference velocity of the slave arm is also transferred to the virtual arm as follows:

$$\boldsymbol{v}_{vir,r}(t) = \boldsymbol{v}_{s,r}(t) = \boldsymbol{K}_s \boldsymbol{G} \boldsymbol{f}_m. \tag{2.4}$$

Similar to the velocity command to the slave arm (2.2), the velocity command to the virtual arm is given as follows

$$\boldsymbol{v}_{vir,c}(t) = \boldsymbol{v}_{vir,r}(t) - \boldsymbol{K}_s \boldsymbol{f}_{vir}(t) - \boldsymbol{D}(\boldsymbol{v}_{vir}(t - \Delta t) - \boldsymbol{v}_{vir}(t - 2\Delta t)), \tag{2.5}$$

$$\boldsymbol{x}_{vir}(t) = \boldsymbol{x}_{vir}(t - \Delta t) + \boldsymbol{B}^{-1} \boldsymbol{v}_{vir}(t) \Delta t, \tag{2.6}$$

where $\boldsymbol{v}_{vir}$ is the velocity of the virtual arm. The matrix $\boldsymbol{B}^{-1}$ transforms $\boldsymbol{v}_{vir}$ into $\dot{\boldsymbol{x}}_{vir}$. In order to predict the movement of the slave arm and present it in the virtual environment, the virtual arm control law is almost the same as the slave arm control law. When high-ratio reduction gears are employed at the joints of the the slave arm, $\boldsymbol{v}_s$ can be approximated by $\boldsymbol{v}_{s,c}$ (see Appendix **??**).

The contact force between the massive payload and the end effector of the slave arm in the virtual environment is simulated by a spring-damper system, as demonstrated in Fig. 2.6. $\boldsymbol{f}_{vir}$ is simulated as follows:

$$\boldsymbol{f}_{vir}(t) = -\boldsymbol{K}_t \boldsymbol{B}(\boldsymbol{x}_{vir}(t - \Delta t) - \boldsymbol{x}_t(t - \Delta t)) - \boldsymbol{D}_t(\boldsymbol{v}_{vir}(t - \Delta t) - \boldsymbol{v}_t(t - \Delta t)), \tag{2.7}$$

where $\boldsymbol{K}_t$ is a spring constant matrix, $\boldsymbol{D}_t$ is a viscosity coefficient matrix, $\boldsymbol{x}_t$ is a position vector of the virtual massive payload, and $\boldsymbol{v}_t$ is a velocity vector of the payload. The massive payload acceleration is represented by

$$\dot{\boldsymbol{v}}_t(t) = \boldsymbol{M}^{-1} \boldsymbol{f}_{vir}(t), \quad \boldsymbol{M} = \begin{bmatrix} m\boldsymbol{E} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \tag{2.8}$$

where m is the mass, $\boldsymbol{E} \in \Re^{3\times3}$ is an identity matrix, and $\boldsymbol{I} \in \Re^{3\times3}$ is the massive payload inertia tensor. A satellite or a module of a space structure can be the massive payload, and hence m and $\boldsymbol{I}$ are known (they can be obtained from the design parameter). However, a modeling error might be included in the inertia matrix $\boldsymbol{M}$. Using (2.8)–(2.10) the position of the payload is estimated, we can infer that an estimation error is caused by the modeling error in $\boldsymbol{M}$ in (2.8). The coupling term in the slave arm controller compensates for the position error caused by the modeling error (the third term of the right-hand side of (2.2)).

The position and velocity of the massive payload are calculated by

$$\boldsymbol{v}_t(t) = \boldsymbol{v}_t(t - \Delta t) + \dot{\boldsymbol{v}}_t(t)\Delta t, \tag{2.9}$$

$$\boldsymbol{x}_t(t) = \boldsymbol{x}_t(t - \Delta t) + \boldsymbol{B}^{-1}\boldsymbol{v}_t(t)\Delta t. \tag{2.10}$$

2.3.3 Virtual environment

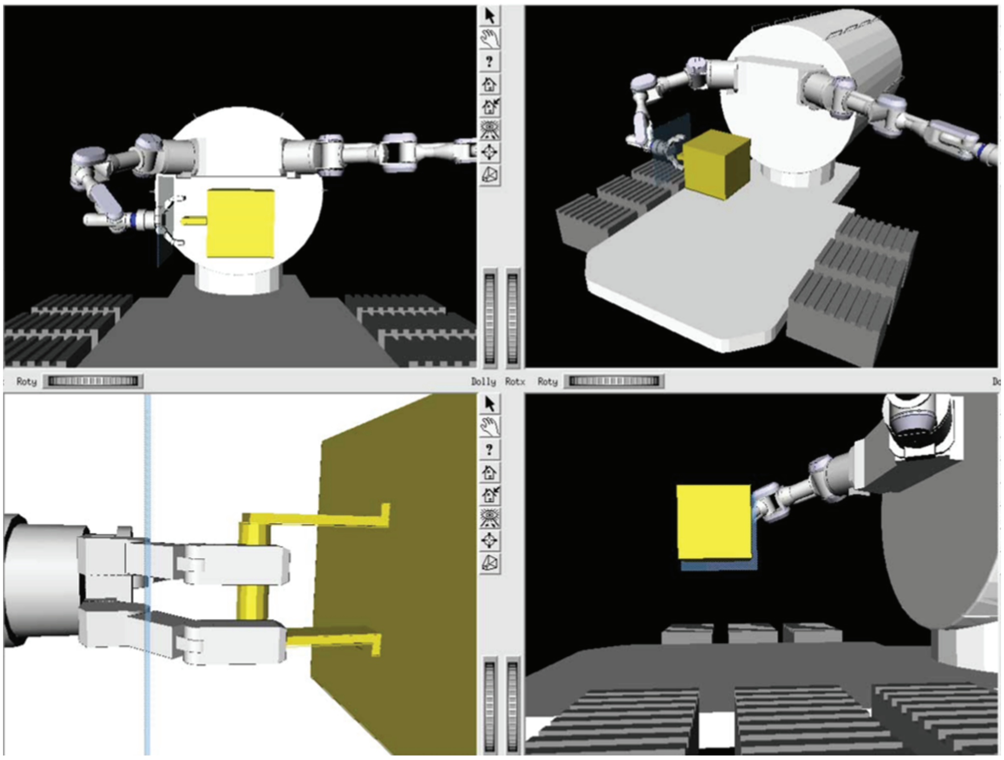


Figure 2.7: Virtual environment.

The virtual environment was made for the operator by computer graphics to show the movement of the target, slave arm, and virtual arm. By assuming high reduction gear ratios at the joints, the velocity of the virtual arm, $\boldsymbol{v}_{vir}$, was approximated (As discussed in Section 3.2.3). In the virtual environment, the dynamics of the massive payload was only considered (2.7)–(2.10). For computing the force

Table 2.1: Payload mass and inertia

Mass [kg]	Inertia [kgm^2]
500	diag [7.47 7.60 7.61]
3000	diag [44.82 45.61 45.64]

Table 2.2: Gain parameters of the experiment

$\boldsymbol{K}_v$	diag [80/1000 80/1000 80/1000 0 0 0]
$\boldsymbol{K}_s$	diag [1/1400 1/1400 1/1400 0 0 0]
$\boldsymbol{D}$	diag [0.3 0.3 0.3 0 0 0]
$\boldsymbol{K}_t$	diag [746000 746000 746000 0 0 0]
$\boldsymbol{D}_t$	diag [1000 1000 1000 0 0 0]

feedback term of the velocity command (2.5), the dynamics of the massive payload is used ((2.7)–(2.10)). The video frame rate is 33 ms, which is much bigger compared to the virtual environment calculation time (1 ms/loop). In Fig. 2.7, the virtual arm, and the massive payload are displayed on a screen from the four different viewpoints. On the computer graphics, it is also possible to zoom the camera. A solid model is used to display the motions of the virtual arms. By superimposing the joint data transferred from space with a wireframe model on the motion of the virtual arms, the slave arms motions are created. This can be used to check whether the slave arm follows the virtual arm after a time delay.

2.4 Verification experiments

2.4.1 HIL simulation validity

In order to confirm the validity of the system, HIL simulations were performed. The slave arm grasps a massive payload of 3,000 kg and pulls it along the x-axis at a speed of 0.01 m/s for 20 s, in the HIL simulation. In Table 2.1, the experimental assumption of the mass, and inertia of the virtual payload are presented.

Instead of human teleoperation, the step input of the reference velocity of $\boldsymbol{v}_{s,r}$

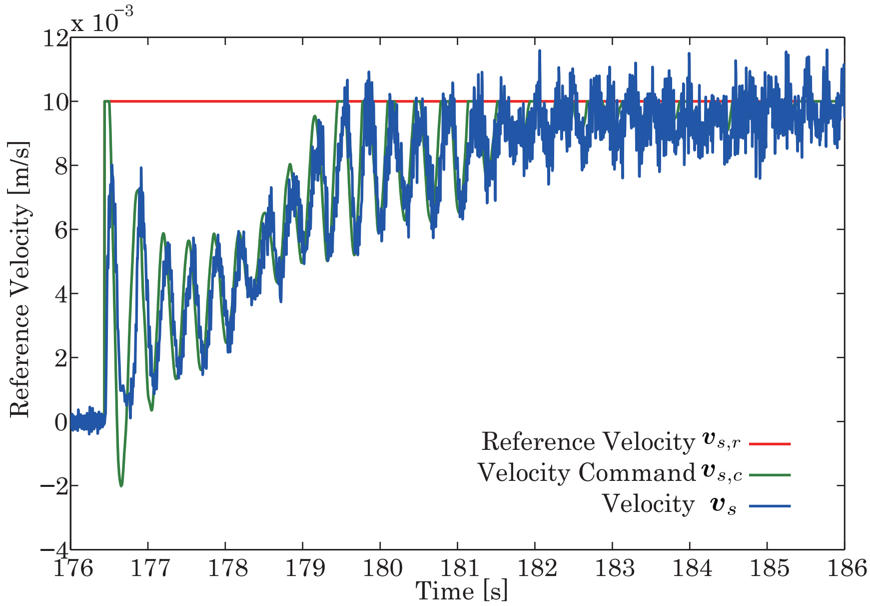


Figure 2.8: The $\boldsymbol{v}_{s,c}$ and $\boldsymbol{v}_{s,r}$(x-axis) comparison.

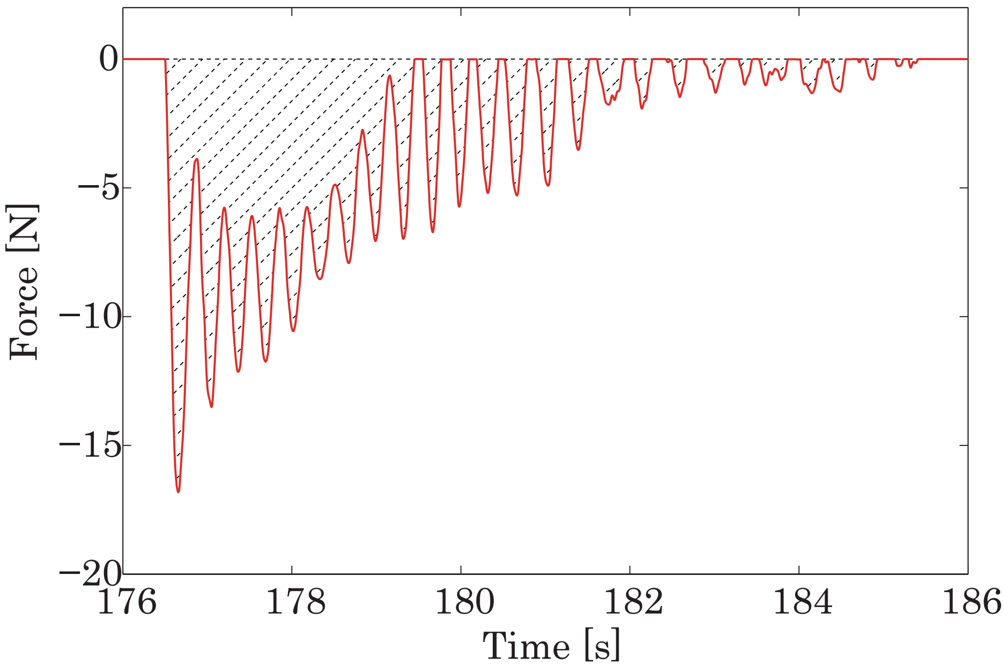


Figure 2.9: Force of slave arm(x-axis).

= 0.01 m/s was given in the HIL simulation. Using (2.2) with $\boldsymbol{v}_{s,r}$ = 0.01 m/s, the velocity command $\boldsymbol{v}_{s,c}$ is calculated. The plot of the $\boldsymbol{v}_{s,r}$, $\boldsymbol{v}_{s,c}$, and $\boldsymbol{v}_{s}$ during the HIL simulation are demonstrated in Fig. 2.8. By employing force feedback, the end effector velocity $\boldsymbol{v}_s$ reduces and asymptotically converges to $\boldsymbol{v}_{s,r}$ (0.01 m/s). The measured forces by the wrist force sensor of the slave arm are shown in Fig 2.9. A momentum of 30 Ns is necessary to produce the acceleration and increase the velocity of the massive payload velocity into 0.01 m/s. A step reference velocity $\boldsymbol{v}_{s,r} = 0.01$ m/s was given at t = 176.5 s and $\boldsymbol{v}_s$ almost converged to $\boldsymbol{v}_{s,r}$ at t = 186 s. $\boldsymbol{f}_s$ plotted in Fig. 2.9 was integrated from 176.5 s to 186 s, and the integral value was 28.43 Ns. The velocity of the massive payload is assumed to be almost the same as the velocity of the slave arm. Therefore, the massive payload movement in the HIL simulation almost agrees with the theory. There is an approximately 5% error in the momentum required to accelerate the massive payload from 0 to 0.01 m/s. The applied force is adequately mitigated by the force feedback term of the control law of the slave arm, and the velocity of the slave arm asymptotically converged to the given reference velocity.

2.4.2 The results comparison of with and without force feedback

In order to verify the effectiveness of the force feedback in (2.2) in reducing the resultant force during the massive-payload manipulation mission, another HIL simulation was performed. A huge force will be generated between the payload and the end effector of the slave arm when a virtual payload of 3,000 kg is manipulated in a HIL simulation without force feedback. Hence, there is a risk that the slave arm used in this chapter would get broken. Therefore, instead of a 3,000 kg payload, a payload of 500 kg is used as the massive payload in the HIL simulation.

The slave arm grasped and pulled the massive payload of 500 kg for 1 min at a speed of 0.01 m/s along the x-axis with and without force feedback. The force generated between the slave arm and the payload in an automatic manipulation experiment for the payload of 500 kg without force feedback is demonstrated in Fig. 2.10. As shown in Fig. 2.10, a maximum force of 23 N was generated when the payload was grasped and pulled during the non-feedback experiment. Lastly, the arm stopped pulling (see Fig. 2.10). The automatic manipulation experiment of the 500 kg payload with the force feedback is demonstrated in Fig. 2.11. As

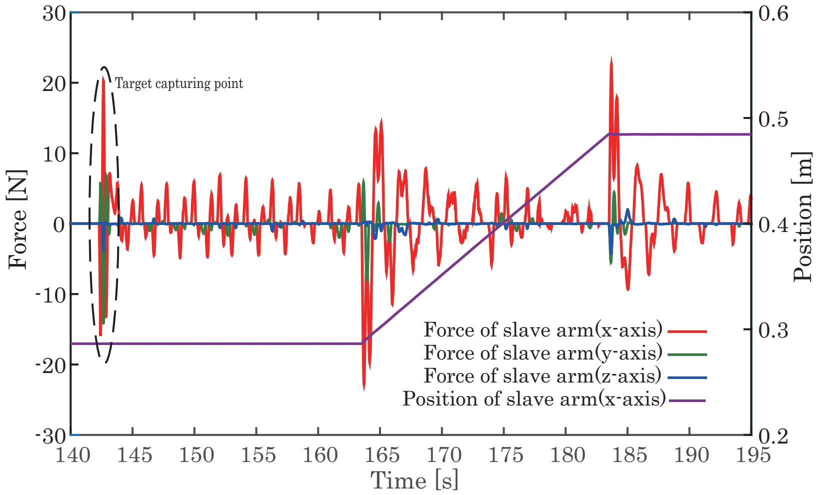


Figure 2.10: Force and position of slave arm without force feedback.

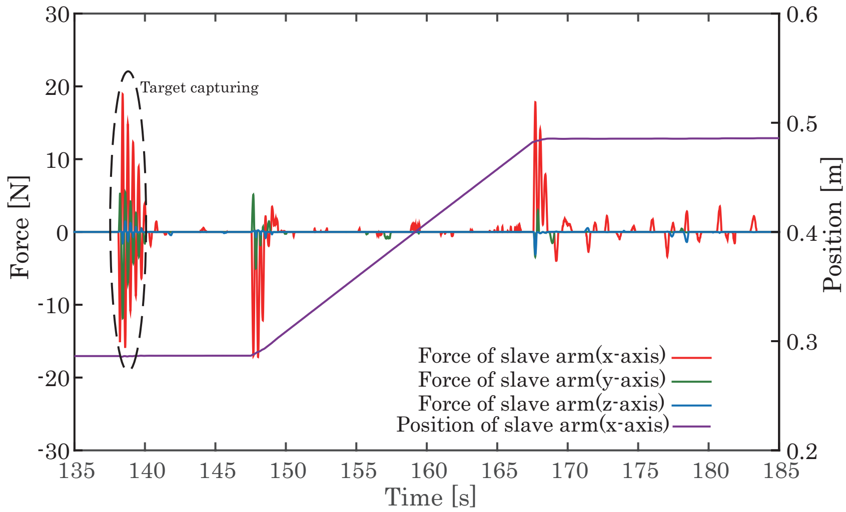


Figure 2.11: Force and position of slave arm with force feedback.

shown in Fig. 2.11, a maximum force of 18 N was generated when the payload was grasped and pulled during the experiment with force feedback. Lastly, the arm stopped pulling (see Fig. 2.11). The generated force was kept small during the manipulation task. In order to compare the results of the experiments with and without force feedback, the mean square error of $\boldsymbol{f}_s$ is calculated as follows:

$$MSE(\boldsymbol{f}_s) = \frac{1}{N}\sum_{t=0}^{\infty} \|\boldsymbol{f}_s\|^2, \tag{2.11}$$

where N indicates the number of data items. When the force feedback was applied and was not applied, $MSE(\boldsymbol{f}_s)$ was 5.41 N^2 and 15.35 N^2, respectively. Thus, it can be concluded that in order to reduce the force generated at the slave arm during the massive payload manipulation, the force feedback in the slave arm controller is beneficial.

2.5 HIL simulation of massive-payload manipulation

2.5.1 Scenario of experiments

The HIL simulations were performed in two types (HILS 1 and HILS 2).

First, in the HILS 1, the slave arm grasped and transferred a 4,500 kg virtual massive payload along the x-axis at a constant speed of 0.01 m/s for 20 s. Next, in the HILS 2, the slave arm grasped and transferred a 4,500 kg virtual massive payload along the x-axis by human teleoperation with a communication delay of 5 s. The input values for the y- and z-axes components are assumed to be zero in the teleoperation experiment.

In [28], the authors verified the effectiveness of the model-based teleoperation using the mixed force and motion commands for the communication delay problem by performing a teleoperation experiment between German Aerospace Center (DLR, Germany) and Tohoku University (Japan). In this chapter, we verified the effectiveness of the modified version of the model-based teleoperation for massive-payload manipulation.

The result of the virtual payload manipulation of 4,500 kg simulation is presented in this chapter, although two similar HILSs were carried out with virtual payloads of 3,000 kg and 6,000 kg. In Table 2.3, the mass, and inertia used in the HILS are demonstrated. The same gain parameters used in the experiments presented in Section 2.4 were employed in this experiment.

Table 2.3: Mass and inertia parameters of the payload (4500 kg)

Mass [kg]	Inertia [kgm^2]
4500	diag [67.22 68.42 68.46]

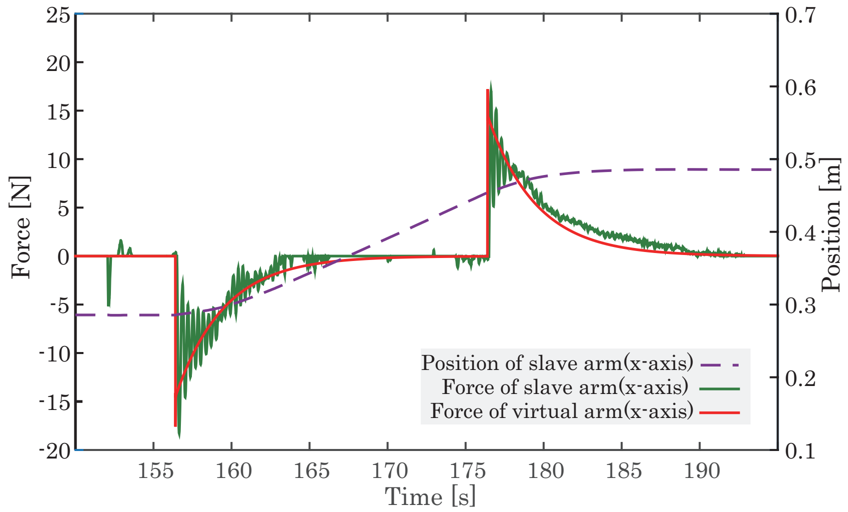


Figure 2.12: Slave arm and virtual arm forces (x-axis) (in HILS 1).

2.5.2 The end effector force comparison between the slave arm and the virtual arm (HILS 1)

The end effector force of the slave arm generated along the x-axis is compared with that of the virtual arm (see Fig. 2.12) to evaluate the accuracy of the simple virtual force model demonstrated in Fig. 2.6.

Although the simple spring-damper model did not reproduce the oscillation of the force, this model reproduced the mean value of the force with adequate accuracy. it can be concluded that by employing the simple spring-damper virtual force model, the dynamics of the massive payload of the slave side was modeled suitably.

2.5.3 The end effector position comparison between the slave arm and the virtual arm (HILS 1)

In Fig. 2.13, the positions of the end effectors of the slave and virtual arm along the x-axis are demonstrated. In Fig. 2.13, since the positions of both arms along the y- and z-axes were almost constant and identical, the positions of arms along the x-axis are only demonstrated.

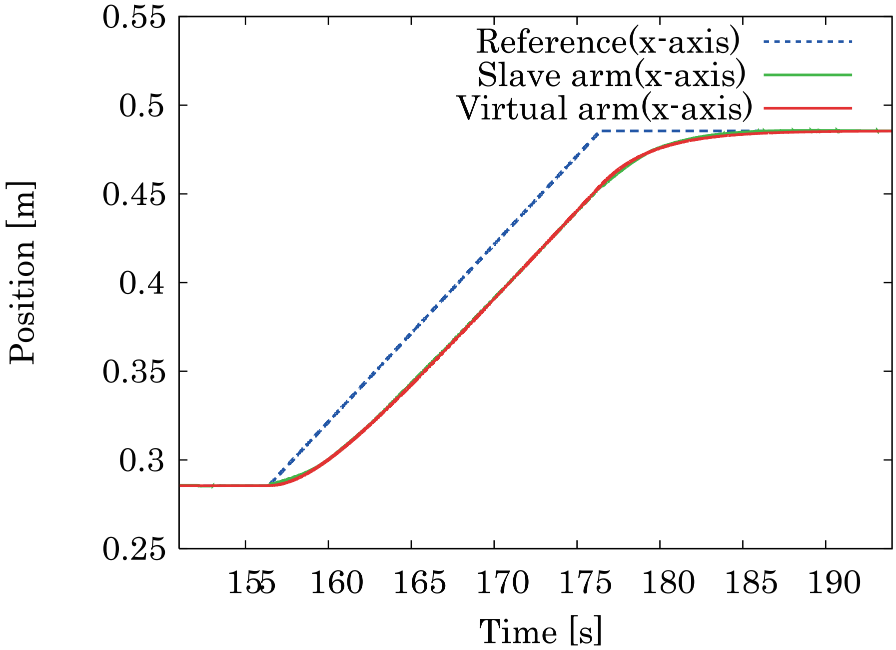


Figure 2.13: Slave arm and virtual arm positions (x-axis) (in HILS 1).

In HILS 1, the motion of the virtual arm almost agrees with the motion of the slave arm (note that the time delay was zero) since the position of the virtual arm was sent to the slave arm, and it was used in the slave arm control (the third term of the right-hand side of (2.2)).

2.5.4 The end effector force comparison between the slave arm and the virtual arm (HILS 2)

The end effector force along the x-, y-, and z-axes of the slave arm and the end effector position along the x-axis of the slave arm are demonstrated respectively in

Fig. 2.14. The reference velocity of the slave arm provided by human teleoperation is demonstrated in Fig. 2.15 (see (2.4)).

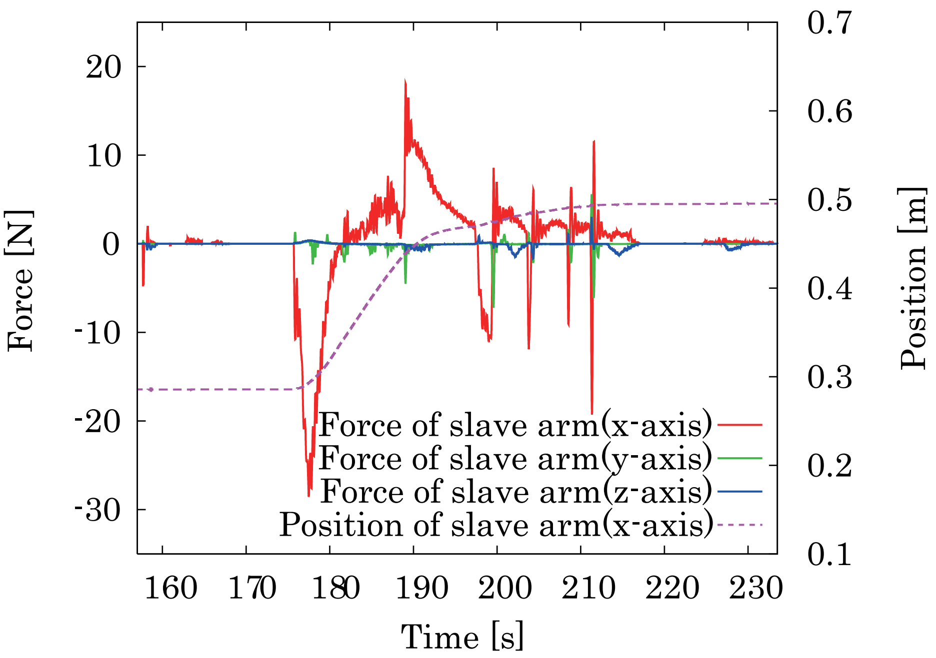


Figure 2.14: Slave arm force and position (x-axis) (in HILS 2).

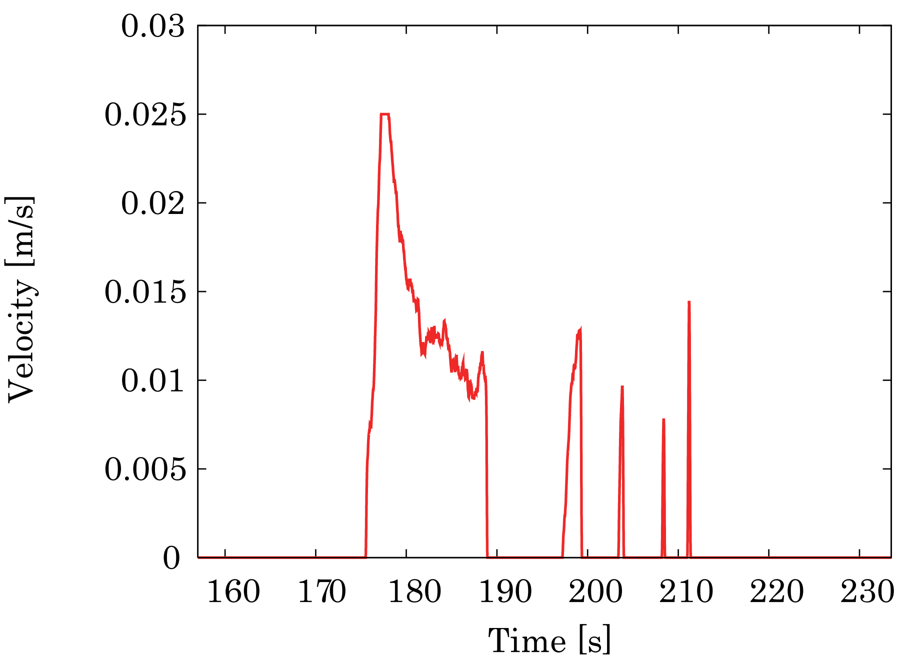


Figure 2.15: Slave arm reference velocity (x-axis) (in HILS 2).

In HILS 2, since the end effector reference velocity $\boldsymbol{v}_{s,r}$ generated by human teleoperation tends to be bigger than the constant reference velocity $\boldsymbol{v}_{s,r} = 0.01$ m/s given in HILS 1 (see Fig. 2.15), the maximum end effector force generated in HILS 2 was larger (approximately 28 N).

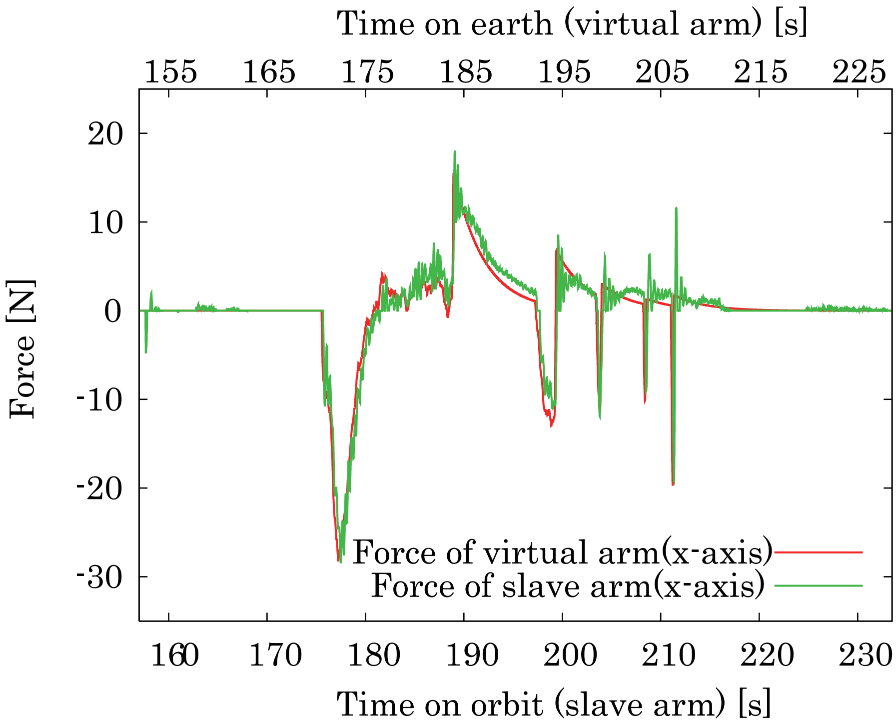


Figure 2.16: Slave arm and virtual arm forces (x-axis) (in HILS 2).

The generated forces at the slave and virtual arms are demonstrated in Fig. 2.16. Note that in order to compare the results of the slave and virtual arms, the time axis on earth (for the virtual arm) is shifted 5 s later for the time axis on orbit (for the slave arm) (see Fig. 2.16 and Fig. 2.17).

As shown in Fig. 2.16, although a small oscillation of the slave arm force was not reproduced in the generated force at the virtual arm, the force simulation was achieved appropriately.

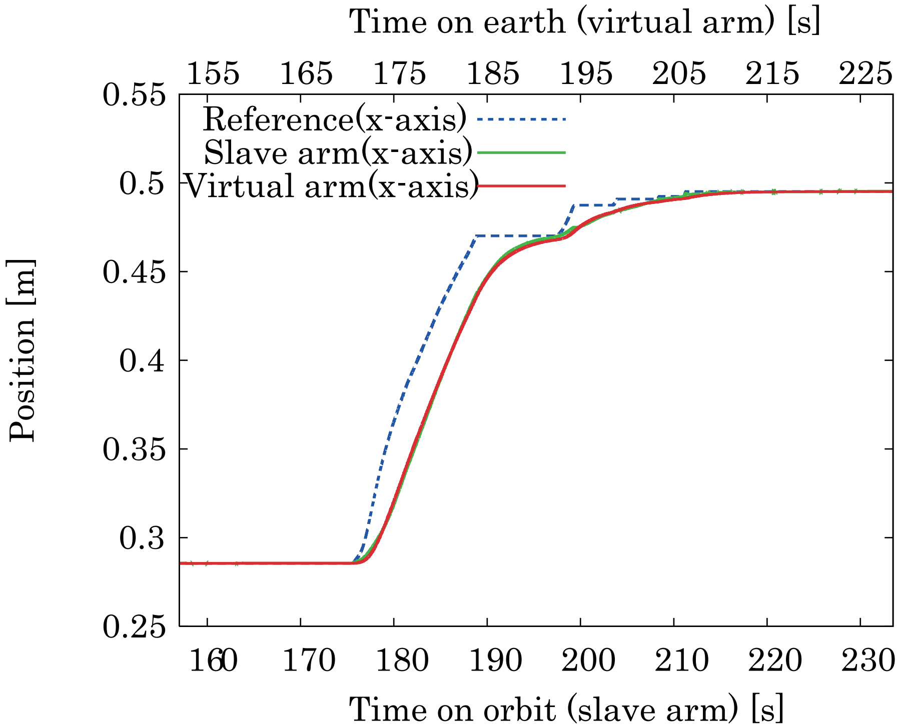


Figure 2.17: Slave arm and virtual arm positions (x-axis) (in HILS 2).

2.5.5 The end effector position comparison between the slave arm and the virtual arm (HILS 2)

The end effectors positions of the slave and virtual arms are demonstrated in Fig. 2.17. In HILS 2, the maximum position error between the arms positions was 0.002 m, which was slightly higher than the error in HILS 1.

Thus, we can claim that the motion of the slave arm was predicted by the motion of the virtual arm appropriately.

2.6 Summary

In this chapter, we proposed a teleoperation system for a massive payload transportation task. By performing hardware-in-the-loop simulations (HILSs), we verified the proposed control strategy.

It is demonstrated that the reduction of the generated force between the slave arm and the massive payload is achieved properly using the proposed control law.

Furthermore, a virtual arm control law is proposed to reproduce the dynamics of massive payloads in a virtual environment. To verify the proposed system, several HILSs were performed, and the results of the simulations were compared for the situations when the force feedback term is applied or not. Our method verified that by reducing the generated force between the end effector and the massive payload, the massive-payload manipulation task can be achieved. In the virtual environment, the movements and forces on the slave side were well reproduced, and the HILSs of massive payload transport were conducted under a time delay. In future work, the modified version of the model-based teleoperation will be utilized in more complex tasks.

Chapter 3 Massive Object Transportation by a Humanoid Robot

3.1 Introduction

As another way to transport the massive object, the method for massive object transportation using the autonomously generating an optimum motion based on a given task is presented in this chapter.

There are some researches to transport the massive object with lifting up by humanoid robots [35], [38] and [62]. However, these researches cannot deal with the massive object that cannot be lifted up due to the heavy mass of the object. There are some other researches using a cart to transport the massive object that cannot be lifted up [63] and [46]. However, when using a cart, it is necessary to lift massive objects once, and it is difficult to use the cart in a narrow place.

In this chapter, one humanoid robot pushes/pulls the massive object which cannot be lifted up to move it.

The external force is generated, when humanoid robot pushes the massive object. The external force has to be estimated to control the humanoid robot not to slip or fall. However, it is hard to estimate the exact external force, due to the friction between the ground and the massive object. In this research, the maximum value of the external force which indicates the risk of slipping is estimated, instead of the exact external force caused by friction. Also, in order to generate an optimum pushing/pulling motion for humanoid robots with less joint torque and constraints such as non-slip, realtime QP solver [64] is used.

In this chapter, the pushing and pulling simulations of massive object transportation by one humanoid robot are performed to verify the proposed method.

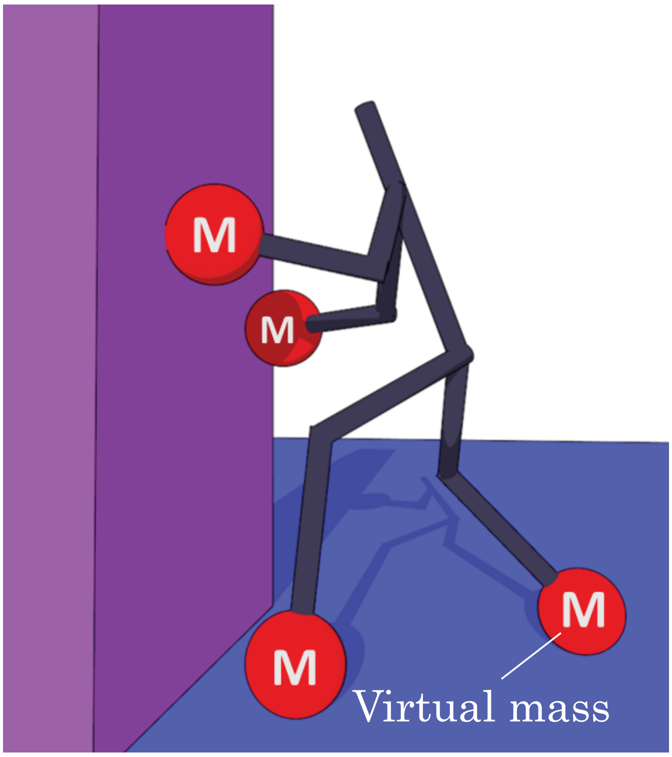


Figure 3.1: Pushing wall situation

3.2 Massive object transportation using quadratic programming

3.2.1 Humanoid robot model and dynamic simulator used in simulation

There are many sizes of humanoid robots. The small-sized humanoid robots such as HOAP [65], Robotics OP2 [66], ASIMO [67] and PLEN [68] have the advantage of the light weight and the chipper cost. However, these small-sized humanoid robots are not suitable for the massive object transportation task due to the output power limition. Therefore, the full-sized humanoid robots such as HRP [69], Atlas [70], PETMAN [71], TALOS [72] and REEM-C [73] are appropriate for the massive object transportation tasks.

In this chapter, the full-size humanoid robot is used for the massive object transportation. For the simulations, a dynamic simulator OpenHRP ([74]) using a model of HRP-2A (32-DOF)([75]) was employed to perform the manipulation in which the humanoid robot HRP-2A pushes/pulls a massive object without stepping.

3.2.2 Assumptions

In the pushing task, a quadratic programming solver was used to optimize the joint torques. Due to the large friction between the object and the ground during the pushing/pulling task, it is not easy to predict if the object moves or not. Hence, in most cases, predicting the reaction force is not possible. So in this work, the maximum reaction force was estimated by assuming the object as a fixed wall . Huge virtual masses were attached to the hands and feet in motion generation to reproduce the wall pushing situation. A massive movable object was used in the verification simulation.

3.2.3 Control laws

Objective function for optimizing torque

The humanoid robot is modeled as a free-floating robot, in which momentum and angular momentum are conserved. Inertial forces of the virtual heavy masses were used as constraints. The dynamic equation of a free-floating robot is commonly given by:

$$\boldsymbol{H}^*(\phi)\ddot{\phi} + \boldsymbol{c}^*(\phi, \dot{\phi}) + \boldsymbol{g}(\phi) = \boldsymbol{\tau}, \tag{3.1}$$

$\boldsymbol{H}^* \in R^{n\times n}$ is the generalized mass matrix ([76]), n is the number of the humanoid joints, $\boldsymbol{c}^* \in R^n$ is the centrifugal and Coriolis term, $\boldsymbol{g} \in R^n$ is the gravity term, $\boldsymbol{\tau} \in R^n$ is the joint torques and ϕ, $\dot{\phi}$, $\ddot{\phi} \in R^n$ are the joint angle, joint angular velocity and joint angular acceleration, respectively. In $\boldsymbol{H}^*$ and $\boldsymbol{c}^*$, instead of the external forces at the hands and feet, virtual masses are applied. The joint torques are normalized using the maximum joint torques as follows:

$$\boldsymbol{\tau}_n = \boldsymbol{N}\boldsymbol{\tau}, \quad \boldsymbol{N} = \begin{bmatrix} \frac{1}{\tau_{max,1}} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{\tau_{max,n}} \end{bmatrix} \tag{3.2}$$

Squared normalized torque with maximum torque

$$\min \quad \boldsymbol{\tau}_n^T \boldsymbol{\tau}_n \tag{3.3}$$

is used to generate the objective function of the QP solver. Substituting (3.1) with (3.3), the objective function is derived as follows:

$$\min \quad \ddot{\phi}^T \boldsymbol{H}^{*T} \boldsymbol{N}^T \boldsymbol{N} \boldsymbol{H}^* \ddot{\phi} + 2\boldsymbol{N}(\boldsymbol{c}^* + \boldsymbol{g}) \boldsymbol{N} \boldsymbol{H}^* \ddot{\phi} \tag{3.4}$$

In this equation, the terms which do not include joint angular acceleration are omitted since the setting variable is the joint angular acceleration.

I. Equality constraint

As an input for the reference hand position, the following equality constraint is employed:

$$\ddot{\boldsymbol{p}}_{hand} = \dot{\boldsymbol{J}}_{FtH} \dot{\phi} + \boldsymbol{J}_{FtH} \ddot{\phi} \tag{3.5}$$

$\boldsymbol{p}_{hand}$, $\dot{\boldsymbol{p}}_{hand}$, $\ddot{\boldsymbol{p}}_{hand} \in R^6$ are hand position, velocity and acceleration, respectively. $\boldsymbol{J}_{FtH} \in R^{6 \times n}$ is the Jacobian matrix from the foot to hand. From the reference hand position, the desired hand acceleration is calculated as follows:

$$\ddot{\boldsymbol{p}}_{hand} = \boldsymbol{K}_p(\boldsymbol{p}_r - \boldsymbol{p}_{hand}) - \boldsymbol{K}_v \dot{\boldsymbol{p}}_{hand} \tag{3.6}$$

$\boldsymbol{K}_p$, $\boldsymbol{K}_v$ are gain matrices. $\boldsymbol{p}_r \in R^6$ is the reference position of the hand. By substituting (3.6) with (3.5), the equality constraint of the joint acceleration is given.

II. Inequality friction constraint

Let $\boldsymbol{f}_F = [f_{Fx} \quad f_{Fy} \quad f_{Fz}]^T$ be the ground reaction force generated at the supporting foot. $\boldsymbol{f}_F$ represents the combined ground reaction force during the double support phase,. The robot is assumed to push the object along x-axis. The following inequality constraint is applied to satisfy the presence of the frictional contact between the supporting foot and ground:

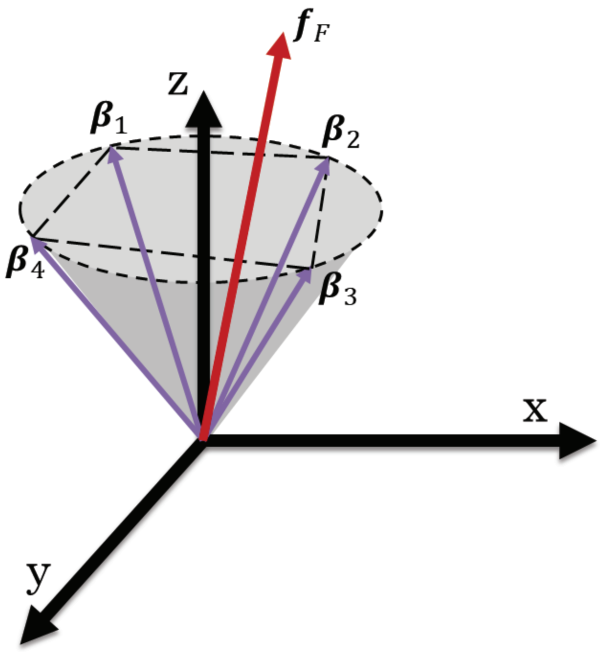


Figure 3.2: Approximated friction cone

$$\mu f_{Fz} > \sqrt{f_{Fx}^2 + f_{Fy}^2} = \frac{1}{\cos\theta} f_{Fx}, \quad \theta \simeq \tan^{-1} \frac{f_{Fy}}{f_{Fx}}, \tag{3.7}$$

where μ is the static friction coefficient. In most QP-based humanoid motion generation, coulomb friction constraint of the contact force is formulated using approximated friction cone (see Fig. 3.2) as follows:

$$\boldsymbol{f}_F = \sum_{i=1}^{m} \rho_i \boldsymbol{\beta}_i, \quad \rho_i \geq 0, \tag{3.8}$$

where $\boldsymbol{\beta}_i \in R^3$ are the basis vectors and ρ_i are the coefficients. When every ρ_i are positive, and the ground reaction force $\boldsymbol{f}_F$ is in the approximated friction cone.

However, the computational cost of using the inequality constraint given by (3.8) in QP solver is very high. Hence, the inertial force generated at the supporting foot is used as the friction constraint instead of the approximated friction cone. In motion generation, a humanoid robot is modeled as a free-floating robot (described in section 3.2.3). In this model, the gravitational force only affects the robot body and does not influence the virtual masses attached to the supporting foot and

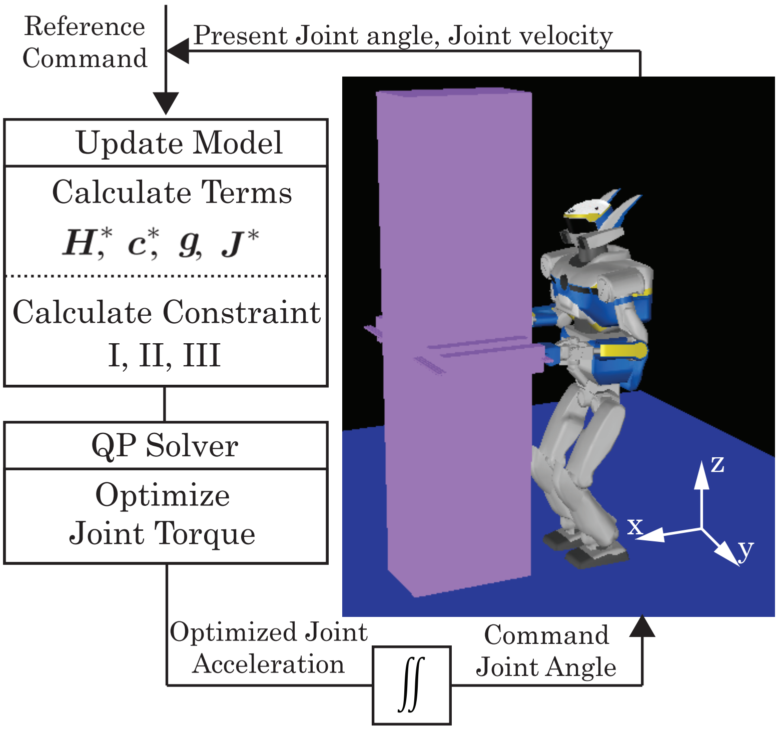


Figure 3.3: Simulation loop overview

hands. In this model, slight movements of the supporting foot and hands are allowed since the robot is free-floating.

Let $M_{foot(vir)}$, $\ddot{\boldsymbol{p}}_{foot} \in R^3$, and $\boldsymbol{g}_{part(real)} \in R^3$ be the virtual masses attached to the supporting foot (or both feet at the double support phase), acceleration vector of the supporting foot, and the gravitational term acting on the supporting foot, respectively. The ground reaction force is given by

$$\boldsymbol{f}_F = M_{foot(vir)}\ddot{\boldsymbol{p}}_{foot} + \boldsymbol{g}_{part(real)}, \tag{3.9}$$

In (3.9), the centrifugal and Coriolis terms are omitted due to the diminutive movement of the CoM. The inequality friction constraint is given by substituting (3.9) with (3.7). Since θ in (3.7) is a nonlinear function of f_{Fx} and f_{Fy}, it is not easy to apply the inequality friction constraint in QP solver. The ground reaction force is the reaction of the pushing force generated by the hand position command $\boldsymbol{p}_r$ in (3.6).

$\ddot{\boldsymbol{p}}_{foot}$ is obtained from (3.10).

$$\ddot{\boldsymbol{p}}_{foot} = \dot{\boldsymbol{J}}^{*}\dot{\phi} + \boldsymbol{J}^{*}\ddot{\phi} \tag{3.10}$$

$\boldsymbol{J}^{*} \in R^{3\times n}$ is the generalized Jacobian matrix([76]) from the waist to foot. Since $\boldsymbol{J}^{*}$ also reflects the movement of the waist on the foot, $\ddot{\boldsymbol{p}}_{foot}$ can be obtained by time derivation of $\boldsymbol{J}^{*}\dot{\phi}$. Also for $\boldsymbol{J}^{*}$, virtual mass is applied to hand and foot links.

III. Inequality joint constraint

The following inequality constraint ([77]) is used as a joint limit condition.

$$\ddot{\phi}^{\min} < \ddot{\phi} < \ddot{\phi}^{\max} \tag{3.11}$$

$$\begin{aligned}\ddot{\phi}^{\min} &= \max[\mu_p(\phi_{\min} - \phi),\ \mu_v(\dot{\phi}_{\min} - \dot{\phi})]\\ \ddot{\phi}^{\max} &= \min[\mu_p(\phi_{\max} - \phi),\ \mu_v(\dot{\phi}_{\max} - \dot{\phi})]\end{aligned}$$

$\mu_p > 0$, $\mu_v > 0$ are gain parameters. $\ddot{\phi}^{\min}$, $\ddot{\phi}^{\max}$ are minimum and maximum limitation of joint angular acceleration respectively. $\ddot{\phi}^{\min}$, $\ddot{\phi}^{\max}$ are updated in every loop of the simulation. Figure 3.3 describes the entire simulation loop.

3.3 Massive object transport simulation

3.3.1 Overview of simulation

In this chapter, the pushing and pulling simulation of massive object transportation is performed. The mass of the massive object used in both simulations is 25 kg. The virtual mass at each foot and hand are 1000 kg and 500 kg, respectively. The coefficient of the static friction is 0.8. The coefficient of the kinetic friction is 0.6. The same friction coefficients are applied between the floor and the massive object and between the floor and the humanoid robot feet.

The free-floating and virtual mass models are used in the pushing/pulling motion generattion so that (3.3) is minimized subject to (3.5) (3.7), and (3.11). In OpenHRP Simulator, the generated motion is simulated. As illustrated in Fig. 3.3, the next motion is generated online by referring to the simulation results. During each loop of the dynamic simulation, the desired motion is generated.

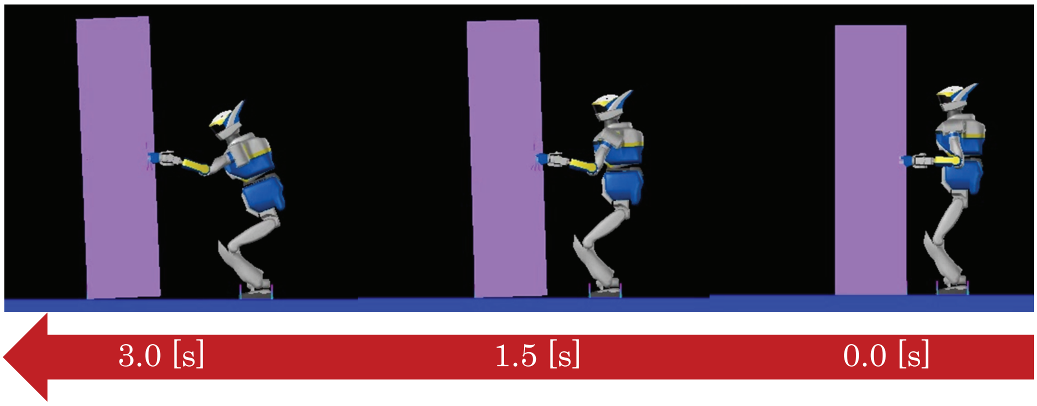


Figure 3.4: Pushing simulation overview

3.3.2 Pushing simulation

The massive object gets tilted in the pushing direction due to the friction, during the pushing task. A reference command is placed in both the positive direction of x-axis and negative direction of z-axis to suppress the effect of the titled object. $p_{hand,x} + 0.045$ m and $p_{hand,z} - 0.006$ m are the reference command positions of the x-axis and z-axis, respectively. The reference command is defined as the initial position of the other axes; therefore the initial position of these axes is constant.

Result of the force/torque sensors (pushing) In the pushing simulation, the object is pushed by the robot to the positive direction of the x-axis. The reaction force is detected using the the wrist force sensors of the robot. Therefore the detected forces along x-axis are negative (see Figs. 3.5 and 3.6). f_{FSx}, f_{FSy}, f_{FSz} are the force sensor data of each axis. In Figs. 3.7 and 3.8, $Cond = \mu f_{FSz} - \sqrt{f_{FSx}^2 + f_{FSy}^2}$ is demonstrated to make sure that the non-slip condition is fulfilled. The force generated along the z-axis is sufficiently more significant than the forces generated along the x and y-axes. Therefore, we can claim that the pushing task is performed without slipping. In Figs. 3.9 and 3.10, only the torque in y-axis is about 38 Nm, due to the massive object tilting during the pushing task, and the torques in x and z-axes are almost zero. In Figs. 3.11 and 3.12, only the torque in y-axis vibrates around 0 Nm, and the torques in x and z-axes are almost zero.

Result positions of the right and left-hand (pushing) In the pushing sim-

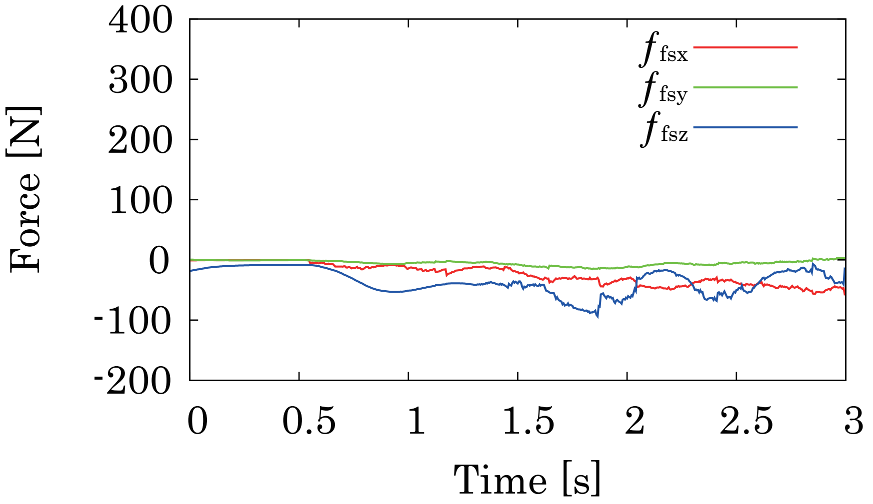


Figure 3.5: Right hand forces (pushing)

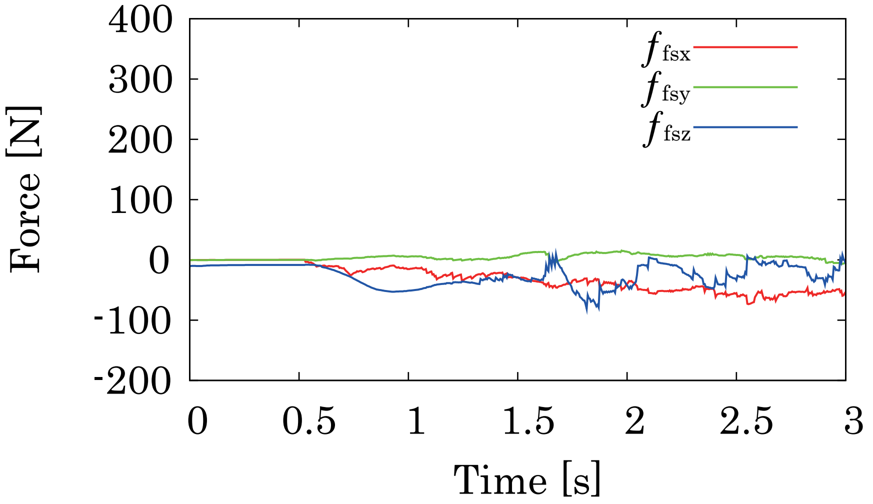


Figure 3.6: Left hand forces (pushing)

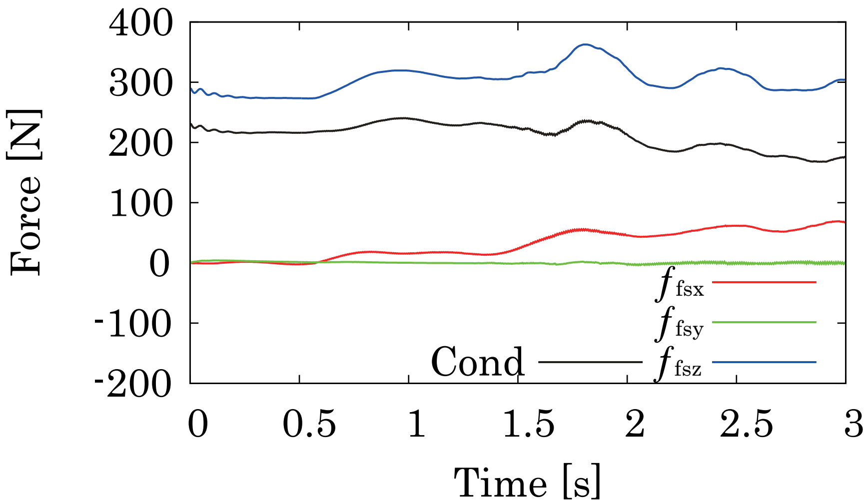


Figure 3.7: Right foot forces (pushing)

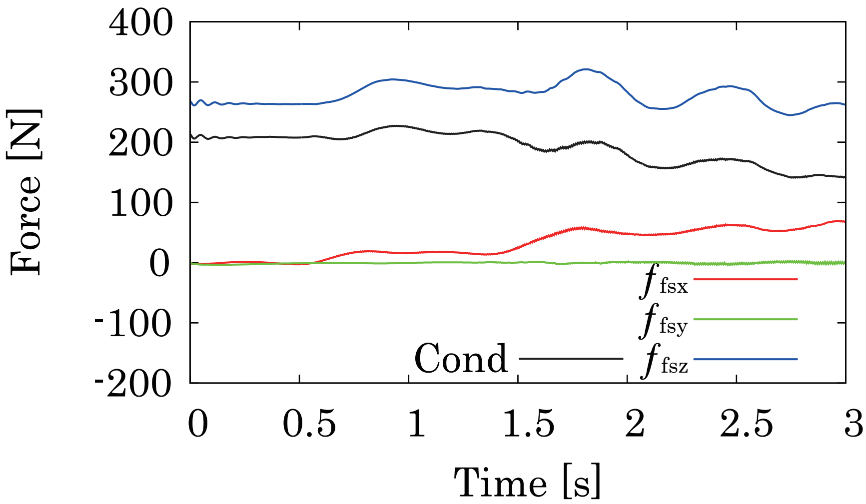


Figure 3.8: Left foot forces (pushing)

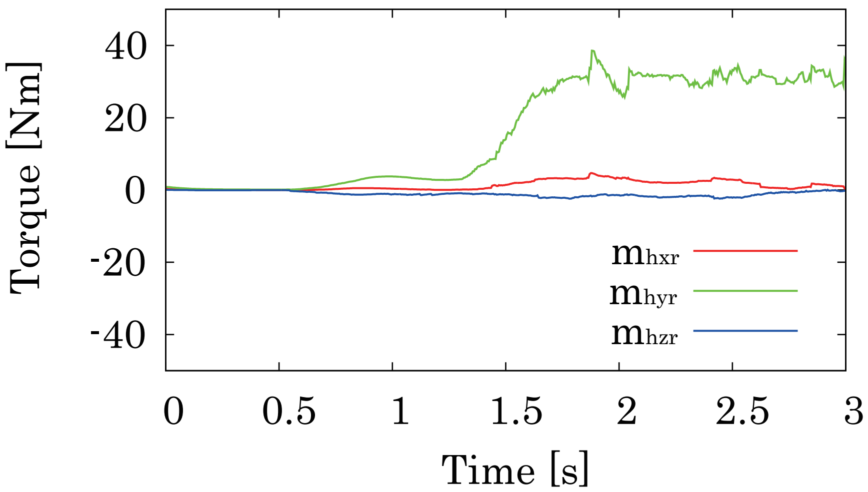


Figure 3.9: Right hand torques (pushing)

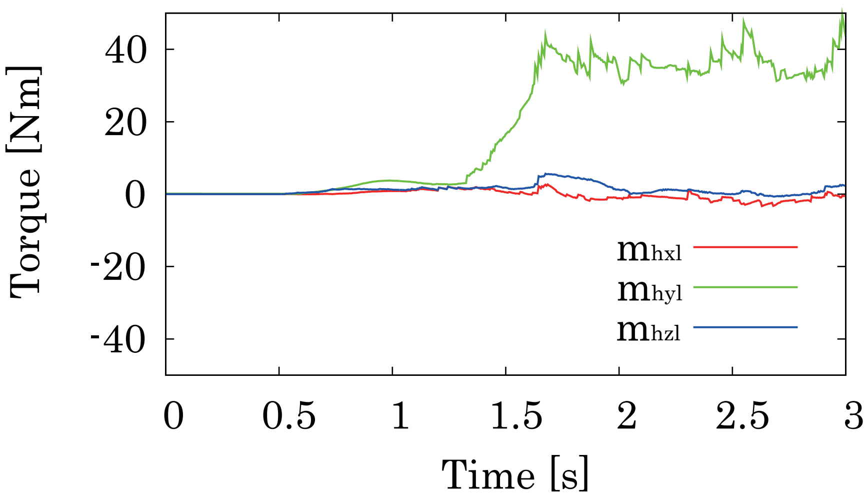


Figure 3.10: Left hand torques (pushing)

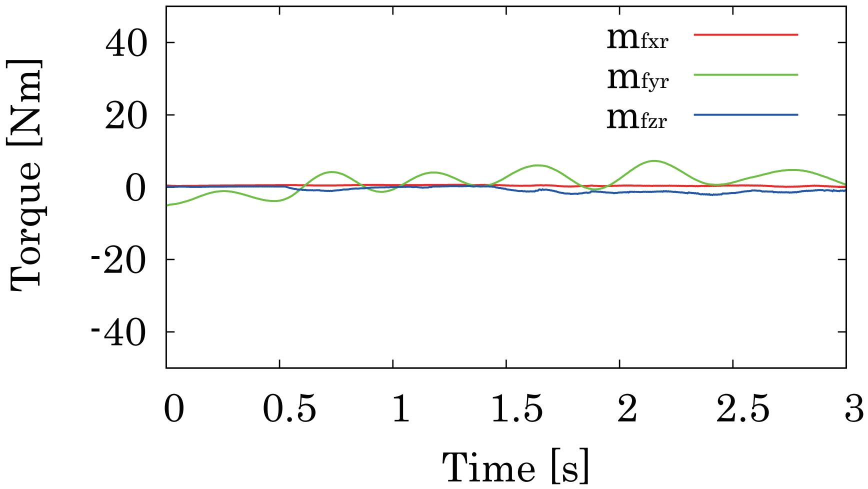


Figure 3.11: Right foot torques (pushing)

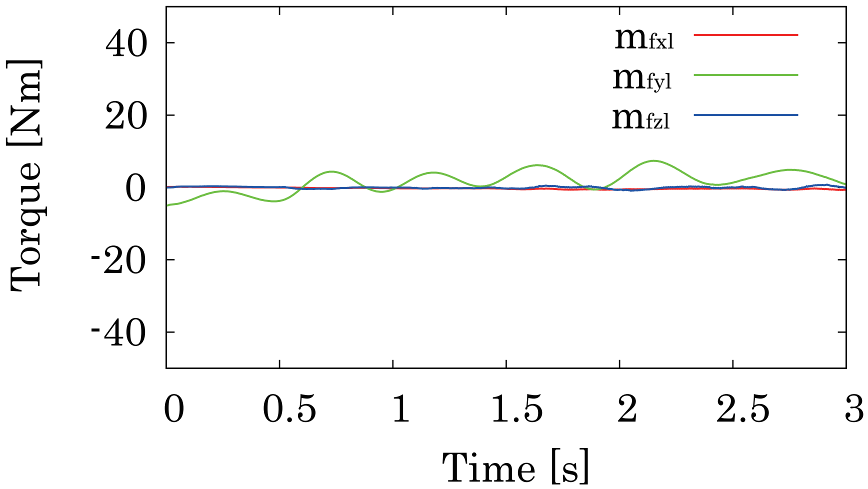


Figure 3.12: Left foot torques (pushing)

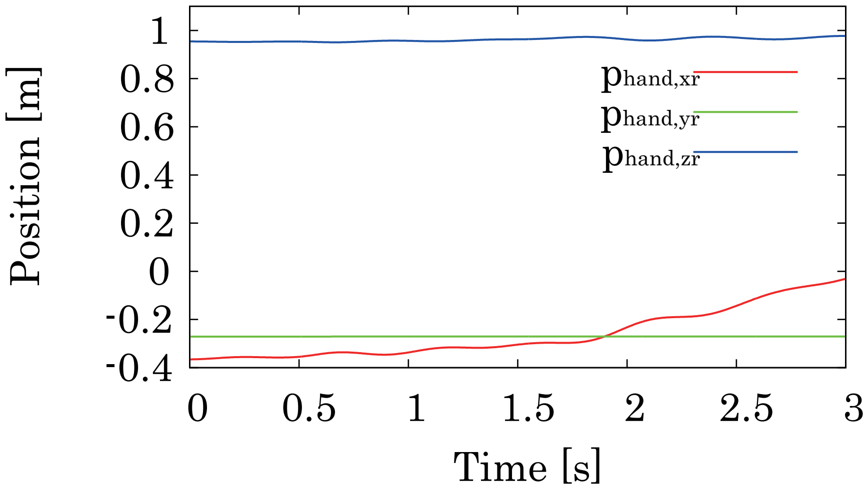


Figure 3.13: Right hand positions (pushing)

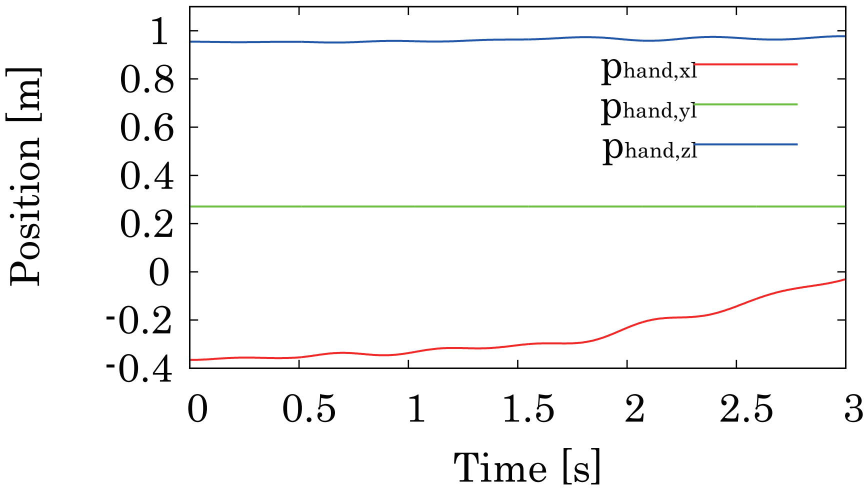


Figure 3.14: Left hand positions (pushing)

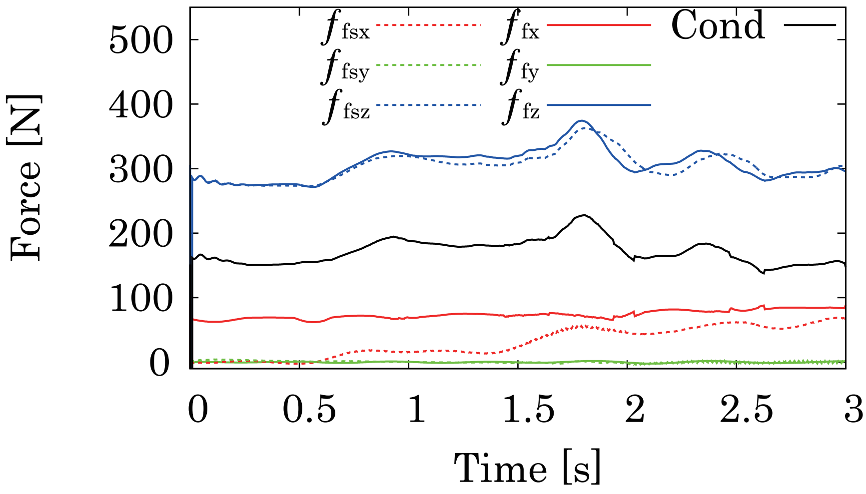


Figure 3.15: The force sensor data vs estimated force in right foot (pushing)

ulation, positions of the both hand in y and z-axes are almost constant, as shown in Figs. 3.13 and 3.14. The hands moved along x-axis about 0.39 m.

Comparison between the force sensor data and the estimated maximum force in foot (pushing) In Fig. 3.15, $Cond = \mu f_{Fz} - \sqrt{f_{Fx}^2 + f_{Fy}^2}$ is demonstrated to make sure that the non-slip condition is fulfilled. We can claim that the non-slip condition is conservatively fulfilled since the value of the estimated force in the x-axis is greater than the value of the force sensor.

3.3.3 Pulling simulation

The massive object gets tilted in the pulling direction due to the friction, during the pulling task. A reference command is placed in both the negative direction of x-axis and positive direction of z-axis to suppress the effect of the titled object. $p_{hand,x} - 0.045$ m and $p_{hand,z} + 0.006$ m are the reference command positions of the x-axis and z-axis, respectively. The reference command is defined as the initial position of the other axes; therefore the initial position of these axes is constant.

Result of the force/torque sensors (pulling) In the pulling simulation, the object is pulled by the robot to the negative direction of the x-axis. The reaction force is detected using the the wrist force sensors of the robot. Therefore the detected forces along x-axis are positive (see Figs. 3.17 and 3.18). In Figs. 3.19

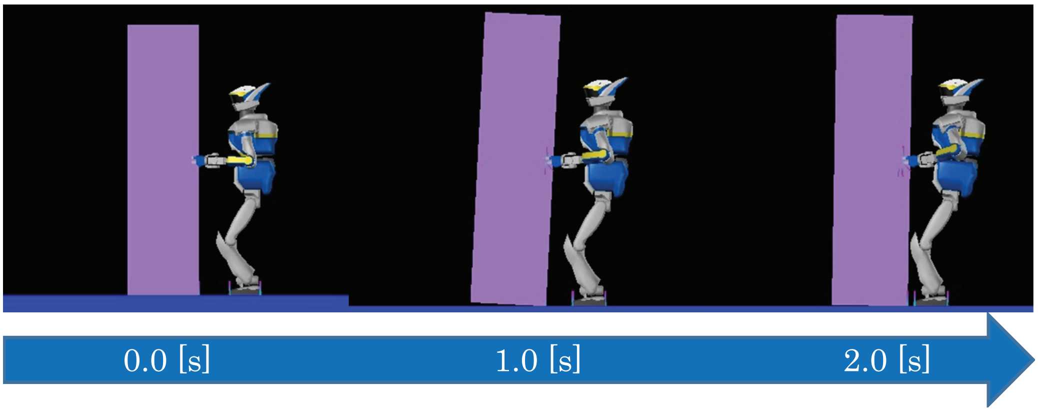


Figure 3.16: Pulling simulation overview

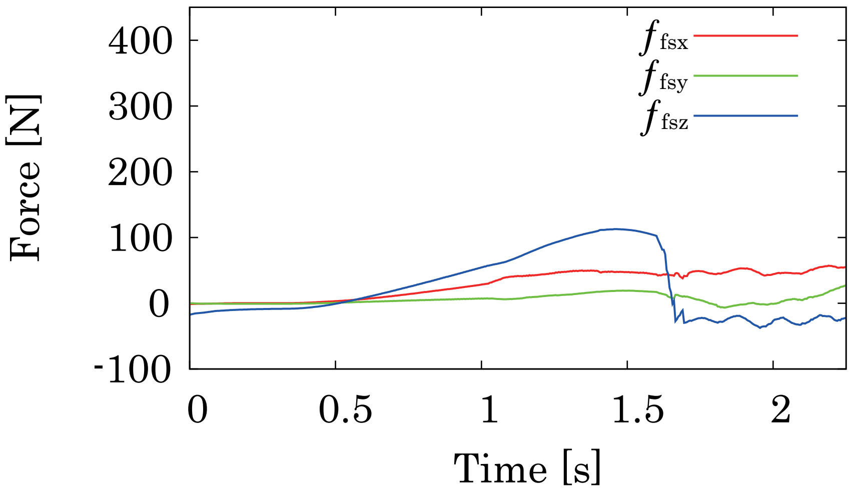


Figure 3.17: Right hand forces (pulling)

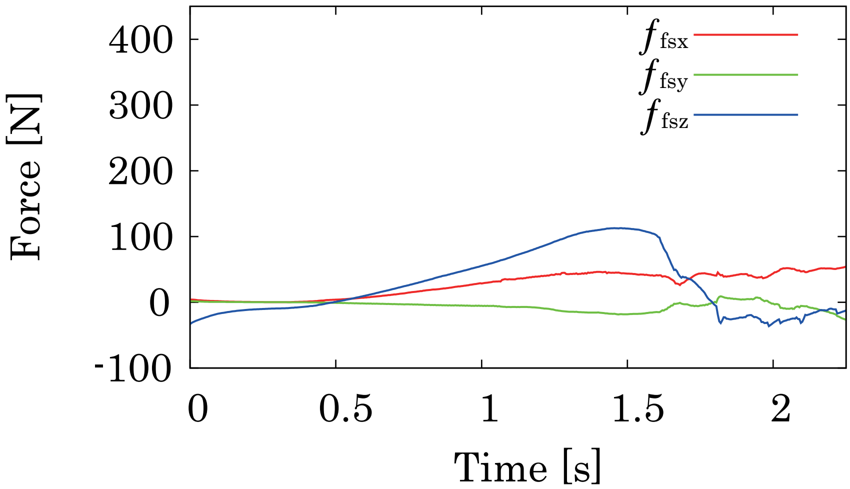


Figure 3.18: Left hand forces (pulling)

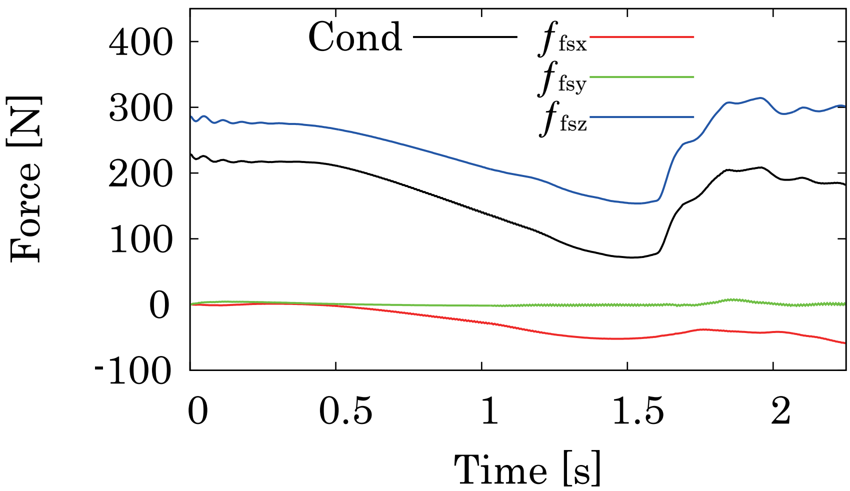


Figure 3.19: Right foot forces (pulling)

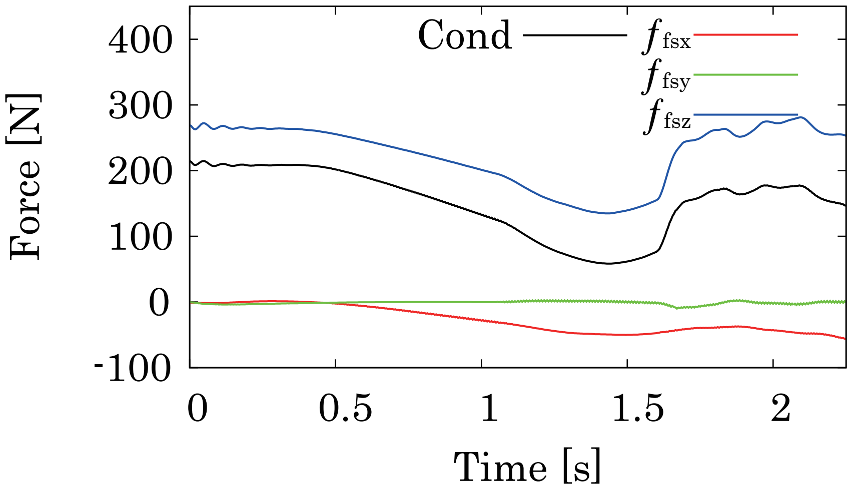


Figure 3.20: Left foot forces (pulling)

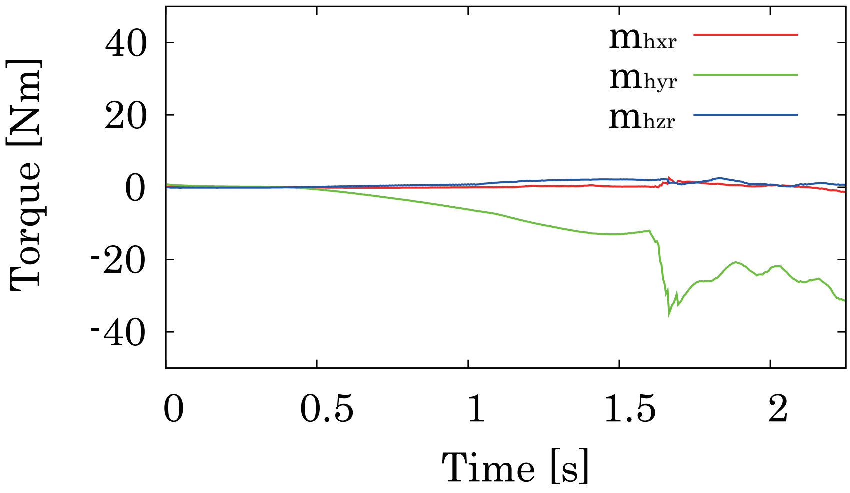


Figure 3.21: Right hand torques (pulling)

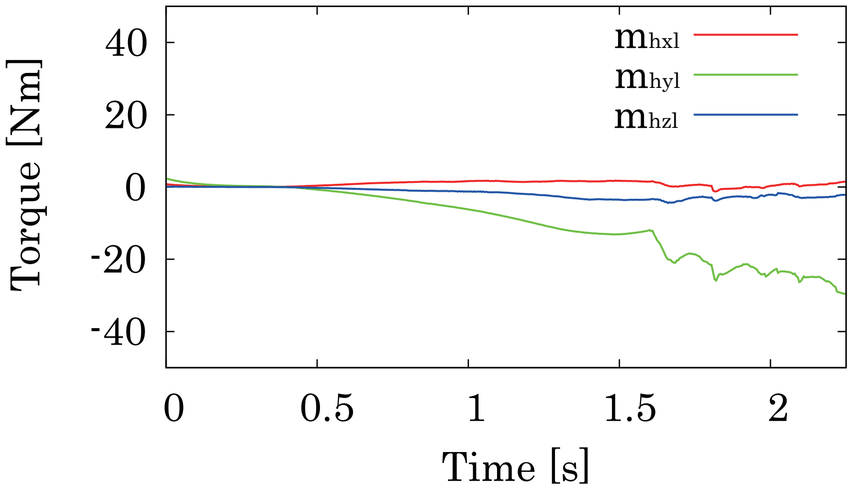


Figure 3.22: Left hand torques (pulling)

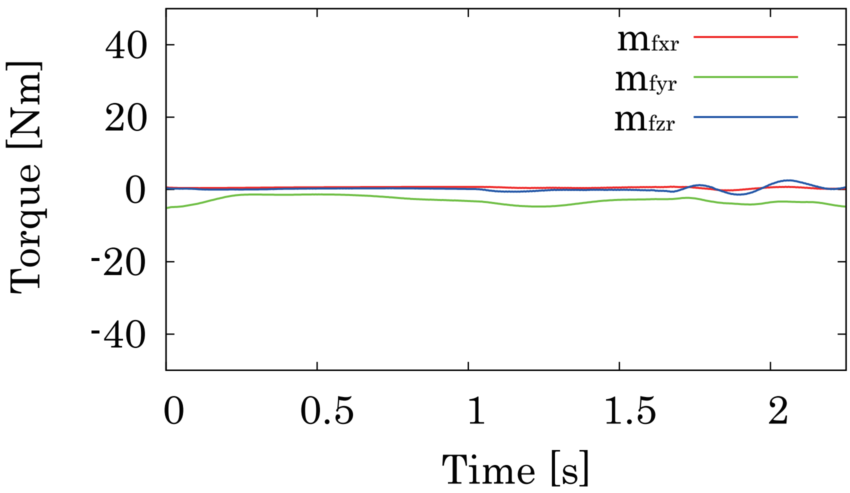


Figure 3.23: Right foot torques (pulling)

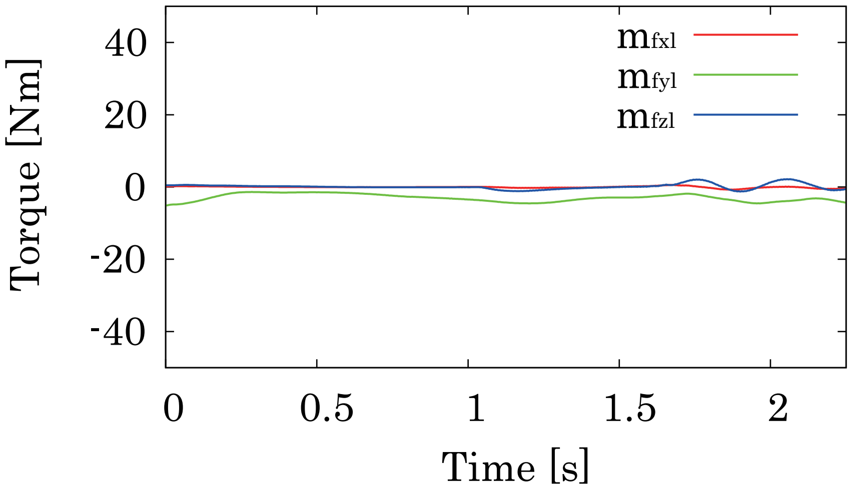


Figure 3.24: Left foot torques (pulling)

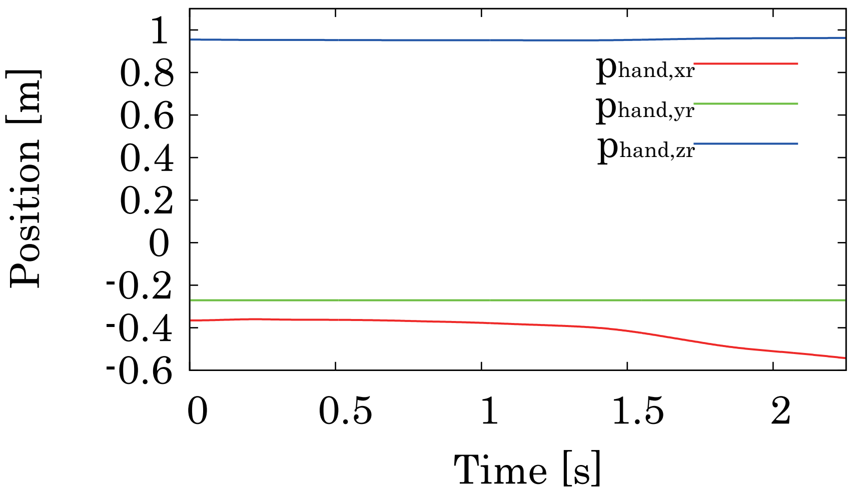


Figure 3.25: Right hand positions (pulling)

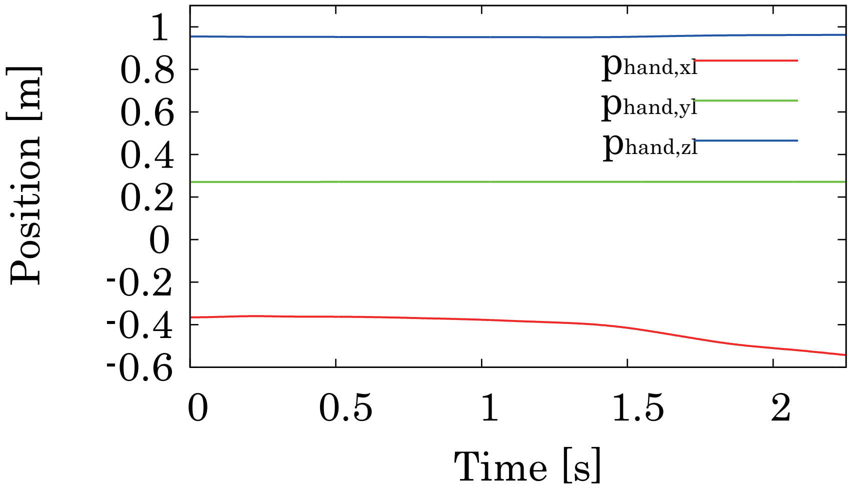


Figure 3.26: Left hand positions (pulling)

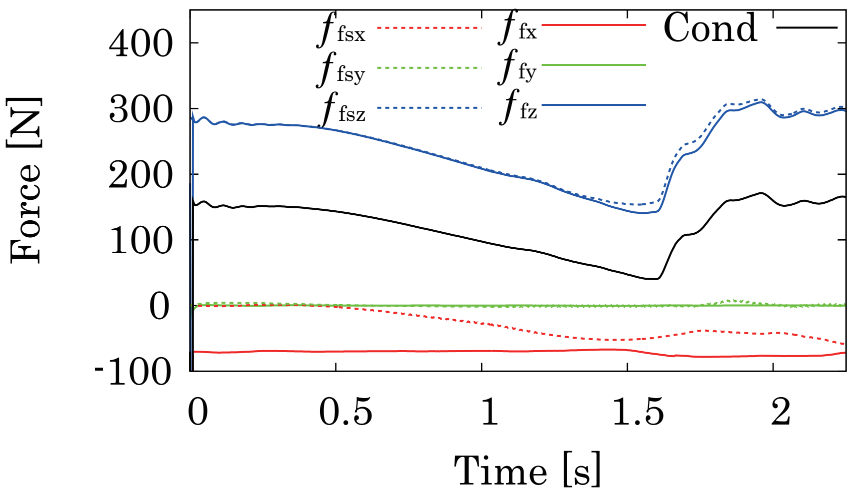


Figure 3.27: The force sensor data vs estimated force in right foot (pulling)

and 3.20, The force generated along the z-axis is sufficiently more significant than the forces generated along the x and y-axes. Therefore, we can claim that the pushing task is performed without slipping. In Figs. 3.21 and 3.22, only the torque in y-axis is about -38 Nm, due to the massive object tilting during the pulling task, and the torques in x and z-axes are almost zero. In Figs. 3.23 and 3.24, only the torque in y-axis is about -4 Nm, and the torques in x and z-axes are almost zero.
Result positions of the right and left hand (pulling) In the pulling simulation, positions of the both hand in y and z-axes are almost constant, as shown in Figs. 3.25 and 3.26. The hands moved along x-axis about 0.20 m.
Comparison between the force sensor data and the estimated maximum force in foot (pulling) In Fig. 3.27, We can claim that the non-slip condition is conservatively fulfilled since the value of the estimated force in the x-axis is greater than the value of the force sensor.

3.4 Summary

In this chapter, the robot motion generation problem is discussed in the massive object transportation task using a humanoid robot. In the proposed motion generation, a free-floating model is utilized and virtual masses are applied at the end of the limbs as the constraint.

The feet and hands acceleration are used to assess the constraint forces. Then, these forces are used as the inequality constraint conditions of the QP problem.

The motions of the pushing/pulling massive object are generated online using a real-time QP solver. By performing dynamic simulations using OpenHRP, the generated motions are verified.

The motion generation method proposed in this chapter will be extended to the motion generation of multiple humanoid robots acting on a cooperative task.

Chapter 4 Massive Object Transportation by Two Humanoid Robots

4.1 Introduction

As an extension of massive object transportation by one humanoid robot discussed in chapter 3, this chapter presents massive object transportation by two humanoid robots.

There are previous researches which transport the massive object by cooperation of humanoid robots. In [39], a symmetry position/force hybrid control for cooperative transportation by multiple humanoid robots are proposed. In [38], cooperative transportation by enhancing the trackability of given trajectory of the two humanoid robots are performed. In [40], two small humanoid robots hold a single object and considered as a quadruped robot. However, in these researches, the target was light enough to lift up by humanoid robots.

On the other hand, there are some researches to use a cart to transport the massive object [47], [46]. However, when using a cart, it is necessary to lift massive objects once, and it is difficult to use the cart in a narrow place.

In order to overcome this problems, a scenario is assumed that the massive object is too heavy to lift up, and carts cannot be used. The massive object mass is 200 kg which is 3 times heavier than the mass of the humanoid robot. Therefore, the humanoid robot pushes a massive object and the other humanoid robot pulls the object simultaneously.

The external force is generated, when humanoid robot pushes the massive ob-

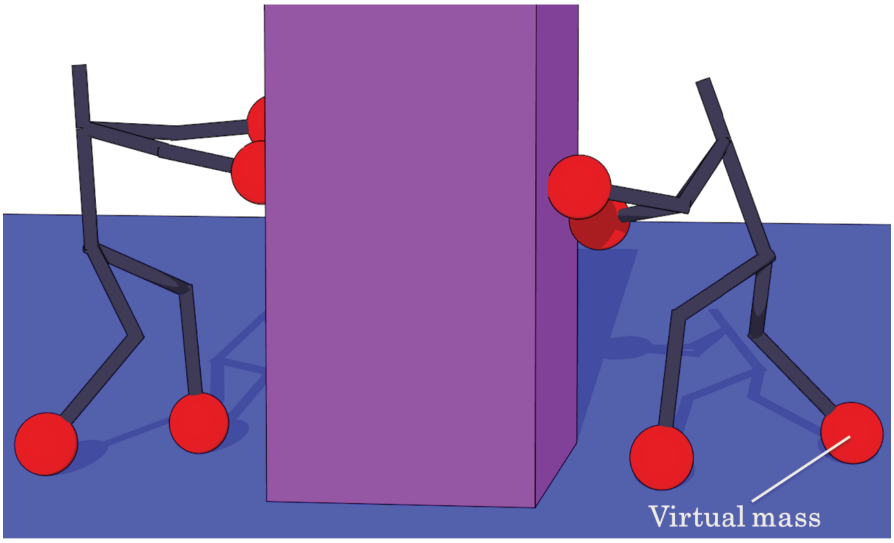


Figure 4.1: Concept of wall pushing/pulling situation

ject. The external force has to be estimated to control the humanoid robot not to slip or fall. However, it is hard to estimate the exact external force, due to the friction between the ground and the massive object.

Instead of estimating the exact external force, the maximum reaction force is estimated in this research. The maximum reaction force is estimated as Fig. 4.1, assuming that the huge virtual masses are attached to the end of the limbs of humanoid robots.

There are some researches for the cooperation task for humanoid robots to transport the object using QP solver [42]. In [42], contact force and robot posture are stabilized by using a QP solver. However, in [42], external forces are not considered such as frictions between the ground and the massive object. In this research, in order to generate the massive object transportation motion with less joint torque and without slipping of feet due to the friction, a realtime QP solver is used to optimize the joint torque when transporting the massive object with constraints such as non-slip condition [64].

In this chapter dynamic simulations of massive object (200 kg) transportation by two humanoid robots were performed. In order to show the advantage of proposed method, the simulation using a conventional PD controller is performed and the result of two simulations (QP vs PD) are compared.

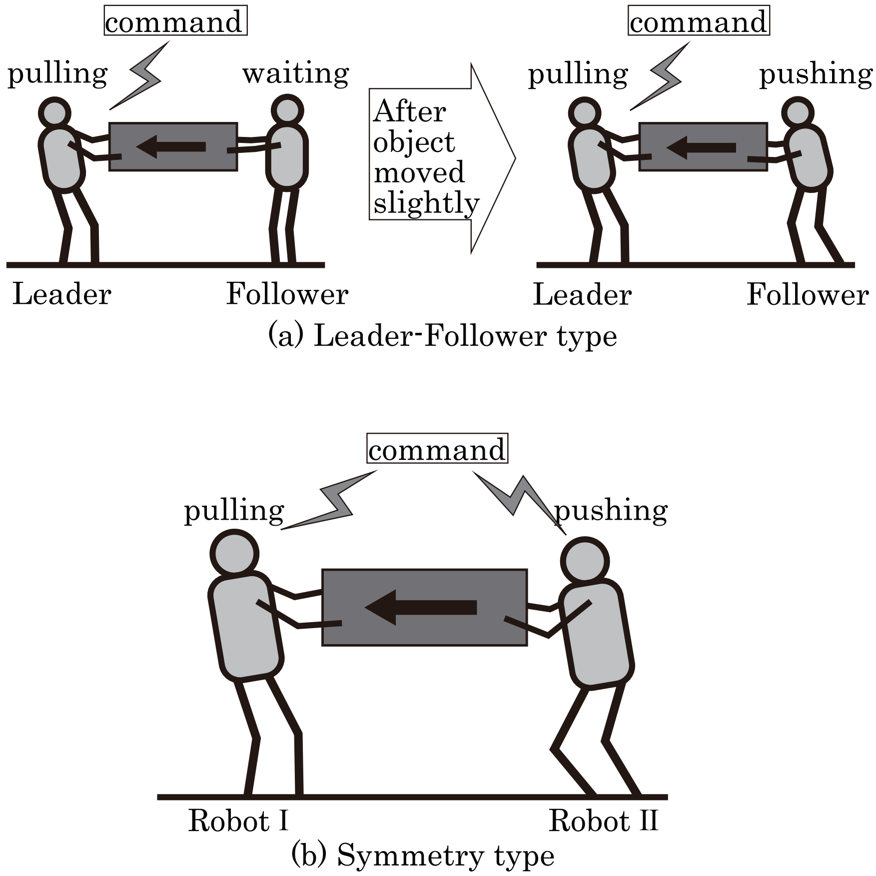


Figure 4.2: Two types of multiple humanoid robots cooperation.

4.2 Quadratic programming based online optimal motion generation

4.2.1 Assumptions

Generally, there are two types of multiple humanoid robots cooperation (see Fig. 4.2). In the leader-follower type, the follower recognizes the movement of the object since the leader starts to transport the object. The force sensor of the follower's hand helps it to realize the commencement of the motion, and then the robot can start to carry the object subsequently. If the object is too heavy to transport by one humanoid, since the follower cannot recognize the force caused by the movement of the target object, the leader-follower type cannot be utilized. Therefore in this chapter, the symmetry type is used.

A model of HRP-2A (32-DOF), and a model of HRP-2 (30-DOF) [75] are employed to perform dynamic simulations in an OpenHRP [74]. In this simulation, the HRP-2A and HRP-2 humanoid robots are assumed to push and pull a massive object simultaneously without stepping, respectively. Furthermore, the object is considered to be too heavy to lift, so the humanoid robots grasp a part of the object and transport it by a simultaneous pulling and pushing action. Hence there is no slippage between the hands and the object.

4.2.2 Pushing/pulling motion generation

For both pushing and pulling humanoid robots, the same motion generation algorithm is applied.

Objective Function

For motion generation, a free-floating robot model is employed. The momentum and angular momentum are conserved in the free-floating robot mode. Using inertial forces of virtual heavy masses, constraints are modeled. The dynamic equation of a free-floating robot is generally given by:

$$\boldsymbol{H}^*(\boldsymbol{\phi})\ddot{\boldsymbol{\phi}} + \boldsymbol{c}^*(\boldsymbol{\phi}, \dot{\boldsymbol{\phi}}) + \boldsymbol{g}(\boldsymbol{\phi}) = \boldsymbol{\tau}, \tag{4.1}$$

where $\boldsymbol{H}^* \in \Re^{n \times n}$ is the generalized mass matrix [76], n is the number of the robot joints, $\boldsymbol{c}^* \in \Re^n$ is centrifugal and Coriolis term, $\boldsymbol{g} \in \Re^n$ is the gravity term, $\boldsymbol{\tau} \in \Re^n$ is the joint torques and $\boldsymbol{\phi}$, $\dot{\boldsymbol{\phi}}$, $\ddot{\boldsymbol{\phi}} \in \Re^n$ are the joint angle, joint angular velocity and joint angular acceleration, respectively. As stated earlier, virtual heavy masses are attached at the ends of the limbs to represent the frictional force applied to the feet from the ground and the reaction force applied to the hands from the massive object. However the gravitational force is applied to the humanoid robot, the attached virtual heavy masses are assumed to be free-floating. The pulling and pushing motion of the robots are generated independently. Using the maximum joint torques, the joint torques are normalized as follows:

$$\boldsymbol{\tau}_n = \boldsymbol{N}\boldsymbol{\tau}, \quad \boldsymbol{N} = \begin{bmatrix} \frac{1}{\tau_{max,1}} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{\tau_{max,n}} \end{bmatrix}. \tag{4.2}$$

The following objective function is employed in the motion generation:

$$F(\boldsymbol{\tau}_n, \ddot{\boldsymbol{p}}_d, \ddot{\boldsymbol{p}}_{hand}) = w_1 \boldsymbol{\tau}_n^T \boldsymbol{\tau}_n + w_2 \|\boldsymbol{B}_O(\ddot{\boldsymbol{p}}_d - \ddot{\boldsymbol{p}}_{hand})\|^2 \tag{4.3}$$

where to extract x and z components are extracted from the $\boldsymbol{B}_O$ matrix. (for the coordinate system see Fig. 4.3) given by

$$\boldsymbol{B}_O = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

w_1, w_2 are weight coefficient. $\boldsymbol{p}_d$, $\boldsymbol{p}_r$, and $\boldsymbol{p}_{hand}$ are the six dimensional desired, reference and actual hand position vectors, respectively. $\ddot{\boldsymbol{p}}_d$ is given as follows:

$$\ddot{\boldsymbol{p}}_d = \boldsymbol{K}_p(\boldsymbol{p}_r - \boldsymbol{p}_{hand}) - \boldsymbol{K}_v \dot{\boldsymbol{p}}_{hand} \tag{4.4}$$

where $\boldsymbol{K}_p$ and $\boldsymbol{K}_v$ are diagonal gain matrices. In order to generate the pushing/pulling forces of the robots, the reference hand position $\boldsymbol{p}_r$ is designed. The actual hand acceleration vector is calculated using

$$\ddot{\boldsymbol{p}}_{hand} = \dot{\boldsymbol{J}}_{FtH}\dot{\boldsymbol{\phi}} + \boldsymbol{J}_{FtH}\ddot{\boldsymbol{\phi}}, \tag{4.5}$$

where $\boldsymbol{\phi}$ is the joint angle vector, and $\boldsymbol{J}_{FtH}$ is the Jacobian matrix from the foot to the hand. The objective function (4.3) can be written as function of $\ddot{\boldsymbol{\phi}}$ as follows:

$$\begin{aligned} F(\ddot{\boldsymbol{\phi}}) \;=\;& w_1 \left\{ \ddot{\boldsymbol{\phi}}^T \boldsymbol{H}^{*T} \boldsymbol{N}^T \boldsymbol{N} \boldsymbol{H}^* \ddot{\boldsymbol{\phi}} + 2(\boldsymbol{c}^* + \boldsymbol{g})^T \boldsymbol{N}^T \boldsymbol{N} \boldsymbol{H}^* \ddot{\boldsymbol{\phi}} \right\} \\ & + w_2 \left\{ \ddot{\boldsymbol{\phi}}^T \boldsymbol{J}_{FtH}^T \boldsymbol{B}_O^T \boldsymbol{B}_O \boldsymbol{J}_{FtH} \ddot{\boldsymbol{\phi}} \right. \\ & \left. + 2(\dot{\boldsymbol{\phi}}^T \dot{\boldsymbol{J}}_{FtH}^T - \ddot{\boldsymbol{p}}_d^T) \boldsymbol{B}_O^T \boldsymbol{B}_O \boldsymbol{J}_{FtH} \ddot{\boldsymbol{\phi}} \right\} + \boldsymbol{N}_a, \end{aligned} \tag{4.6}$$

where $\boldsymbol{N}_a$ is the non-linear term without the joint acceleration and is given by:

$$\begin{aligned} \boldsymbol{N}_a \;=\;& w_1(\boldsymbol{c}^* + \boldsymbol{g})^T \boldsymbol{N}^T \boldsymbol{N}(\boldsymbol{c}^* + \boldsymbol{g}) \\ & + w_2 \left\{ \ddot{\boldsymbol{p}}_d^T \boldsymbol{B}_O^T \boldsymbol{B}_O \ddot{\boldsymbol{p}}_d + (\boldsymbol{\phi}^T \boldsymbol{J}_{FtH}^T - 2\ddot{\boldsymbol{p}}_d^T) \boldsymbol{B}_O^T \boldsymbol{B}_O \dot{\boldsymbol{J}}_{FtH} \dot{\boldsymbol{\phi}} \right\}. \end{aligned} \tag{4.7}$$

In order to minimize the objective function F given in (4.6) subject to the equality/inequality constraints, the QP problem is used to find the joint acceleration vector $\ddot{\boldsymbol{\phi}}$. In the following subsections, the constraints are defined.

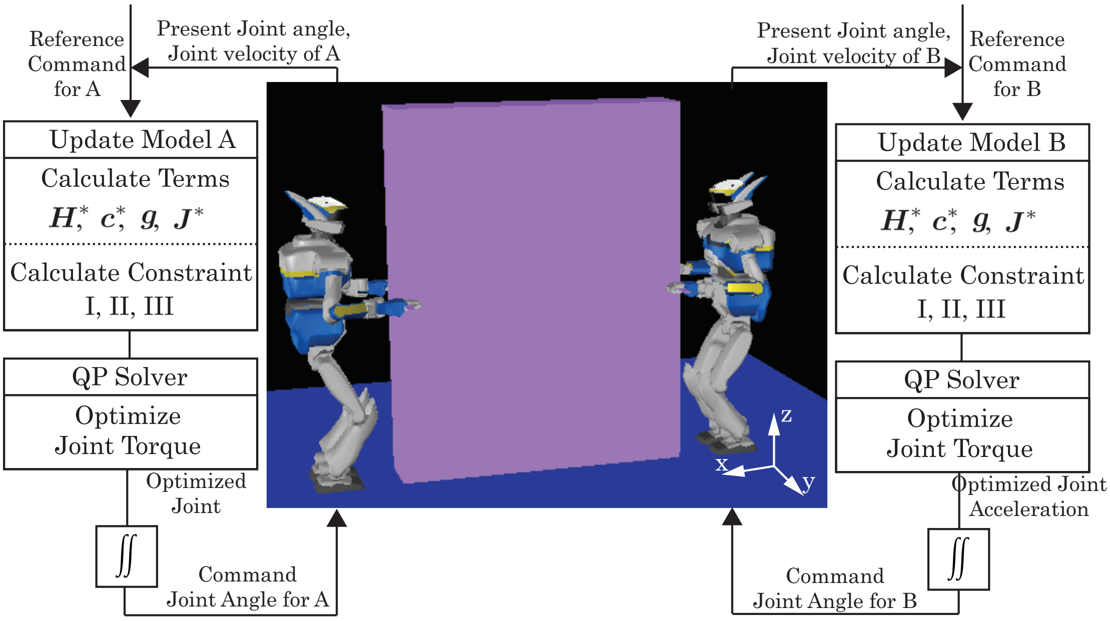


Figure 4.3: Overview of simulation loop

Hand reference input constraint

The following equality constraint constrains the orientation and the movement along the y-axis of the hands (see Fig. 4.3):

$$\boldsymbol{B}_c \ddot{\boldsymbol{p}}_{hand} = \boldsymbol{B}_c \ddot{\boldsymbol{p}}_d, \tag{4.8}$$

where $\boldsymbol{B}_c$ is a matrix to extract y and postural components given by:

$$\boldsymbol{B}_c = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

$\ddot{\boldsymbol{p}}_d$ and $\ddot{\boldsymbol{p}}_{hand}$ are given by (4.4) and (4.5), respectively. By solving the QP problem using the objective function (4.3), the hand acceleration along x and z axes are achieved (see Fig. 4.3). However, in the QP problem, the hand acceleration along x and z axes are bounded by following inequality constraints:

$$\ddot{\boldsymbol{p}}_{hand,x,\min} < \ddot{\boldsymbol{p}}_{hand,x} < \ddot{\boldsymbol{p}}_{hand,x,\max}$$
$$\ddot{\boldsymbol{p}}_{hand,z,\min} < \ddot{\boldsymbol{p}}_{hand,z} < \ddot{\boldsymbol{p}}_{hand,z,\max}.$$

Frictional constraint

In most of QP problems aimed for humanoid robot motion generation, approximated friction cone (Fig. 3.2) has been employed to represent the Coulomb friction constraint [48], [78], [79], [80]. Using the approximated friction cone, the friction force is calculated as follows:

$$\boldsymbol{f}_F = \sum_{i=1}^{m} \rho_i \boldsymbol{\beta}_i, \quad \rho_i \geq 0, \tag{4.9}$$

where $\boldsymbol{\beta}_i \in \Re^3$ are the basis vectors and ρ_i are the coefficients. All ρ_i must be positive to satisfy the non-slip condition. By creating the non-slip inequality constraint in QP problems using (4.9), the computational cost increases drastically. Therefore, instead of using the approximated friction cone, the inertial force generated at the supporting foot is assumed as a virtual mass. Lastly, it is employed to obtain non-slip constraints.

The ground reaction force generated at the supporting foot is indicated as $\boldsymbol{f}_F = [f_{Fx} \quad f_{Fy} \quad f_{Fz}]^T$. The following inequality constraint gives the condition to keep frictional contact between the supporting foot and ground:

$$\mu f_{Fz} > \sqrt{f_{Fx}^2 + f_{Fy}^2} = \frac{1}{\cos\theta} f_{Fx}, \quad \theta \simeq \tan^{-1}\frac{f_{Fy}}{f_{Fx}}, \tag{4.10}$$

which μ is the static friction coefficient. $\boldsymbol{f}_F$ can be obtained as an inertial force provided at the virtual mass since a virtual heavy mass is attached to the supporting foot (or feet).

The virtual mass attached to the supporting foot, six-dimensional position vector of the supporting foot, and gravitational term acting on the supporting foot are indicated as $M_{foot(vir)}$, $\boldsymbol{p}_{foot} \in \Re^6$, and $\boldsymbol{g}_{part(real)} \in \Re^3$, respectively. The ground reaction force is obtained by

$$\boldsymbol{f}_F = M_{foot(vir)} \boldsymbol{B}_P \ddot{\boldsymbol{p}}_{foot} + \boldsymbol{g}_{part(real)}, \tag{4.11}$$

where $\boldsymbol{B}_P$ is a matrix to extract x, y, and z components of the foot acceleration defined as:

$$\boldsymbol{B}_P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

In (4.11), the Centrifugal and Coriolis terms can be omitted due to the small movement of the CoM. The inequality friction constraint is obtained by substituting (4.11) into (4.10). $\ddot{\boldsymbol{p}}_{foot}$ is calculated using:

$$\ddot{\boldsymbol{p}}_{foot} = \dot{\boldsymbol{J}}^* \dot{\boldsymbol{\phi}} + \boldsymbol{J}^* \ddot{\boldsymbol{\phi}}, \tag{4.12}$$

where the generalized Jacobian matrix from the waist to the foot is indicated as $\boldsymbol{J}^* \in \Re^{6\times n}$ is [76].

Joint torque limitation

Joint servo is modeled as a simple proportional controller and it is obtained by

$$\boldsymbol{\tau} = \boldsymbol{P}(\boldsymbol{\phi}_{ref} - \boldsymbol{\phi}), \tag{4.13}$$

where the joint torque vector and the joint servo gain matrix are indicated as $\boldsymbol{\tau}$ and $\boldsymbol{P}$, respectively. The reference and present joint vectors are indicated as $\boldsymbol{\phi}_{ref}$ and $\boldsymbol{\phi}$, respectively. The joint angle limitation for limiting the torque is obtained by:

$$\boldsymbol{\phi}_i - \frac{\boldsymbol{\tau}_{\max}}{\boldsymbol{P}} < \boldsymbol{\phi}_{i+1,ref} < \boldsymbol{\phi}_i + \frac{\boldsymbol{\tau}_{\max}}{\boldsymbol{P}}, \tag{4.14}$$

where the joint vector at the i-th step is denoted as $\boldsymbol{\phi}_i$.

The joint torque limitation $\boldsymbol{\phi}_{\tau,\min/\max}$ and also the joint structural limitation $\boldsymbol{\phi}_{p,\min/\max}$ limit the joint vector. Consequently, the joint limits are calculated as

$$\begin{aligned} \boldsymbol{\phi}_{\min} &= \max[\boldsymbol{\phi}_{\tau,\min},\ \boldsymbol{\phi}_{p,\min}] \\ \boldsymbol{\phi}_{\max} &= \min[\boldsymbol{\phi}_{\tau,\max},\ \boldsymbol{\phi}_{p,\max}]. \end{aligned} \tag{4.15}$$

Since the joint acceleration is the design variable, a QP solver cannot directly deal with both joint structural limitation and joint torque limitation. Hence,

Table 4.1: Simulation parameters

Parameter	Value
Mass of massive object	200 kg
Virtual mass of the hand and foot	4000 kg
Coefficient of static friction	0.8
Coefficient of kinetic friction	0.6

the joint structural/torque limitation is implicitly considered by converting the joint limitation into the joint acceleration limitation. The following inequality constraint [77] provides the joint acceleration limitation:

$$\ddot{\boldsymbol{\phi}}^{\min} < \ddot{\boldsymbol{\phi}} < \ddot{\boldsymbol{\phi}}^{\max} \tag{4.16}$$

$$\begin{aligned} \ddot{\boldsymbol{\phi}}^{\min} &= \max[\mu_p(\boldsymbol{\phi}_{\min} - \boldsymbol{\phi}),\ \mu_v(\dot{\boldsymbol{\phi}}_{\min} - \dot{\boldsymbol{\phi}})] \\ \ddot{\boldsymbol{\phi}}^{\max} &= \min[\mu_p(\boldsymbol{\phi}_{\max} - \boldsymbol{\phi}),\ \mu_v(\dot{\boldsymbol{\phi}}_{\max} - \dot{\boldsymbol{\phi}})] \end{aligned}$$

where $\mu_p > 0$ and $\mu_v > 0$ indicate the gain parameters, $\ddot{\boldsymbol{\phi}}^{\min}$ and $\ddot{\boldsymbol{\phi}}^{\max}$ indicate the minimum and maximum limitations of the joint angular acceleration, respectively. In every loop of the simulation, $\ddot{\boldsymbol{\phi}}^{\min}$ and $\ddot{\boldsymbol{\phi}}^{\max}$ are updated. A whole simulation loop is demonstrated in Fig. 4.3.

4.3 Proposed method comparison with a PD controller

4.3.1 PD controller

By performing dynamic simulations, the proposed motion generation method using QP solver (which is abbreviated as QP) is verified compared to a conventional PD controller (which is abbreviated as QP). A PD controller described as the following control law is employed in the simulations.

$$\ddot{\boldsymbol{\phi}}_d = \boldsymbol{J}_{FtH}^{-1}(\ddot{\boldsymbol{p}}_d - \dot{\boldsymbol{J}}_{FtH}\dot{\boldsymbol{\phi}}), \tag{4.17}$$

as in (4.4), $\ddot{\boldsymbol{p}}_d$ can be calculated. Noting that the humanoid robot HRP-2 receives the joint command as the input, the obtained $\ddot{\boldsymbol{\phi}}_d$ is integrated twice and given as the joint angle command to the humanoid robot.

The parameters presented in Table 4.2 are employed in both QP and PD simulations. In both QP and PD simulations, the same hand reference position $\boldsymbol{p}_r$ is used in the calculation of $\ddot{\boldsymbol{p}}_d$ (given by (4.4)). The hand refrence position is given by:

$$\begin{aligned} \boldsymbol{p}_r &= \boldsymbol{p}_{hand,d} + \boldsymbol{p}_{hand,off} \\ \boldsymbol{p}_{hand,d} &= FK(\boldsymbol{\phi}_d), \end{aligned}$$

where $\boldsymbol{p}_{hand,off}$ denotes the offset of the hand reference position, which is given in Table 4.2. In the OpenHRP simulator, the kinetic friction coefficient given in Table 4.2 was applied in the simulations. $FK()$ is a function used to solve the forward kinematics. Hence, by solving the forward-kinematics with the joint command $\boldsymbol{\phi}_d$, the desired hand position $\boldsymbol{p}_{hand,d}$ can be calculated.

Table 4.2: Parameters of the QP and PD simulations

	Value
Massive object mass	200 kg
Virtual mass of hands	4000 kg
Virtual mass of feet $\boldsymbol{M}_{foot(vir)}$	4000 kg
Static friction coefficient μ	0.8
Kinetic friction coefficient	0.6
Reference hand position offset $\boldsymbol{p}_{hand,off}$	$\begin{bmatrix} 0.025\ \text{(m)} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$

4.3.2 Comparison of the transported distance of the massive object

An overview of the QP and PD simulations results are demonstrated in Figure 4.4.

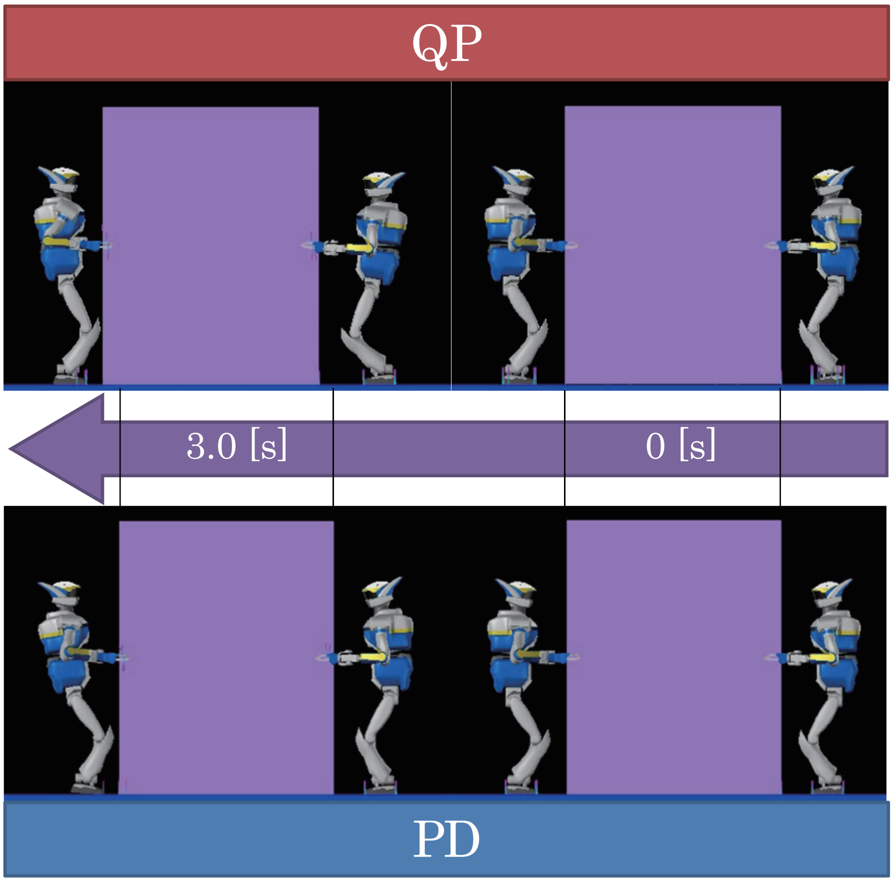


Figure 4.4: QP and PD simulations.

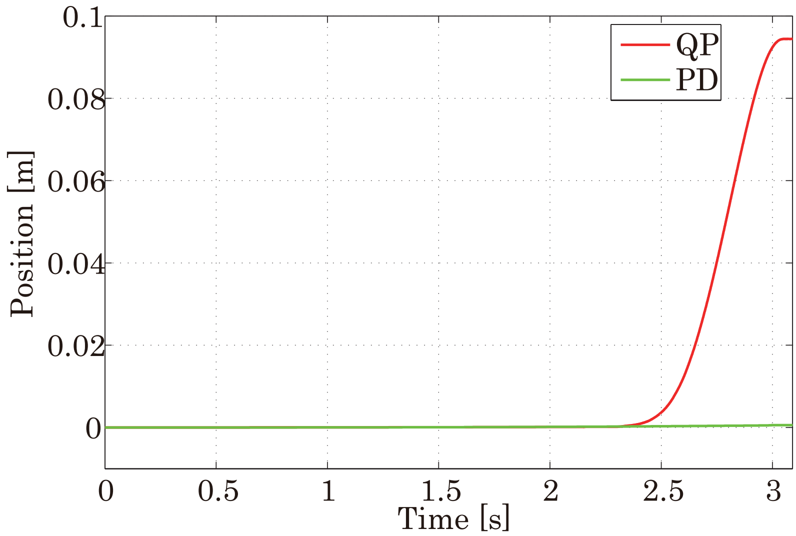


Figure 4.5: Transported distance of the massive object (QP vs. PD).

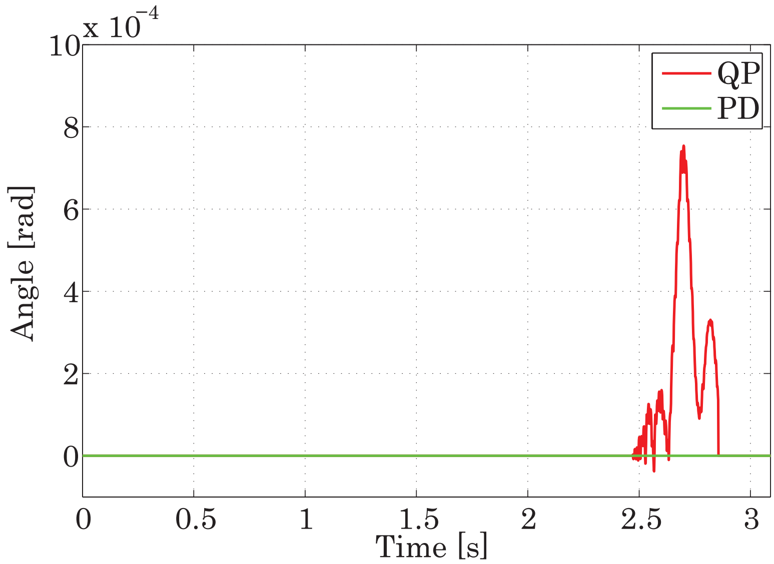


Figure 4.6: Orientation of the massive object during transportation (QP vs. PD).

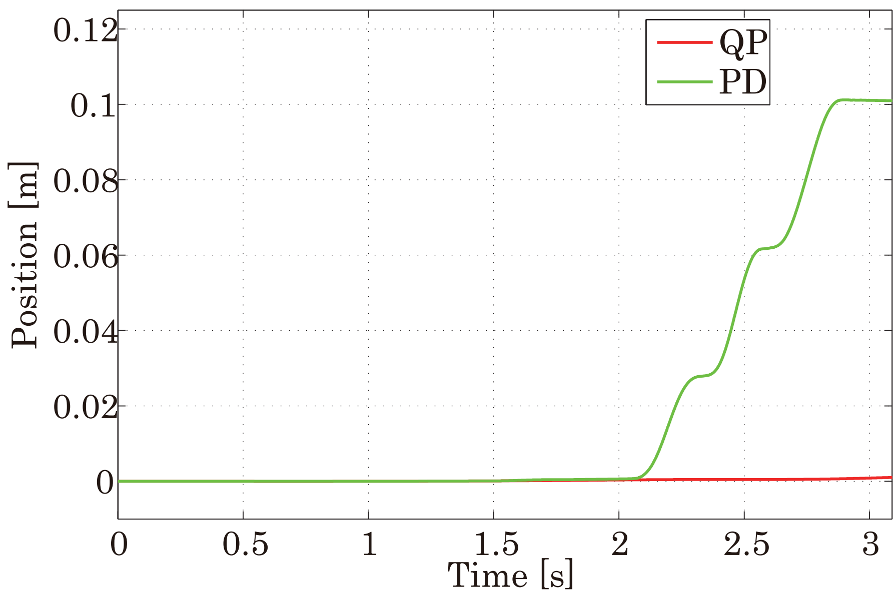


Figure 4.7: Comparison of the slipping distance of the pulling side robot (right foot) using QP and PD (x-axis).

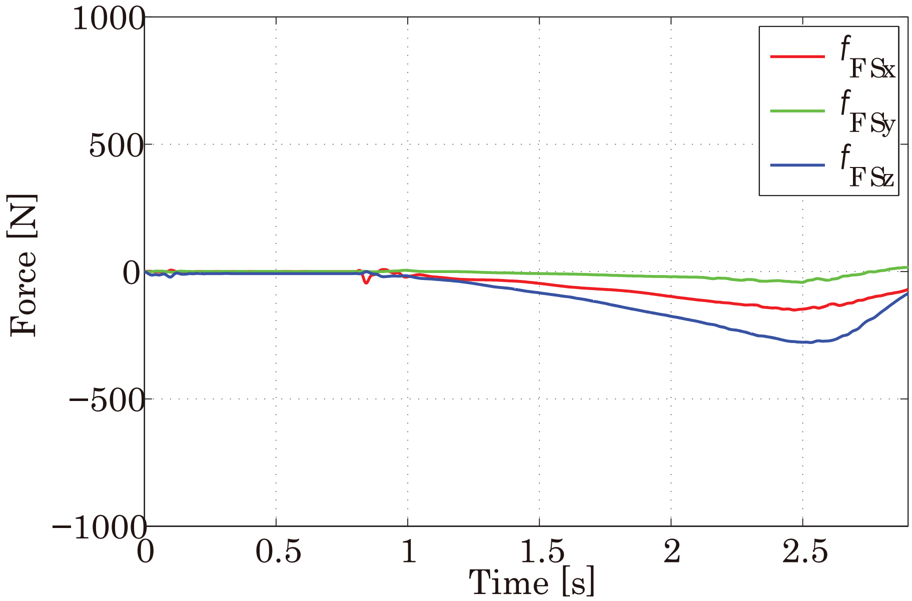


Figure 4.8: Generated forces at the right hand of the pushing robot (QP).

Using the proposed method (QP), Fig. 4.5 demonstrate how the robots transported the 200 kg massive object roughly 0.1 m.); on the other hand, using the conventional PD controller (PD), since the pulling robot slipped, the robots could not relocate the object (see Fig. 4.7). Fig. 4.6 indicates that during transportation, the object did not rotate significantly. The maximum tilting angle was about 0.0007 rad (0.04 °).

4.3.3 Comparison of the generated forces at the hands

The simulations were implemented in a 3D space. Since there were no notable differences between the forces generated at the right and left hand/foot, only the results of the right hand/foot are presented in the following subsections. The force sensor data of the right hands of both pushing and pulling robots are demonstrated in Figures 4.8 $\sim$ 4.11, respectively, using the QP and PD methods in the simulations. The object is pushed/pulled in the positive direction of the x-axis; hence, in Figs. 4.8 $\sim$ 4.11, the forces of the wrist force sensor along the x-axis are reported as negative. In both pushing/pulling phases, the forces generated at hands along the z-axis are relatively significant. There are two main reasons for

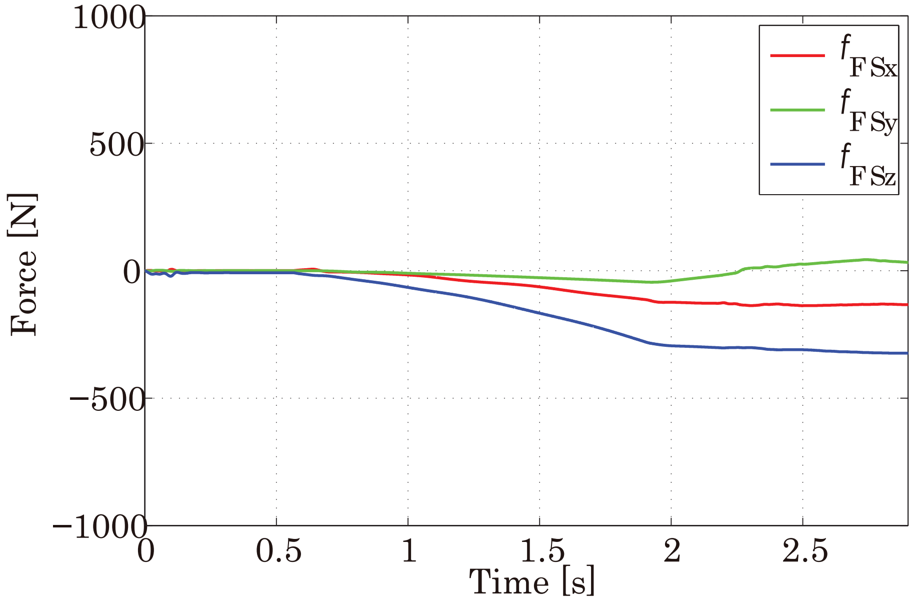


Figure 4.9: Generated forces at the right hand of the pushing robot (PD).

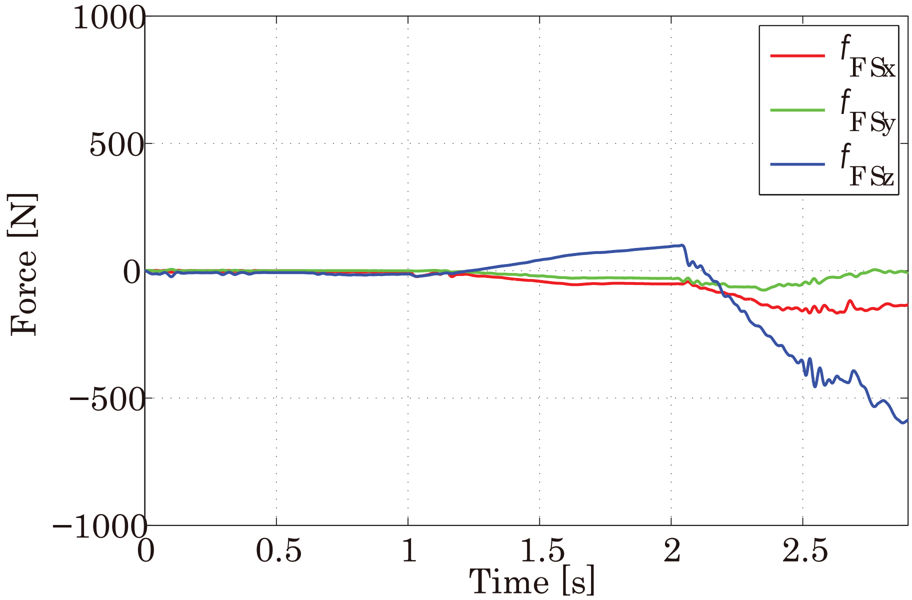


Figure 4.10: Generated forces at the right hand of the pulling robot (QP).

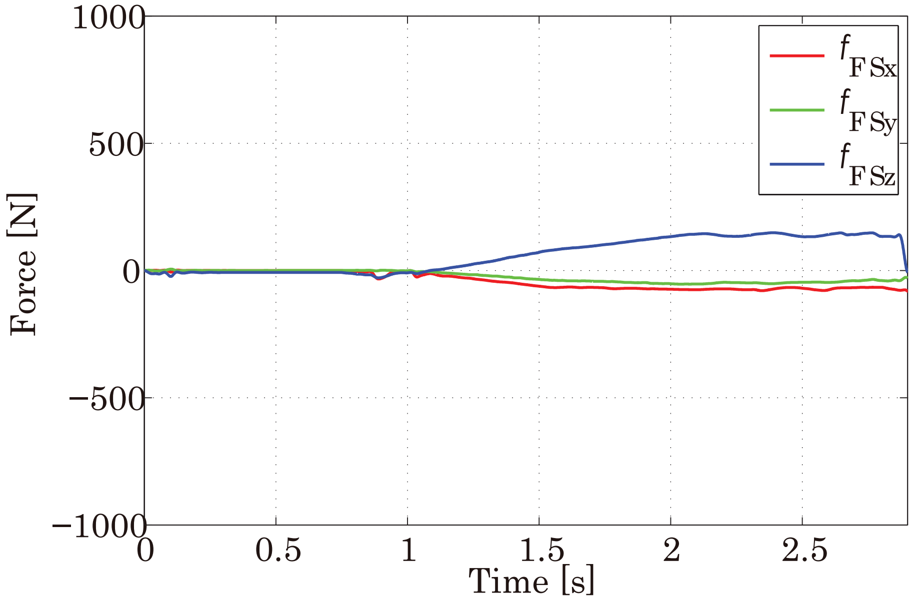


Figure 4.11: Generated forces at the right hand of the pulling robot (PD).

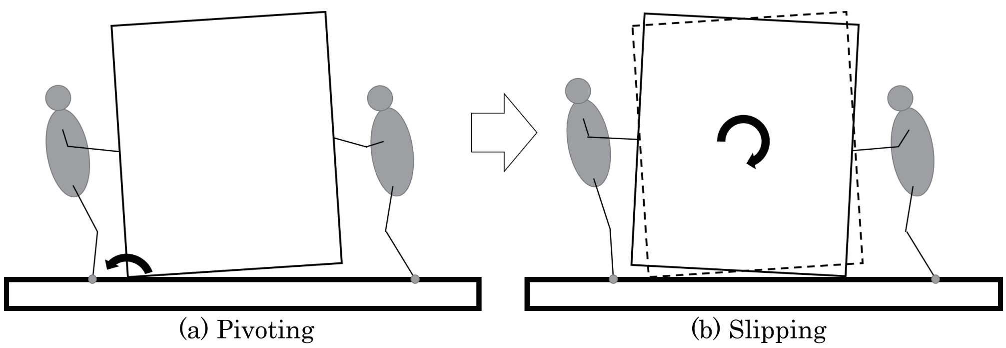


Figure 4.12: Rotational changes of the massive object.

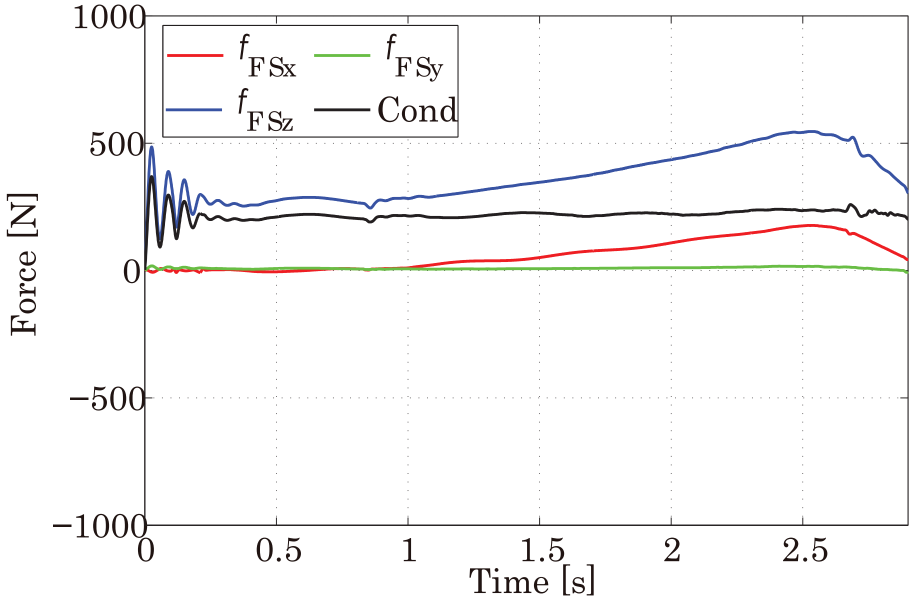


Figure 4.13: Acting forces on the right foot of the pushing robot (QP).

the cause of this force along the z-axis. Firstly, as shown in Fig. 4.12 (a), due to the frictional force between the ground and the massive object, the object tends to pivot about the bottom edge of the pulling side. Secondly, since the robot hands and the massive object are fully constrained, due to the rotation of the object, the pulling robot tends to get tilted too.

Fig. 4.10 and Fig. 4.11 demonstrate significant differences between the generated forces at the right hand of the pulling robot using QP and PD. While using QP, the object suddenly started to move at about the 2 seconds. Before the object started to move, it first rotated slightly (Fig. 4.12 (a)), and then it began to move along the x-axis, while rotating backward (Fig. 4.12 (b)).

4.3.4 Comparison of forces generated at the feet

The ground reaction forces acting on the right foot of the pushing and pulling humanoid robots are demonstrated in Figures 4.13 $\sim$ 4.16, respectively. In the figures, the following index of the non-slip condition is also demonstrated:

$$\mathrm{Cond} = \mu f_{FSz} - \sqrt{f_{FSx}^2 + f_{FSy}^2}, \tag{4.18}$$

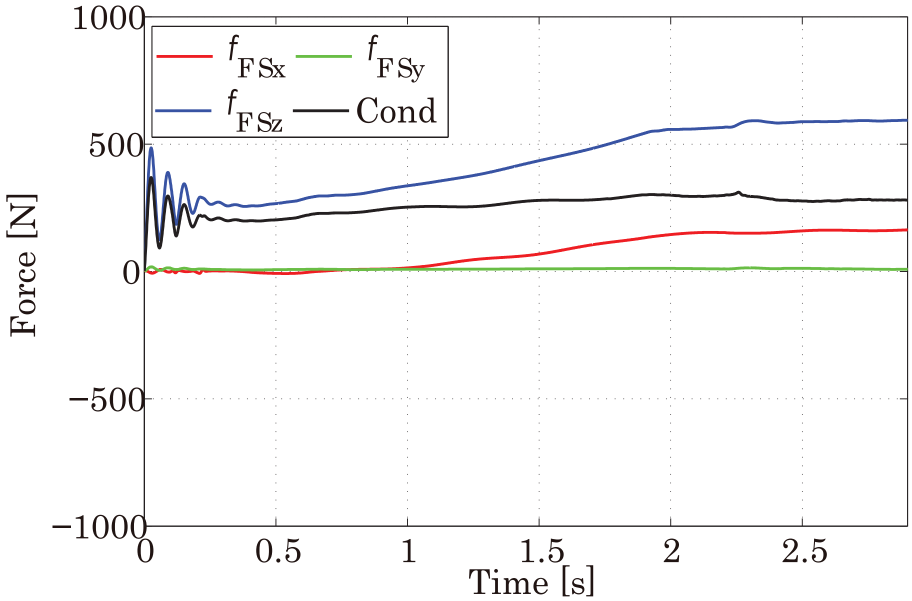


Figure 4.14: Acting forces on the right foot of the pushing robot (PD).

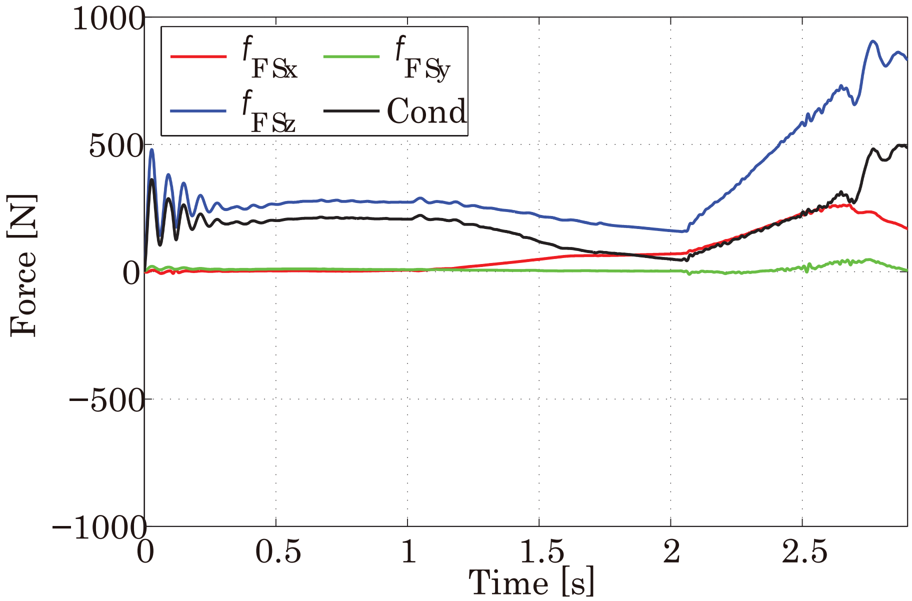


Figure 4.15: Acting forces on the right foot of the Pulling robot (QP).

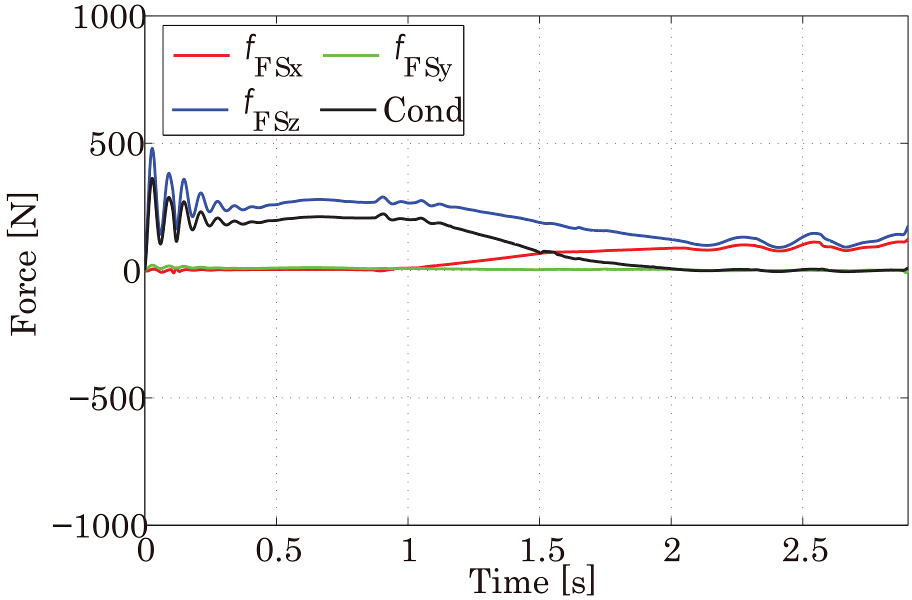


Figure 4.16: Acting forces on the right foot of the Pulling robot (PD).

where the force sensor data of each axis are denoted as f_{FSx}, f_{FSy}, and f_{FSz}. The non-slip condition is satisfied if the index *Cond* is above zero.

During the object transportation using QP, the non-slip indices *Cond* of the pushing/pulling robots were always positive; hence, the non-slip condition was satisfied (see Fig. 4.13 and Fig. 4.15). In contrast, during the object transportation using PD, the non-slip index of the pulling robot reached almost zero after 2 seconds (see Fig. 4.16); hence, the pulling robot's feet slipped, and the massive object could not be transported appropriately.

4.3.5 Comparison of joint torques

The joint torques are normalized using the maximum joint torques of the humanoid robots, shown in Figures 4.17 $\sim$ 4.24.

The resultant joint torques do not surpass the maximum torques since the joint angle commands are bounded in both QP ((4.14) $\sim$ (4.16) and PD simulations (4.14). Hence, the normalized joint torques never exceed ± 1 during the simulations.

In Fig. 4.17, the normalized joint torques are minimized autonomously since

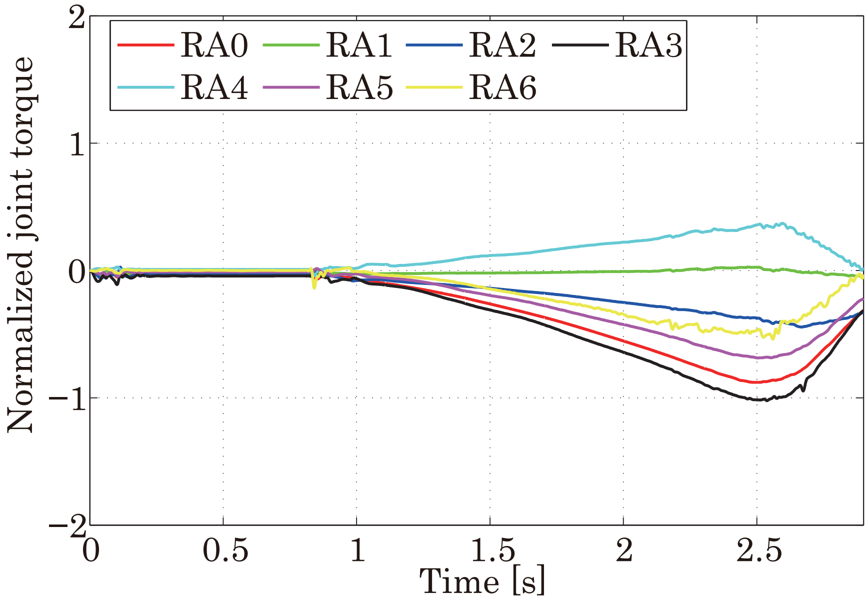


Figure 4.17: Normalized joint torques of the pushing robot's right arm (QP).

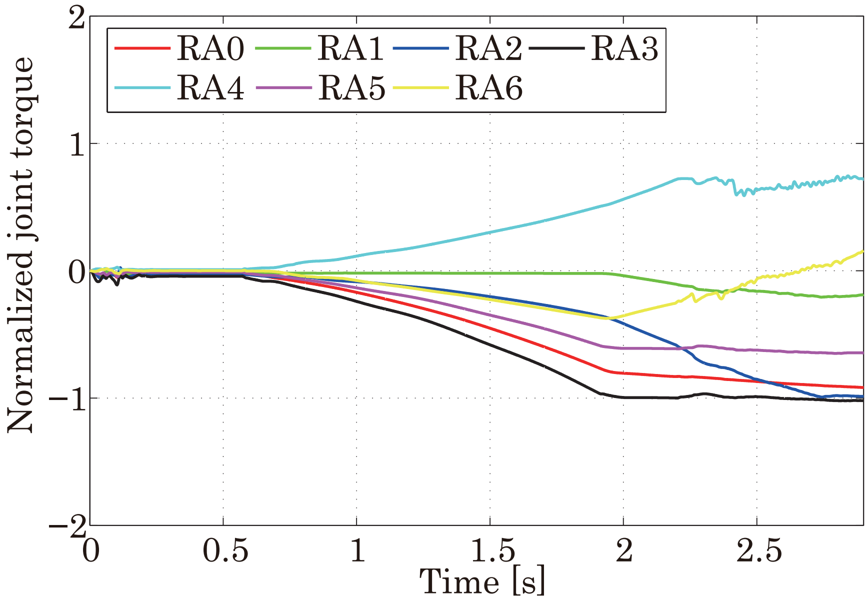


Figure 4.18: Normalized joint torques of the pushing robot's right arm (PD).

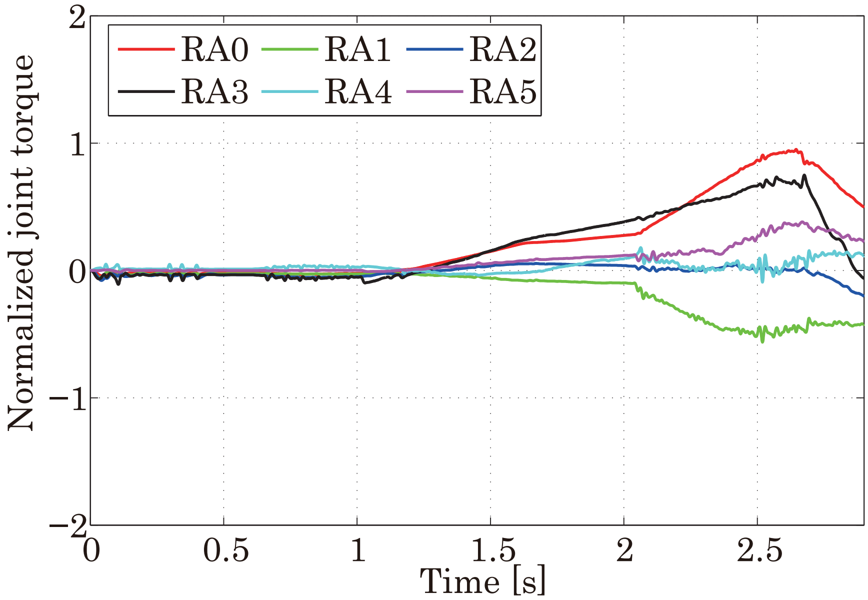


Figure 4.19: Normalized joint torques of the pulling robot's right arm (QP).

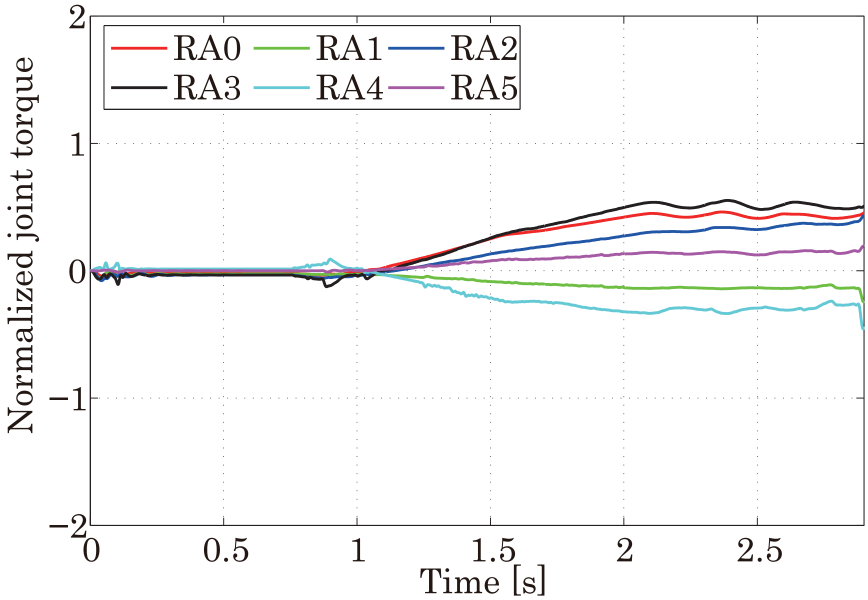


Figure 4.20: Normalized joint torques of the pulling robot's right arm (PD).

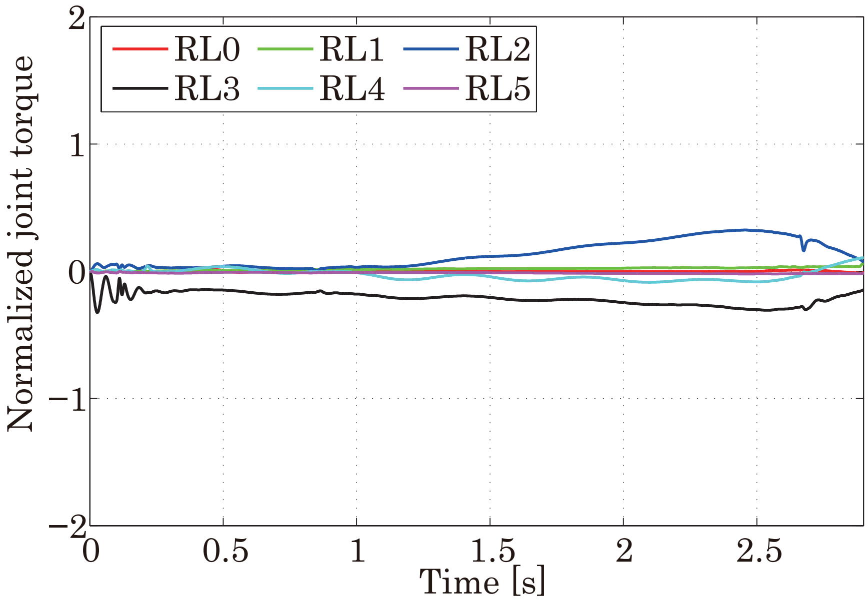


Figure 4.21: Normalized joint torques of the pushing robot's right leg (QP).

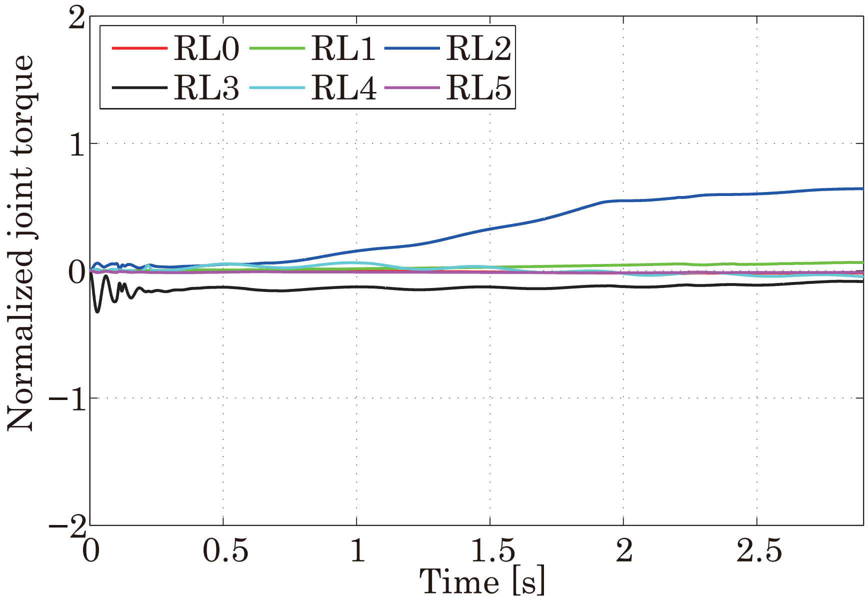


Figure 4.22: Normalized joint torques of the pushing robot's right leg (PD).

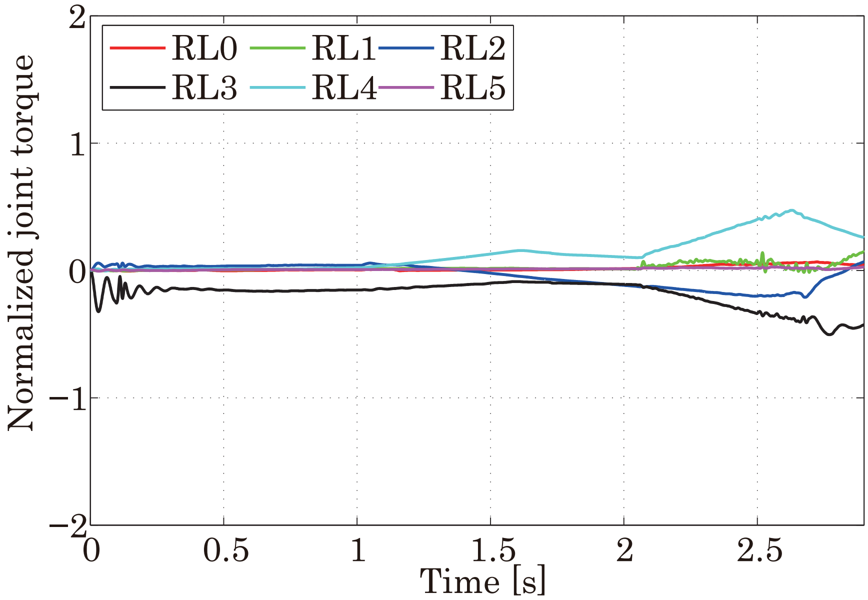


Figure 4.23: Normalized joint torques of the pulling robot's right leg (QP).

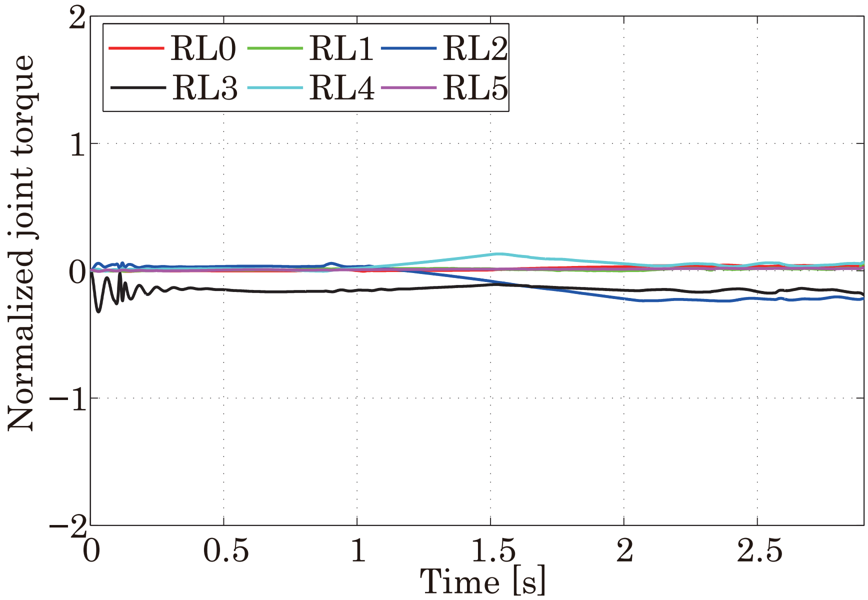


Figure 4.24: Normalized joint torques of the pulling robot's right leg (PD).

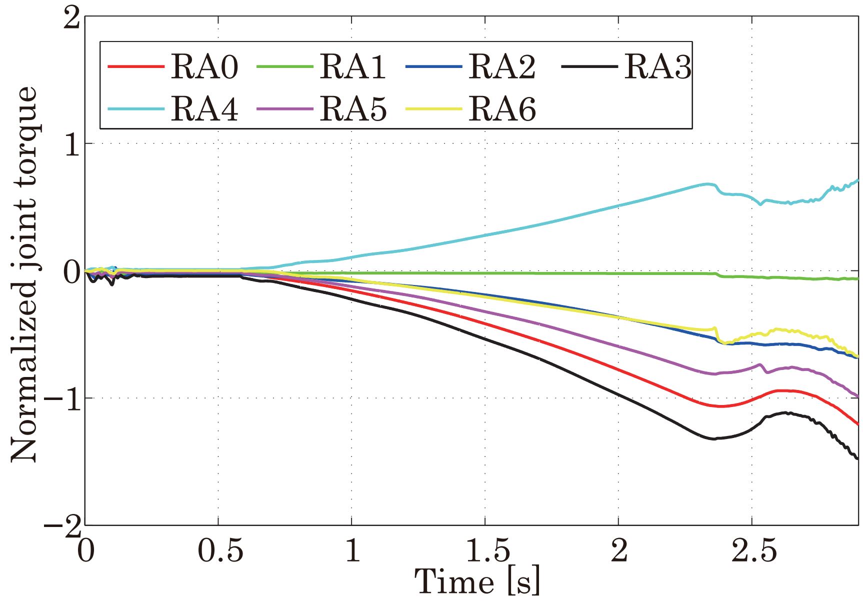


Figure 4.25: Normalized joint torques of the pushing robot's right arm (PD without torque limit).

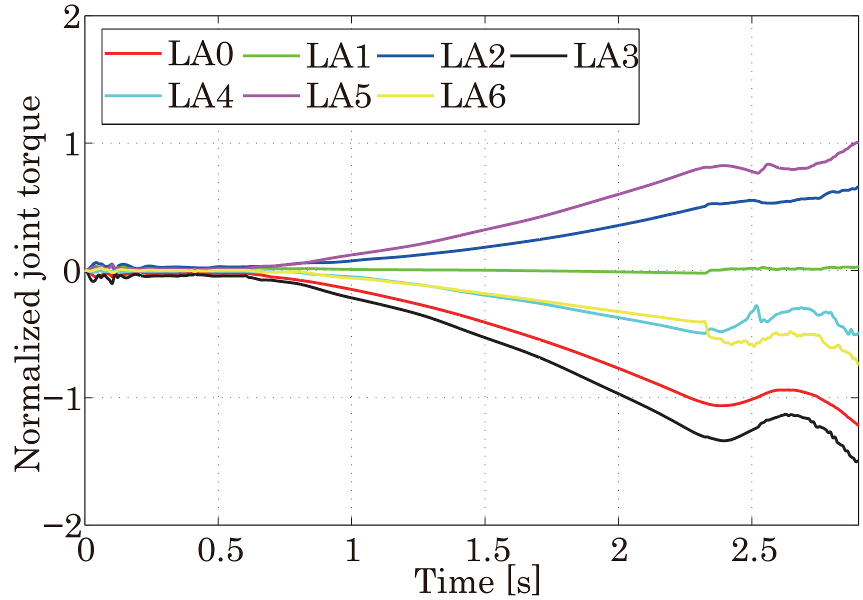


Figure 4.26: Normalized joint torques of the pushing robot's left arm (PD without torque limit).

the joint torques objective function (4.3) is optimized in the QP simulation.

However, as shown in Fig. 4.18, the normalized joint torques of the pushing robot becomes saturated in PD simulation. The shortage of the pushing robot's hands forces is caused due to this saturation of joint torques. The pulling robot's feet slip due to the shortage of pushing hand force. Hence, the massive object transportation cannot be achieved appropriately using the PD.

Worth noting that some of the normalized joint torques in the PD simulation would exceed one if there were no joint torque limits (see Fig. 4.25 and Fig. 4.26).

4.4 Summary

In this chapter, we proposed a method to generate motions for two humanoid robots in order to transport a massive object. A quadratic programming problem with inequality/equality constraints was solved to obtain the joint trajectory (acceleration level) of a multi-degree of freedom system. A free-floating model is adopted in the proposed motion generation method. Virtual masses are attached at the end of limbs to efficiently calculate the force constraints generated at the end of the limbs.

In order to validate the effectiveness of the proposed method, by performing the dynamic simulation, we compared our method results with the conventional PD controller. The results indicate that the generated motion to transport a 200kg object using the proposed method meets the constraints, such as the prevention of slip of the robots' feet and the avoidance of excessive joint torque. In contrast, the pulling robot's feet slipped during the transportation using the conventional PD controller, which indicates the advantage of our method. The results confirmed the effectiveness of our method over the conventional PD control method in massive object transportation using humanoid robots.

Chapter 5 Sequential Massive Object Transportation by Two Humanoid Robots

5.1 Introduction

In chapter 4, the generated motion in two humanoid robots was verified in a massive object transportation task without stepping. Therefore, the massive object transportation is limited due to the object's contact with the pulling robot's body (especially the robot's knee). In this chapter, the stepping/posture recovering motion is generated following the pushing/pulling action in order to enable the massive object transportation continuously. By performing an intermittent pushing/pulling simulation of a 200 kg massive object using two humanoid robots, the proposed method was verified. In order to demonstrate the effectiveness of the proposed method, the robots' force sensor data, the normalized joint torques data, and the carried distance of the massive object are presented.

The simulations of massive object transportation using QP and PD and QP without minimizing joint torques objective function (QP (w/o JT)) are compared related to the mass variations of the transported object. The results indicate the advantage of the proposed method (QP) and also the benefit of employing the joint torque minimization objective function in the proposed method.

5.2 Motion generation for sequential massive object transportation by two humanoid robots

The previously generated motion in the two humanoids for pushing/pulling a massive object is used in the pushing/pulling phase.

In this section, the motion generation for stepping/recovering posture is discussed. After completing the pushing/pulling motion, the position of the robots' feet recovers to the initial position relative to the hands' position. On the robot side, there is not a specific way to measure the transported distance of the object; hence, in order to calculate the reference foot position, the robot's hand position is utilized. First, the reference position of each humanoid robots' feet are calculated relative to the robots' hand positions and given by:

$$\begin{aligned} \boldsymbol{p}_{foot,ref,A} &= \boldsymbol{B}_{ex}\boldsymbol{p}_{foot,A} + \boldsymbol{B}_{x}\boldsymbol{p}_{hand,A} - \boldsymbol{d}_{off,A} \\ \boldsymbol{p}_{foot,ref,B} &= \boldsymbol{B}_{ex}\boldsymbol{p}_{foot,B} + \boldsymbol{B}_{x}\boldsymbol{p}_{hand,B} + \boldsymbol{d}_{off,B} \end{aligned} \tag{5.1}$$

$$\boldsymbol{B}_{ex} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \boldsymbol{B}_{x} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{aligned} \boldsymbol{d}_{off,A} &= \begin{bmatrix} off_{x,A} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \\ \boldsymbol{d}_{off,B} &= \begin{bmatrix} off_{x,B} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T, \end{aligned} \tag{5.2}$$

where $\boldsymbol{d}_{off,A}$ and $\boldsymbol{d}_{off,B}$ denote the offset distance from the foot to the hand of the pushing and pulling robots, respectively. In Fig. 5.1, $off_{x,A}$ and $off_{x,B}$ indicate the distances between the foot and hand of the pushing and pulling robots along the x-axis, respectively.

The foot trajectory to the reference foot position is generated after calculating the foot reference positions. The robots' feet are displaced by order of right and left foot using the following equation.

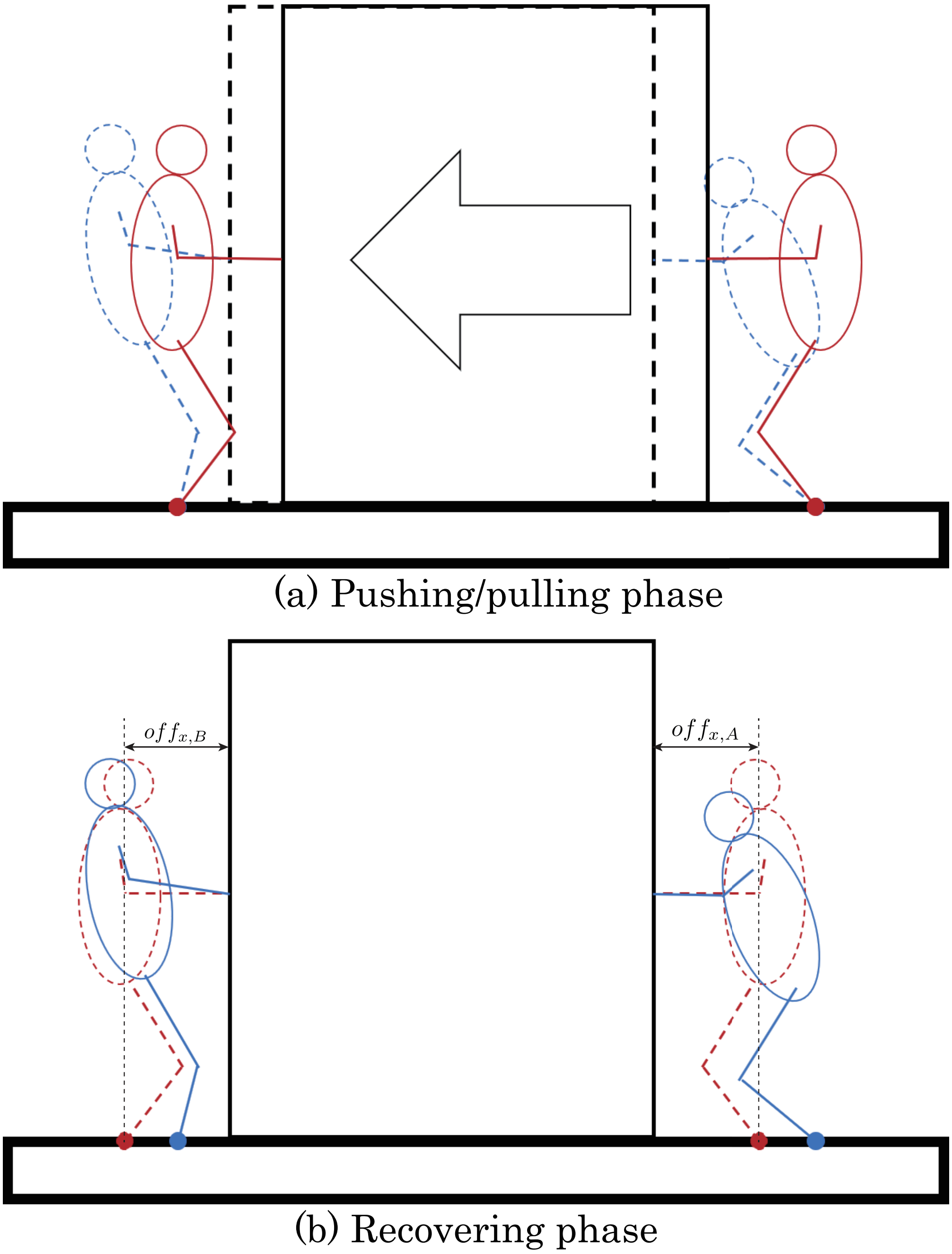


Figure 5.1: Intermittent pushing/pulling motion.

$$\dot{\boldsymbol{\phi}} = \boldsymbol{J}_{all}^{-1}\dot{\boldsymbol{p}}_{all}, \tag{5.3}$$

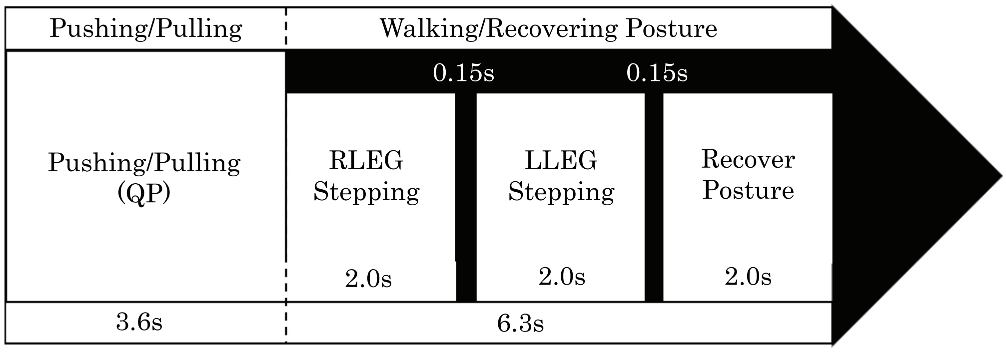


Figure 5.2: Motion sequence.

$$\boldsymbol{J}_{all} = \begin{bmatrix} \boldsymbol{J}_{WtRF} & & & 0 \\ & \boldsymbol{J}_{WtLF} & & \\ & & \boldsymbol{J}_{CtRH} & \\ 0 & & & \boldsymbol{J}_{CtLH} \end{bmatrix},$$

$$\dot{\boldsymbol{p}}_{all} = \begin{bmatrix} \dot{\boldsymbol{p}}_{foot,R} & \dot{\boldsymbol{p}}_{foot,L} & \dot{\boldsymbol{p}}_{hand,R} & \dot{\boldsymbol{p}}_{hand,L} \end{bmatrix}^T.$$

where $\boldsymbol{J}_{all}$ is the Jacobian matrix from waist to limbs. $\boldsymbol{J}_{WtRF}$ and $\boldsymbol{J}_{WtLF}$ indicate the Jacobian matrix from waist to the right and left foot respectively. $\boldsymbol{J}_{CtRH}$ and $\boldsymbol{J}_{CtLH}$ indicate the Jacobian matrix from chest to the right and left hand respectively. $\dot{\boldsymbol{p}}_{all}$ indicates the velocity vector of the limbs of the humanoid robot. When the left foot of the humanoid robot is in the support leg phase, $\dot{\boldsymbol{p}}_{foot,L}$, $\dot{\boldsymbol{p}}_{hand,R}$ and $\dot{\boldsymbol{p}}_{hand,L}$ are zero and only $\dot{\boldsymbol{p}}_{hand,R}$ is used in the trajectory calculation. Similarly, if the right foot of humanoid robot is in the support leg phase, $\dot{\boldsymbol{p}}_{foot,R}$, $\dot{\boldsymbol{p}}_{hand,R}$ and $\dot{\boldsymbol{p}}_{hand,L}$ are zero and only $\dot{\boldsymbol{p}}_{hand,L}$ is used in the trajectory calculation. Each of the support leg phases takes place in 2 seconds. After relocating the feet into the reference position, the posture of the humanoid robots are recovered to the initial stance using the following equation:

$$\boldsymbol{\phi}_{i+1,ref} = \boldsymbol{\phi}_i + \frac{(\boldsymbol{\phi}_{end} - \boldsymbol{\phi}_{start})\boldsymbol{t}_{sp}}{\boldsymbol{t}_d}, \tag{5.4}$$

where $\boldsymbol{\phi}_{start}$ and $\boldsymbol{\phi}_{end}$ indicate the robot joint angles of the initial and the final posture, respectively. $\boldsymbol{t}_{sp}$ and $\boldsymbol{t}_d$ indicate the loop time and the duration time of the recovering posture, respectively. During $\boldsymbol{t}_d$, (5.4) is calculated repeatly.

The time table of the whole generated motions is demonstrated in Figure 5.2. In this figure, the white part indicates the subphase, and the black part indicates the waiting phase. At this point, the pushing/pulling motion generation is completed, and the stepping/recovering posture phase starts. Three subphases are performed in the order of right leg step, left leg step, and recovering posture phase during the stepping/recovering posture phase. Each subphase takes 2 seconds. Between each subphase, there is a waiting time of 0.15 s.

5.3 Sequential massive object transportation simulation

Dynamic simulations of the intermittent pushing/pulling are performed to verify the effectiveness of the proposed method in this kind of sequential transportation. The same pushing/pulling motions are generated using the QP method of chapter 4.

5.3.1 Generated forces at hands

The wrist force sensor data of the robots in QP simulation are demonstrated in Fig. 5.3, 5.4 (pushing side) and 5.5, 5.6 (pushing side).

In the pushing robot, the force (x, y and z axis) tends to be zero at the end of each pushing phase. Due to the tilting of the pushing robot toward the stepping direction, the positive force (z axis) is generated at each foot during the stepping phase.

In the pulling robot, the force (x, y and z axis) is not zero at the end of each pushing phase. The humanoid robot in the pulling side is in an unstable situation at the end of the pulling phase. Therefore the hands forces need to be mitigated during the stepping/recovering phase. Figures. 5.5 and 5.6 demonstrate that the hands forces are around zero at the end of the stepping/recovering posture phase.

The first and second pushing/pulling phases indicate roughly the same forces act on the hands of both pushing/pulling robots.

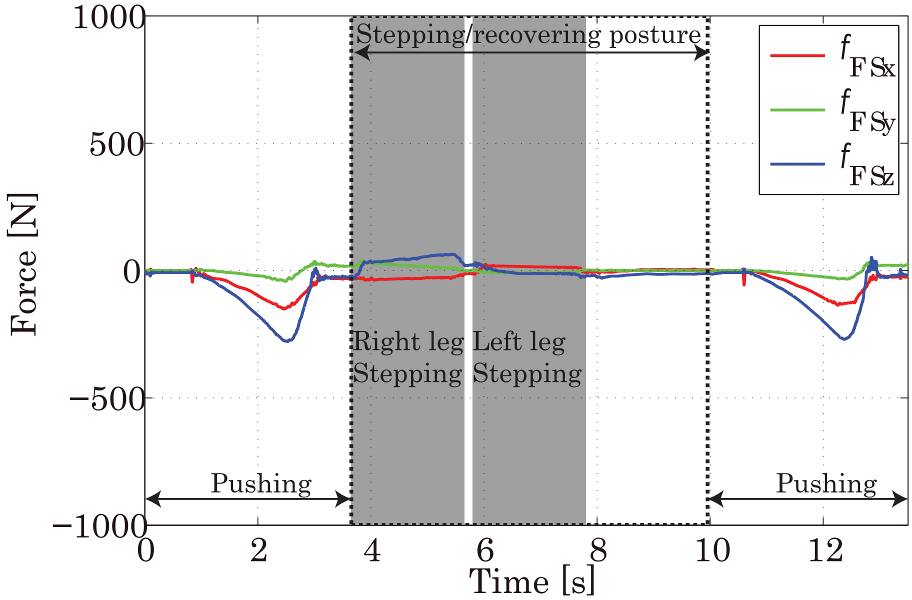


Figure 5.3: Right hand forces (push).

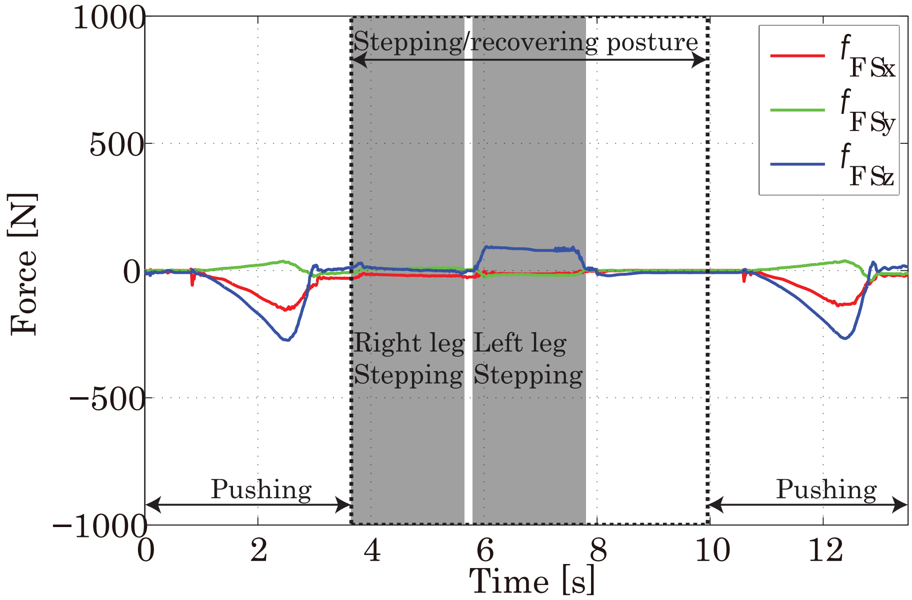


Figure 5.4: Left hand forces (push).

5.3.2 Generated forces at feet

The feet force sensor data in the QP simulation are demonstrated in Figs. 5.7, 5.8 (pushing side) and 5.9, 5.10 (pushing side).

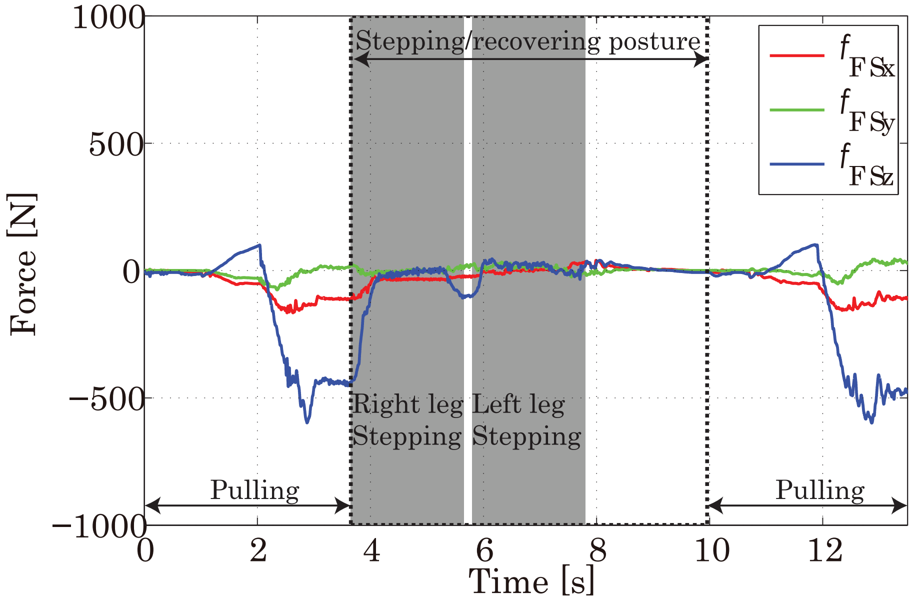


Figure 5.5: Right hand forces (pull).

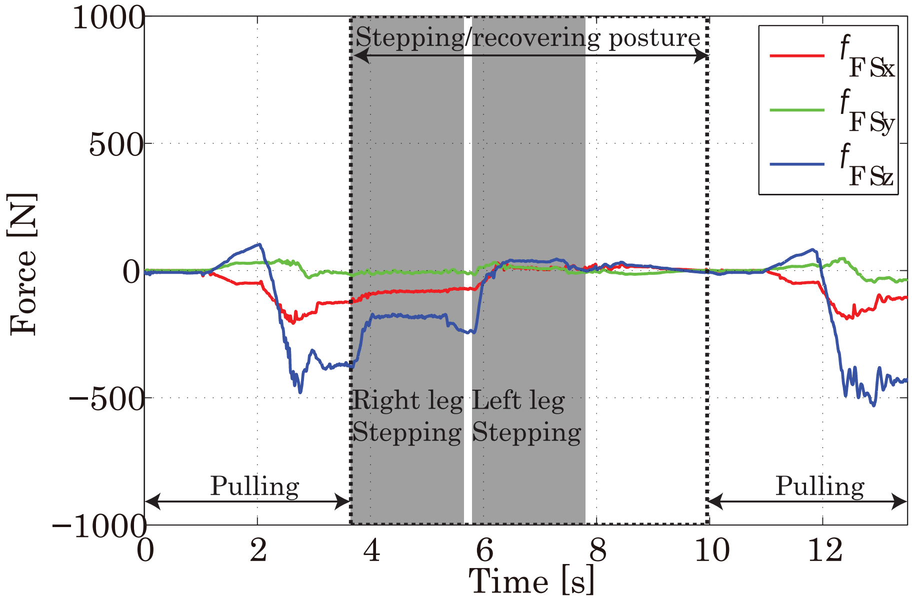


Figure 5.6: Left hand forces (pull).

At the end of each pushing phase, the forces of the pushing robot's feet (x, y and z axis) return to the initial amount (beginning of the pushing/pulling phase). The first and second pushing/pulling phases indicate approximately the same forces

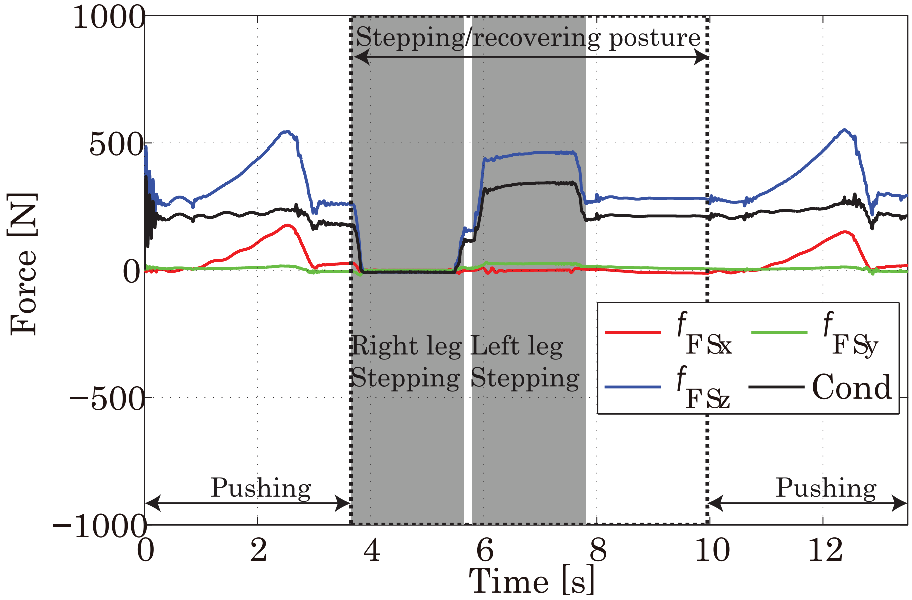


Figure 5.7: Right foot forces (push).

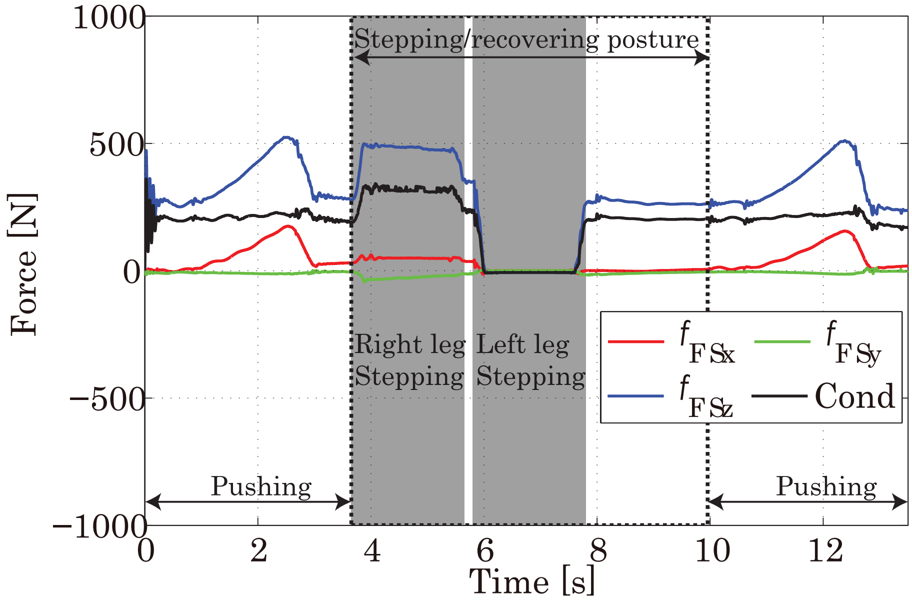


Figure 5.8: Left foot forces (push).

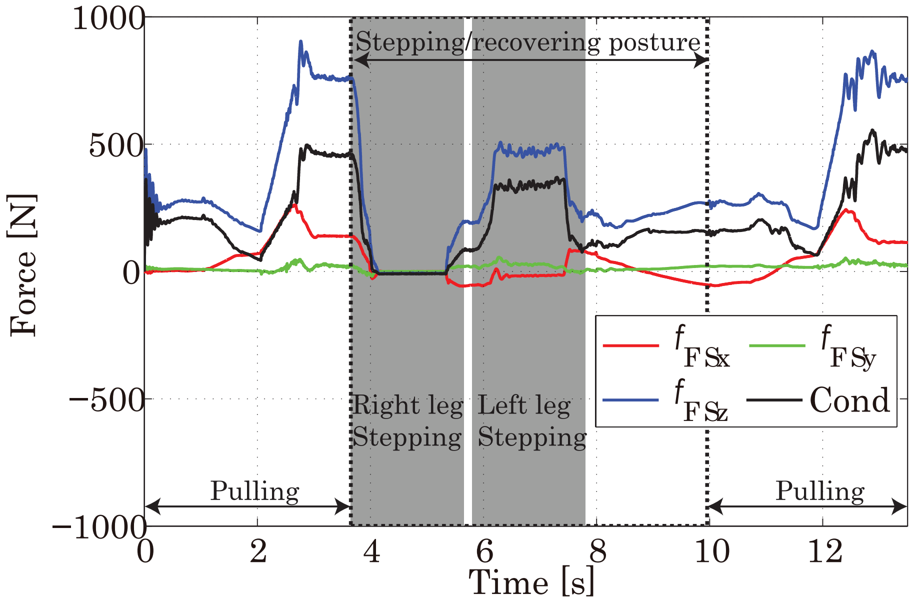


Figure 5.9: Right foot forces (pull).

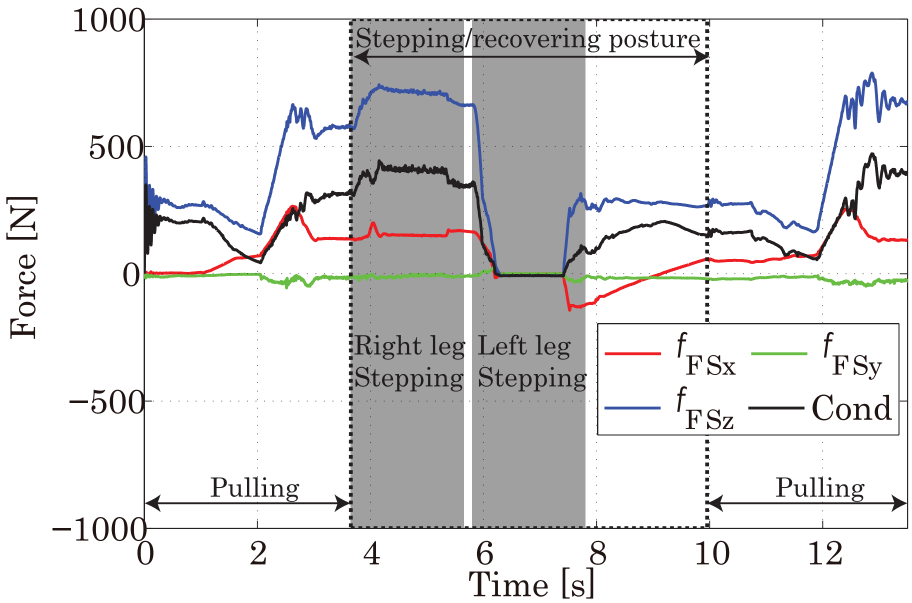


Figure 5.10: Left foot forces (pull).

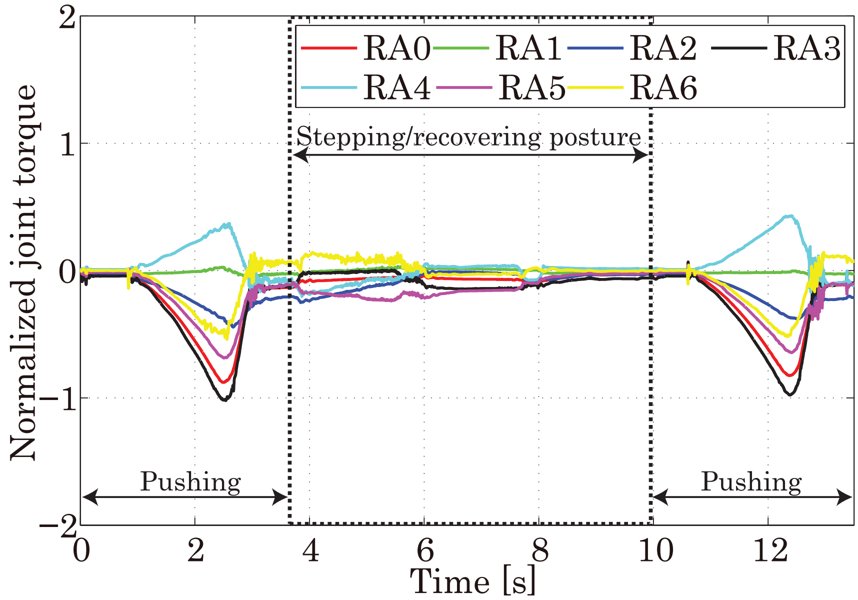


Figure 5.11: Right arm normalized joint torques (push).

act on each pushing robot's foot.

At the end of each pulling phase, the forces of the pulling robot's feet (x, y and z axis) do not return to the initial amount; however, these forces will mitigate at the recovering posture phase. Although the forces are not fully mitigated at the end of the posture recovering phase, approximately the same patterns of the forces occur in the second pulling phase.

The gray scale part of Figs. 5.7, 5.8 (pushing side) and 5.9, 5.10 demonstrate that the non-slip index is going to zero at each stepping phase of the swinging leg.

5.3.3 Normalized joint torques

The normalized joint torques data in the QP simulation are demonstrated in Figs. 5.11 $\sim$ 5.14 (pushing side) and 5.15 $\sim$ 5.18 (pulling side).

At the end of each pushing phase, the normalized joint torques of the pushing robot return to the initial amount (beginning of the pushing phase).

At the end of each pulling phase, the normalized joint torques of the pulling robot's arms return to the initial amount. However, the normalized joint torques of the humanoid robot's legs do not return to the initial amount due to the incomplete

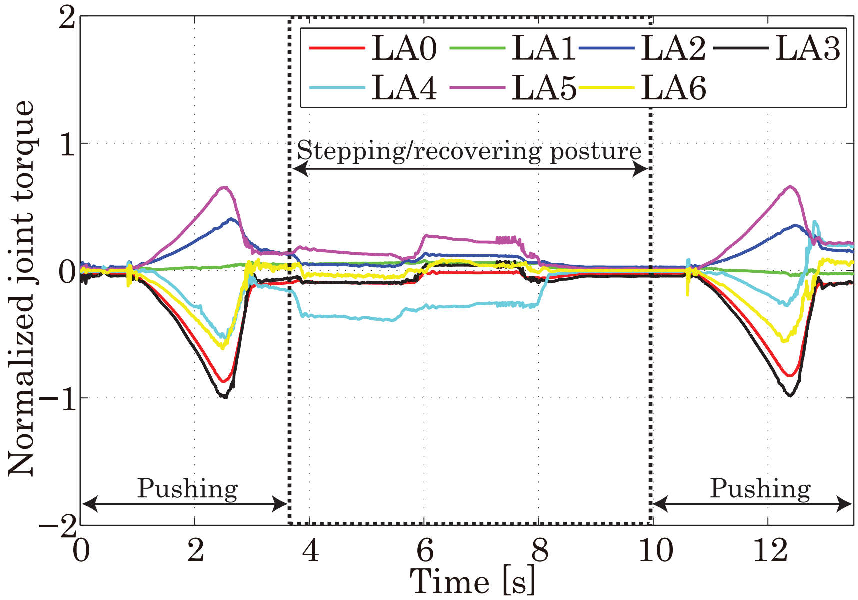


Figure 5.12: Left arm normalized joint torques (push).

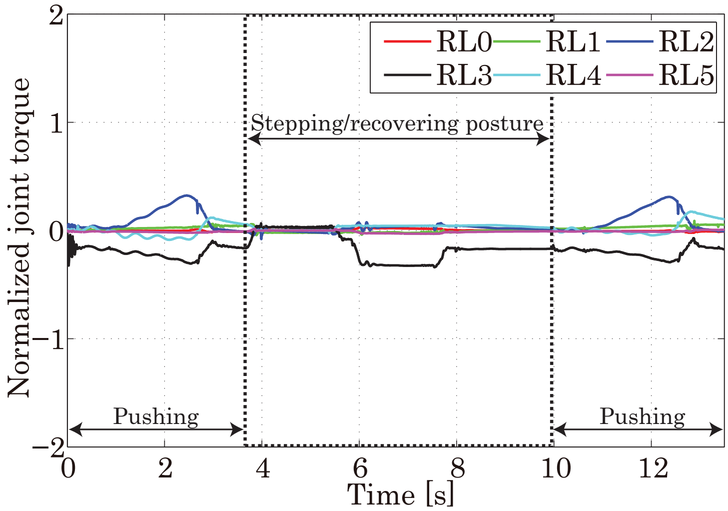


Figure 5.13: Right leg normalized joint torques (push).

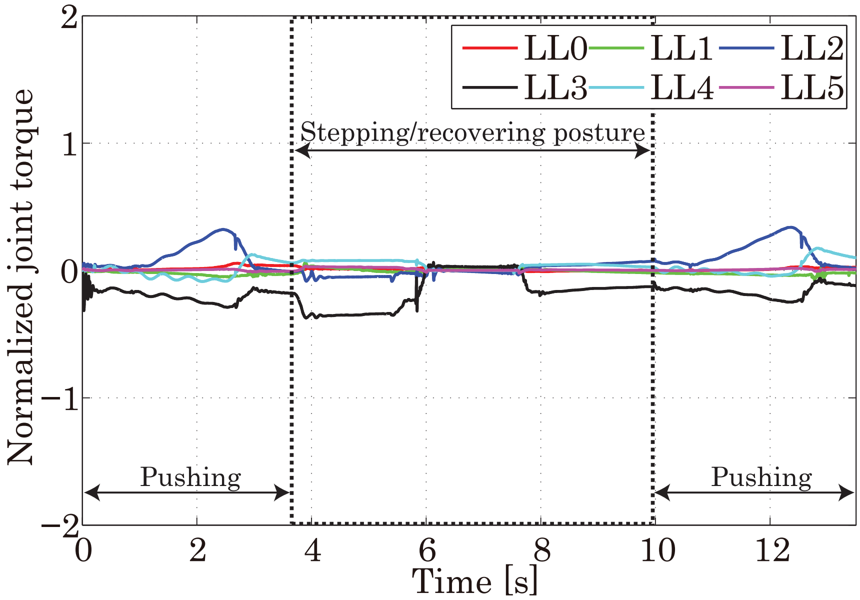


Figure 5.14: Left leg normalized joint torques (push).

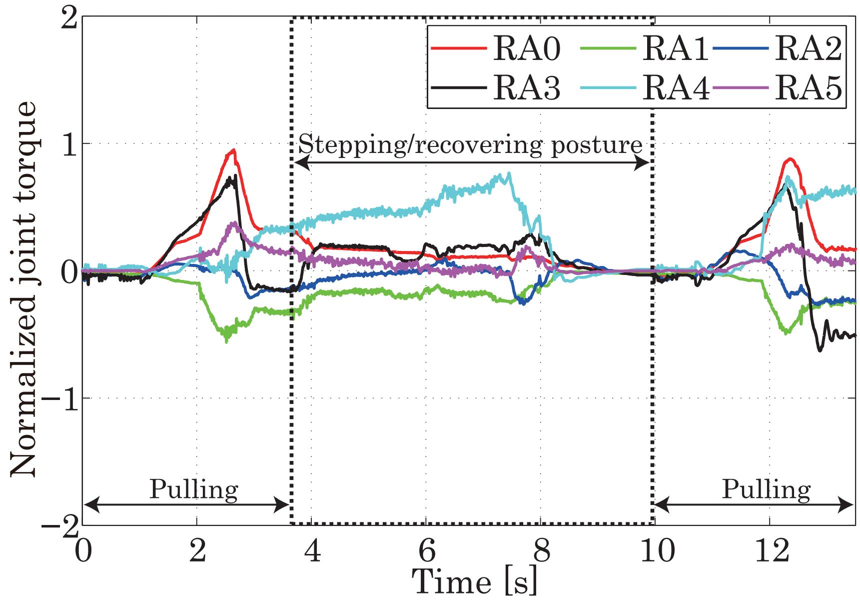


Figure 5.15: Right arm normalized joint torques (pull).

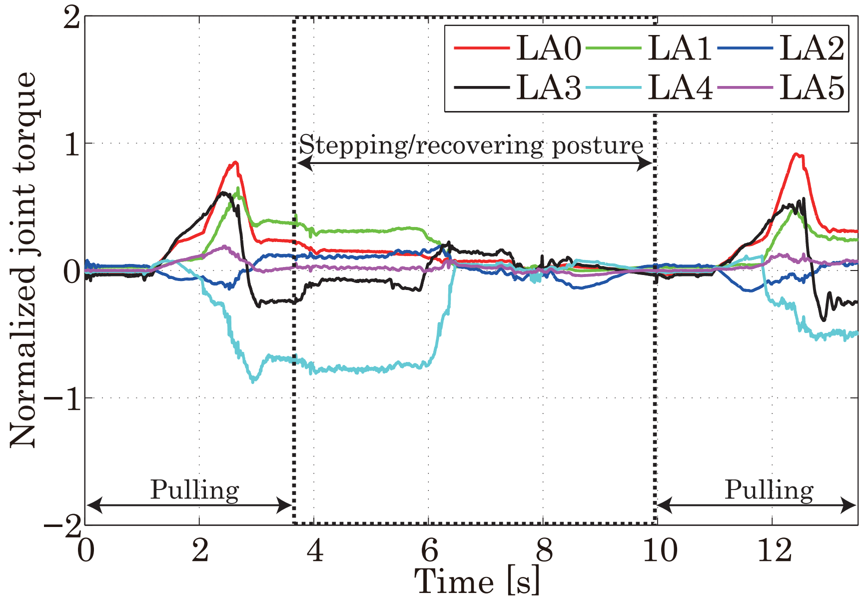


Figure 5.16: Left arm normalized joint torques (pull).

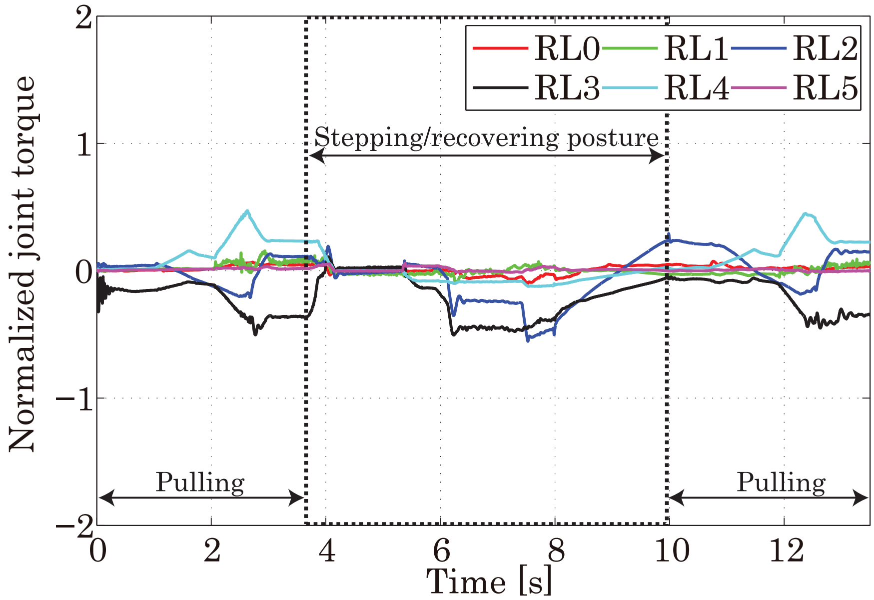


Figure 5.17: Right leg normalized joint torques (pull).

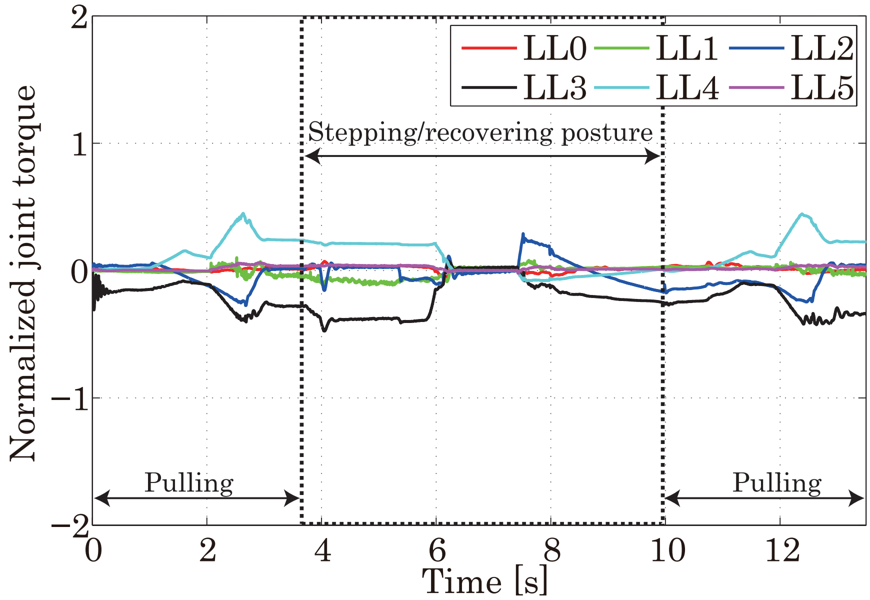


Figure 5.18: Left leg normalized joint torques (pull).

force mitigation of the robot's feet.

Furthermore, thanks to the maximum joint torques constraint, the normalized joint torques do not surpass the normalized joint torques value of 1.

5.3.4 Massive object transported distance and feet position error

The transported distance of the massive object along the x-axis using QP and PD is demonstrated in Figure 5.19. The massive object is carried about 0.2 m by performing the intermittent pushing/pulling motion twice using QP. The solid green line in Fig. 5.19 indicates that the robots could not move the object using PD. In Table 5.1, the distance errors at the end of the stepping/recovering posture phase are listed.

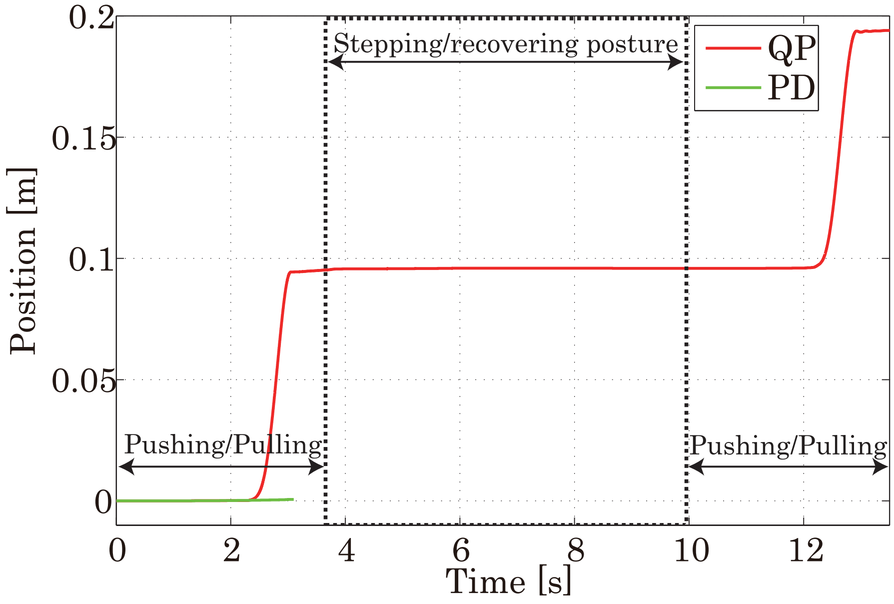


Figure 5.19: Massive object transported distance along x-axis.

Table 5.1: Feet position error (x-axis) at the end of stepping/recovering posture phase

	Pushing side	Pulling side
Distance from massive object to feet	0.0003 m	0.0005 m
Distance of each feet	0.0007 m	0.00056 m

5.4 Comparison of QP, PD, and QP (w/o JT) relative to variable masses of the massive object

In this section, the simulation results of QP, PD, and QP (without minimization of the joint torques objective function, w/o JT) using variable object masses are compared. Masses of 75 kg, 100 kg, 125 kg, 150 kg, 175 kg, and 200 kg are used as the massive object in the QP, PD, and QP (w/o JT) simulations. All simulations are implemented until the massive object reaches the pulling robot's body.

Transported distance/Slip distance of pulling foot [m]							
		75 kg	100 kg	125 kg	150 kg	175 kg	200 kg
QP	T distance	0.1422	0.1434	0.1386	0.1449	0.1357	0.1385
	Slip distance	0.0015	0.0018	0.0018	0.002	0.002	0.002
PD	T distance	0.155	0.1499	0.1246	0.102	0.0624	0.0597
	Slip distance	0.0045	0.009	0.0355	0.058	0.0977	0.0991
QP (w/o JT)	T distance	0.1441	0.1453	0.1378	0.1296	0.132	0.133
	Slip distance	0.0015	0.0016	0.0017	0.0021	0.0018	0.002

Figure 5.20: Transported distance of the massive object and slipped distance of the pulling robot feet along x-axis (relative to variable masses).

5.4.1 Comparison of the foot slipped distance

The values of the foot slipped distance of the pulling robot and the transported distance of the massive object are demonstrated in Figure 5.20. Note that the pushing robot's feet did not slip in all simulations.

The foot slipped distance is demonstrated in Fig 5.21 clearly. the pulling robot's feet slipped at a maximum length of 0.002 m, in both QP and QP (w/o JT) simulations. However, in the PD simulation, the pulling robot's feet slipped distance increases as the mass of the object increases.

The results indicate the advantage of the proposed method over PD.

5.4.2 Comparison of the transported distance of the massive object

The transported distance of the massive object is demonstrated in Fig 5.22. For all masses, The transported distance of the massive object in the QP and QP (w/o JT) simulations was about 0.145 m. However, in the PD simulation, due to the slipping of the pulling robot's feet, as the mass of the massive object increases the transported distance of the object decreases.

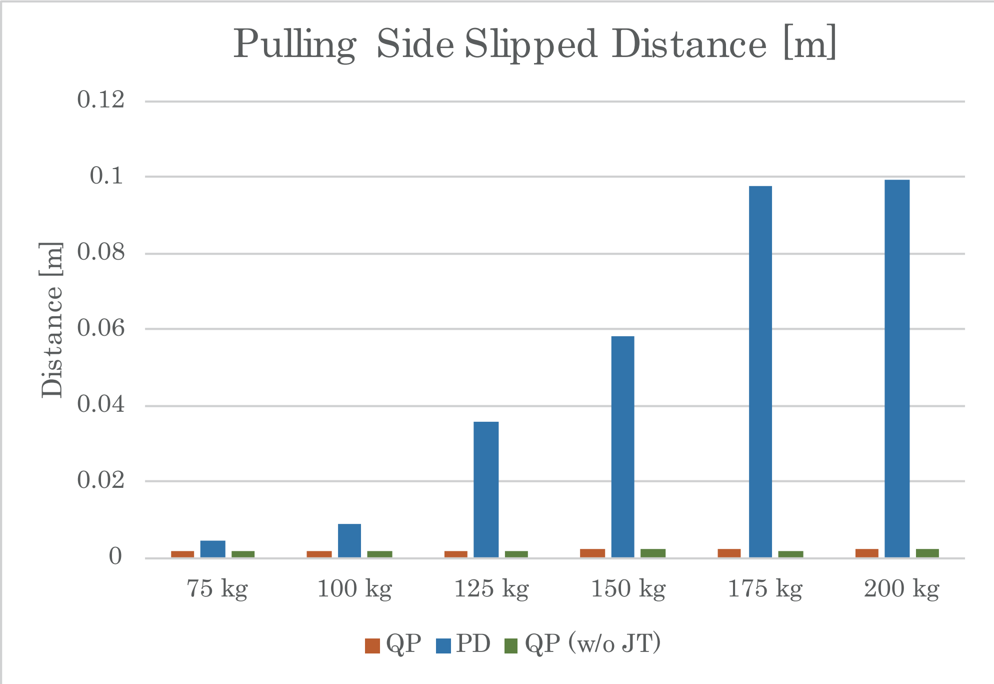


Figure 5.21: Slipped distance of the pulling robot's feet along x-axis (relative to variable masses).

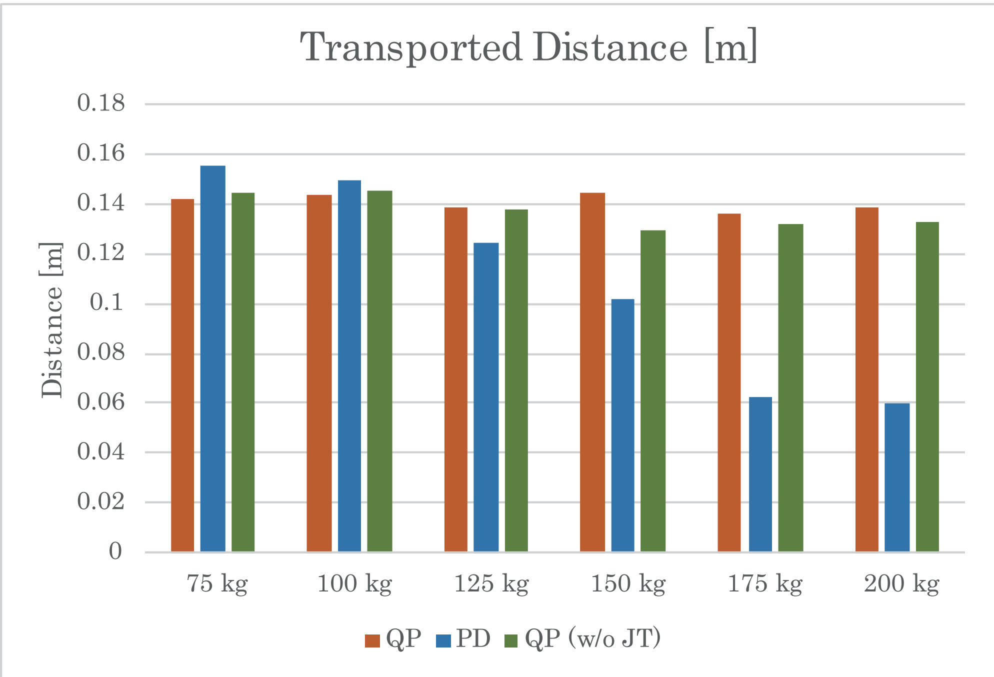


Figure 5.22: Transported distance of the massive object along x-axis (relative to variable masses).

Average Normalized Joint Torque							
		75 kg	100 kg	125 kg	150 kg	175 kg	200 kg
QP	Push	0.0229	0.0332	0.042	0.049	0.0627	0.0623
	Pull	0.0168	0.0213	0.0259	0.0282	0.0415	0.0435
	Sum	0.0397	0.0545	0.0679	0.0772	0.1042	0.1058
	Sum/distance	0.2791842	0.3800558	0.489899	0.5327812	0.7678703	0.7638989
PD	Push	0.0321	0.0519	0.0746	0.1057	0.1289	0.134
	Pull	0.0384	0.0444	0.0449	0.0439	0.0524	0.0581
	Sum	0.0705	0.0963	0.1195	0.1496	0.1813	0.1921
	Sum/distance	0.4548387	0.6424283	0.959069	1.4666667	2.9054487	3.2177554
QP (w/o JT)	Push	0.0299	0.0481	0.0656	0.0736	0.0863	0.0895
	Pull	0.0127	0.0151	0.0185	0.0252	0.043	0.0465
	Sum	0.0426	0.0632	0.0841	0.0988	0.1293	0.136
	Sum/distance	0.295628	0.4349621	0.6103048	0.7623457	0.9795455	1.0225564

Figure 5.23: Table of average of normalized joint torques (relative to variable masses).

5.4.3 Comparison of the average normalized joint torque

The average values of normalized joint torques in the QP, PD, and QP (w/o JT) simulations are demonstrated in Fig 5.23.

The average of normalized joint torques in the QP, PD, and QP (w/o JT) simulations are demonstrated in a bar chart, in Fig 5.24. The robots in QP and QP (w/o JT) simulations could transport the massive object with fewer joint torques. Additionally, the difference in the average of the normalized joint torques gets more significant relative to the mass increase of the massive object. Compare to the QP (w/o JT) simulations; in QP simulations, the pushing robot could transport the massive object by employing fewer joint torques. However, the pulling robot could transport the massive object with fewer joint torques only in 175 kg and 200 kg, in the QP simulations compare to the QP (w/o JT) simulations. Slightly more joint torques were employed by the pulling robot to transport the object with other masses. Importantly, the difference between the QP and QP (w/o JT) in pushing robot is relatively more significant than that in the pulling side. Hence, for all object masses transported in the QP simulations, the sum of the average of the normalized joint torques at the pushing and pulling robots is less than that in the QP (w/o JT) simulation (see Fig. 5.25).

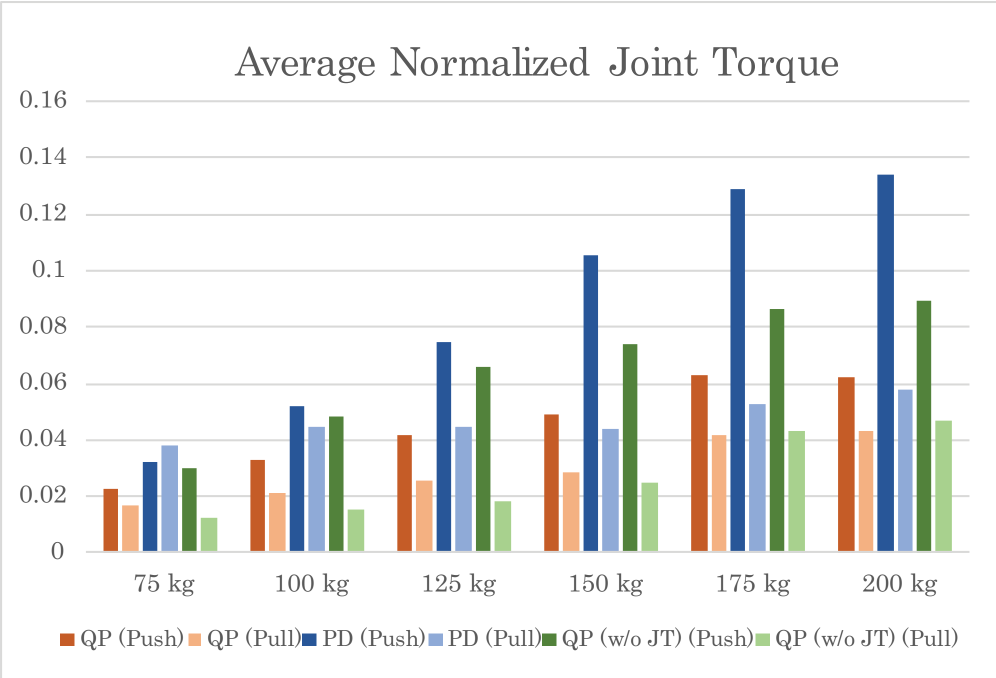


Figure 5.24: Average of normalized joint torques (relative to variable masses).

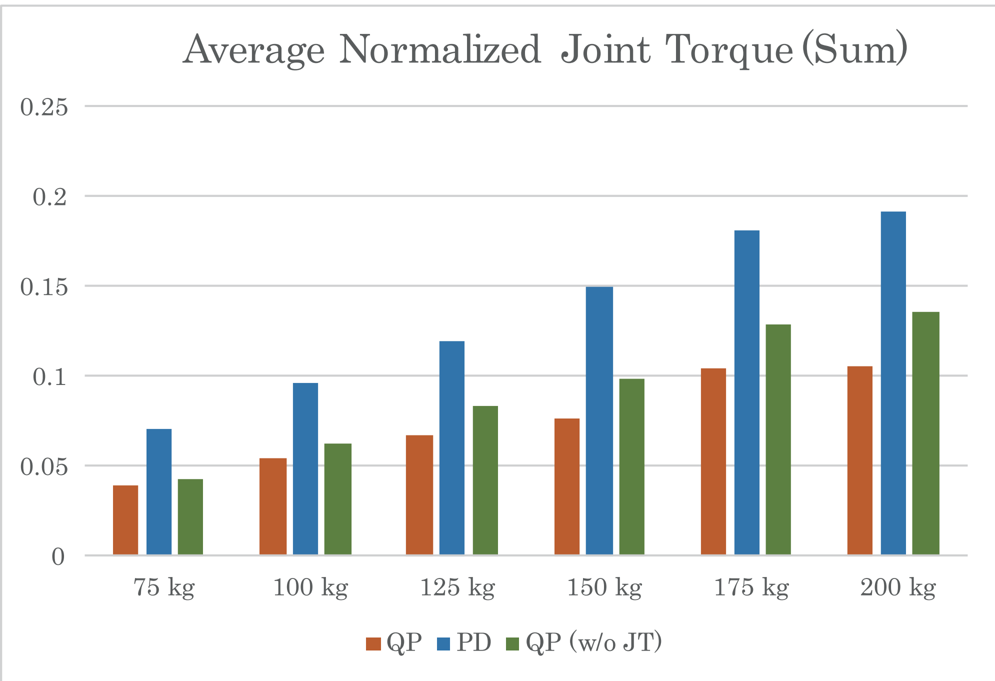


Figure 5.25: Sum of the average of normalized joint torques at the pushing and pulling robots (relative to variable masses).

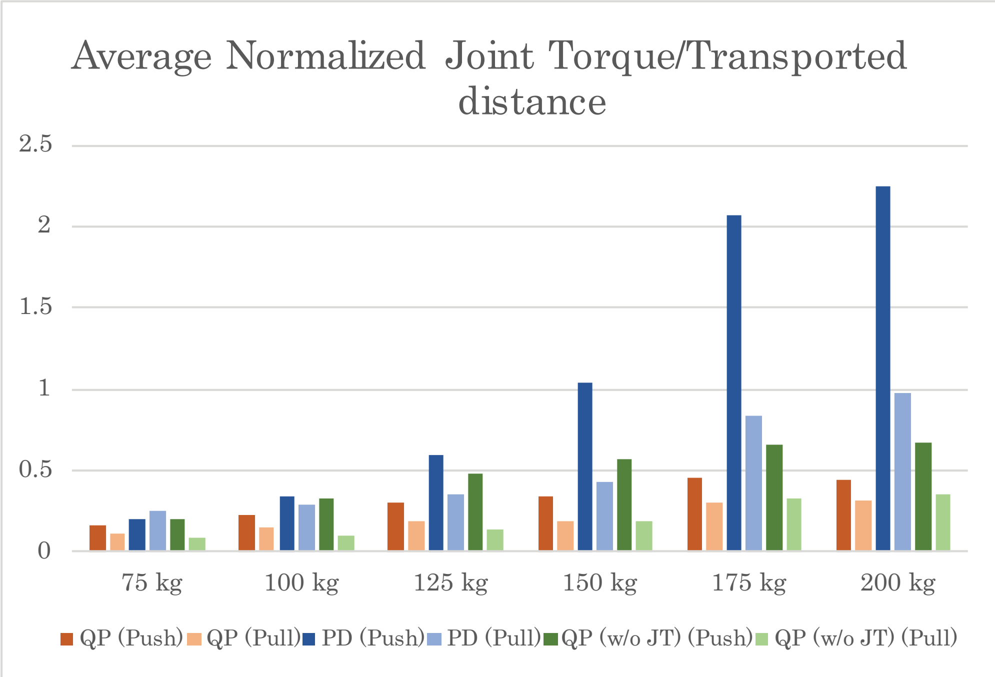


Figure 5.26: Average of normalized joint torques per unit distance (relative to variable masses).

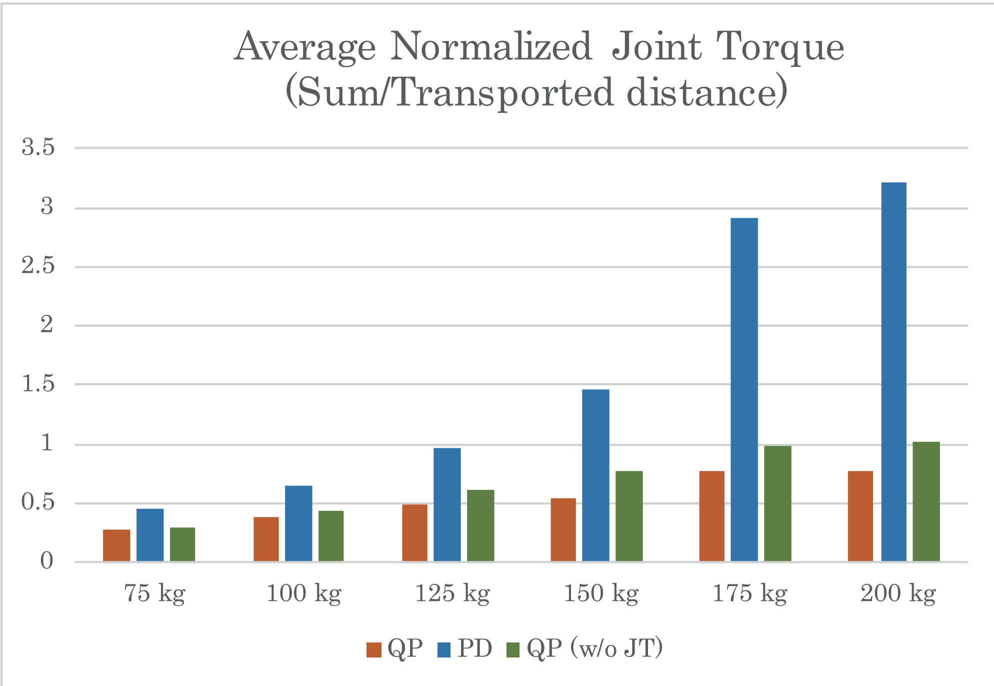


Figure 5.27: Sum of the average of the normalized joint torques at the pushing and pulling robots per unit distance (relative to variable masses).

Furthermore, as the mass of the object increases, the differences in sum of the average of the normalized joint torques at the pushing and pulling robots get more significant; this indicates the advantage of minimization of the joint torques objective function applied in our method (QP).

Due to the slight differences in the transported distance, the avarage of the joint torques per unit distance are demonstrated in Fig. 5.26 and Fig. 5.27.

In QP and QP (w/o JT) simulations, the average of the joint torques per unit distance in Fig. 5.26 is about the same as in Fig. 5.24. However, in PD simulations, the average of the joint torques per unit distance is relatively high compared to those in Fig. 5.24. Ultimately, the advantage of the proposed method in transporting a massive object effectively by employing fewer joint torques is demonstrated in Fig. 5.27.

5.5 Summary

In this chapter, the intermittent pushing/pulling of a 200 kg massive object was performed in a dynamic simulator. By performing one pushing/pulling phase, then one stepping/recovering posture phase, and finally, the last pushing/pulling phase, the massive object (200 kg) was transported about 0.2 m properly. The humanoid robot feet step to the determined position within an error of less than 0.001 m during the stepping/recovering posture phase. Furthermore, to reset the external forces and the joint torques to the initial condition, the humanoid robot posture could be recovered to the initial stance properly.

On the other hand, the simulations of the massive object transportation with QP and PD and QP without minimize joint torque objective function (QP (w/o JT)) are compared with variation of massive object masses in this chapter. The simulation results show that the proposed method QP transported massive object without feet slipping. However, conventional method PD could not transport the massive object effectively according to the increase of the massive object mass, due to the slipping of the feet. Also, the proposed method QP transported massive object with less joint torque per unit distance than the conventional method PD and QP (w/o JT). The difference of the joint torque per unit distance is getting greater according to the increase of the massive object masses. These point show that the advantage of the proposed method and the minimize joint torque objective function. Additionally, the massive object transportation using QP and PD and QP without minimizing joint torques objective function (QP (w/o JT)) was performed in the simulations including comparing the variation of the massive object masses. The simulation results confirmed that the proposed method (QP) was able to transport the massive object without feet slipping. In contrast, the conventional method (PD) employed to carry the massive object could not avoid the robot's feet slipping as the mass of the object increased. Furthermore, the massive object was transported with less joint torque per unit distance using the proposed method (QP) compared to the conventional method PD and QP (w/o JT). It was demonstrated that as the mass of the object increases, the difference of the joint torque per unit distance increases too. The results confirm the advantage of the proposed method and the benefit of using the joint torques objective function minimization.

Chapter 6 Conclusion

6.1 Summary

In this thesis, massive object transportations by robots are described. As a method of massive object transportation by robots, two kinds of methods are described. One is the teleoperation of the manipulator to transport the massive object, and the other is the massive object transportation by two humanoid robots based on a simple command.

The massive object transportation using teleoperation

In the very long time delay teleoperation of the massive object transportation, the unexpected excessive force due to the operator ' s scaled-up commands are the key points. Therefore, the mitigation of the force generated between the end-effector and the massive payload are proposed. Also, the virtual environment is proposed, which is considered the force between the end-effector and the massive payload.

The verification experiment of the proposed system was performed using the hardware-in-the-loop simulations (HILs). The mitigation of generated force between the robot hand and the massive object is verified by comparing the simulation results with and without mitigation term. The HILs simulation of the massive object (6000 kg) transportation by an operator in the very long time delay environment is successfully performed that the force generated at hands is mitigated and is reproduced on the virtual environment.

The massive object transportation by humanoid robots

Massive object transportation by a humanoid robot is also performed as a previous step for the massive object transportation by two humanoids robots, The massive object transportation by two humanoids robots is performed using the real-time QP solver. The joint accelerations are optimized using the objective function to minimize the joint torques and satisfying the inequality/equality constraints. The method for estimating the maximum reaction force from the ground is proposed by attaching huge virtual masses at the end of the limbs.

In order to show the effectiveness of the proposed method, the simulations of the massive object transportation with QP and PD and QP without minimizing joint torques objective function (QP (w/o JT)) are compared relative to the variation of massive object masses in this chapter. The simulation results show that the proposed method QP transported the massive object without feet slipping. However, conventional method PD could not transport the massive object effectively according to the increase in the massive object mass, due to the slipping of the feet. Also, the proposed method QP transported massive object with less joint torques per unit distance compared to the conventional method PD and QP (w/o JT).

In order to transport the massive object further, the simulation of the intermittent pushing/pulling of the massive object (200 kg) by two humanoid robots is performed. As a result, the massive object (200 kg) is properly transported about 0.2 m.

6.2 Future work

Model-based teleoperation of massive object trnasportation using two humanoid robots

As a solution for the limitation of the fully autonomous robots, the teleoperation method mentioned in Chapter 2 can be combined with the method of massive object transportation using two humanoid robots mentioned in Chapter 4. Two humanoid robots can be remotely operated to return to the starting position of transportation by employing the model-based teleoperation using virtual environment. Then perform a massive object transportation task by a simple command from an operator.

Considering an optimum initial posture of the humanoid robots

In this thesis, a simple initial posture of humanoid robots is applied for transportation. In order to improve the efficiency of transportation, the optimum initial posture of the robots for massive object transportation should be pre-calculated using the QP solver.

Considering an optimized asymmetry motion for transporting massive object.

In this thesis, motions of the massive object transportation are generated symmetrically. In order to improve the efficiency of the joint torques and the maximum force of the end effectors, an asymmetric motion generation of the humanoid robots should be considered as future work.

Considering a motion generation for massive object transportation in various environments.

In this thesis, only a flat floor and the same friction coefficient are considered. In the case of the inclined floor and the variable friction coefficient will be considered as future work. The quadratic problem depends on the actual model of the environments. Therefore, the model should be revised to the new environments. However, in the case of the small changes in the environments, the proposed model could be used by adjusting the constraints more conservatively.

Acknowledgements

The research of massive object transportation by robots was conducted in intelligent robots and system laboratory of Hokkaido university graduate school of information science and technology. First of all, I would like to thank my adviser, Professor Konno for his kindness support. I have learnd from him including approaches and ideas for research. He also supported me for not only research but also mental encouragement. I would like to thank Professor Masahiko Onosato and Professor Satoshi Ogasawara for their advices about my research.

I also would like to thank Professor Masaru Uchiyama and Associate Professor Satoko Abiko for the support and useful advices of teleoperation researches. I would like to thank Dr. Shunsuke Komizunai for the discussions about my researches. I greatly thankful to all the members of intelligent robots and systems laboratory.

I am also deeply grateful to the members of the humanoid research group, Mr. Ogawa, Mr. Alireza, Mr. Hamada, Mr. Saito, Mr. Tsukimoto, Mr. Tanaka, Ms. Inoue, and Mr. Sawanobori for their discussions and support about my researches. Specially I am grateful Mr. Ogawa and Mr. Hamada for the robot programming support and grateful Mr. Alireza for the english support. Also I am grateful Mr. Tsukimoto for the support of preparing the papers.

Besides, I am also profoundly grateful to the Hokkaido University graduate school of information science and technology for research assistant.

At last, I would like to thank my family, and friends for encouragement and discussions for researches and life. Without their encouragement, this research would not have accomplished.

References

[1] H. Lee and A. Banerjee, "Intelligent scheduling and motion control for household vacuum cleaning robot system using simulation based optimization," in *2015 Winter Simulation Conference (WSC)*, pp. 1163–1171, 2015.

[2] I. Braithwaite, M. Blanke, G.-Q. Zhang, and J. M. Carstensen, "Design of a vision-based sensor for autonomous pig house cleaning," *EURASIP Journal on Advances in Signal Processing*, vol. 2005, no. 13, p. 473479, 2005.

[3] V. Prabakaran, M. R. Elara, T. Pathmakumar, and S. Nansai, "Floor cleaning robot with reconfigurable mechanism," *Automation in Construction*, vol. 91, pp. 155–165, 2018.

[4] H. Zhang, J. Zhang, G. Zong, W. Wang, and R. Liu, "Sky cleaner 3: A real pneumatic climbing robot for glass-wall cleaning," *IEEE Robotics & Automation Magazine*, vol. 13, no. 1, pp. 32–41, 2006.

[5] N. Mir-Nasiri, M. H. Ali *et al.*, "Portable autonomous window cleaning robot," *Procedia computer science*, vol. 133, pp. 197–204, 2018.

[6] T. Akinfiev, M. Armada, and S. Nabulsi, "Climbing cleaning robot for vertical surfaces," *Industrial Robot: An International Journal*, vol. 36, no. 4, pp. 352–357, 2009.

[7] A. Visioli and G. Legnani, "On the trajectory tracking control of industrial scara robot manipulators," *IEEE transactions on industrial electronics*, vol. 49, no. 1, pp. 224–232, 2002.

[8] D. Constantinescu and E. A. Croft, "Smooth and time-optimal trajectory planning for industrial manipulators along specified paths," *Journal of robotic systems*, vol. 17, no. 5, pp. 233–249, 2000.

[9] S. KuCuk and Z. Bingul, "The inverse kinematics solutions of industrial robot manipulators," in *Proceedings of the IEEE International Conference on Mechatronics, 2004. ICM'04.*, pp. 274–279, 2004.

[10] P.-C. Huang and A. K. Mok, "A case study of cyber-physical system design: Autonomous pick-and-place robot," in *2018 IEEE 24th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, pp. 22–31, 2018.

[11] Y. Xiao, B. Zhou, and J. Xuan, "Robot intelligent welding programming based on line structure light sensing," in *2018 33rd Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pp. 800–804, 2018.

[12] M. Kyrarini, M. A. Haseeb, D. Ristić-Durrant, and A. Gräser, "Robot learning of industrial assembly task via human demonstrations," *Autonomous Robots*, vol. 43, no. 1, pp. 239–257, 2019.

[13] D. Almeida and Y. Karayiannidis, "Folding assembly by means of dual-arm robotic manipulation," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3987–3993, 2016.

[14] S.-Y. Lee and H.-W. Yang, "Navigation of automated guided vehicles using magnet spot guidance method," *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 3, pp. 425–436, 2012.

[15] S. Sabikan, M. Sulaiman, S. N. S. Salim, and M. F. Miskon, "Vision based automated guided vehicle for navigation and obstacle avoidance," in *Proceeding of the 2nd International Conference on Engineering and ICT*, pp. 18–20, 2010.

[16] H. Martinez-Barbera and D. Herrero-Perez, "Development of a flexible agv for flexible manufacturing systems," *Industrial robot: An international journal*, vol. 37, no. 5, pp. 459–468, 2010.

[17] K. Gao, J. Xin, H. Cheng, D. Liu, and J. Li, "Multi-mobile robot autonomous navigation system for intelligent logistics," in *2018 Chinese Automation Congress (CAC)*, pp. 2603–2609, 2018.

[18] J. Wen, L. He, and F. Zhu, "Swarm robotics control and communications: Imminent challenges for next generation smart logistics," *IEEE Communications Magazine*, vol. 56, no. 7, pp. 102–107, 2018.

[19] I. Schjølberg and I. B. Utne, "Towards autonomy in rov operations," *IFAC-PapersOnLine*, vol. 48, no. 2, pp. 183–188, 2015.

[20] C. Mai, S. Pedersen, L. Hansen, K. Jepsen, and Z. Yang, "Modeling and control of industrial rov's for semi-autonomous subsea maintenance services," *Ifac-papersonline*, vol. 50, no. 1, pp. 13 686–13 691, 2017.

[21] H. Komura, H. Yamada, and S. Hirose, "Development of snake-like robot acm-r8 with large and mono-tread wheel," *Advanced Robotics*, vol. 29, no. 17, pp. 1081–1094, 2015.

[22] T. Yanagida, M. Kasahara, and M. Iwase, "Locomotion control of snake-like robot on geometrically smooth surface," *IFAC-PapersOnLine*, vol. 48, no. 11, pp. 162–167, 2015.

[23] M. Hiller, "Modelling, simulation and control design for large and heavy manipulators," *Robotics and autonomous systems*, vol. 19, no. 2, pp. 167–177, 1996.

[24] C. L. J, C. Craig R, and A. David L, "Time delay and communication bandwidth limitation on telerobotic control," *Intelligent Systems and Smart Manufacturing. International Society for Optics and Photonics*, vol. 4195, pp. 405–419, 2001.

[25] C. Secchi, S. Stramigioli, and C. Fantuzzi, "Variable delay in scaled port-hamiltonian telemanipulation," *Mechatronics*, vol. 18, no. 7, pp. 357–363, 2008.

[26] G. NIEMEYER and J.-J. E. SLOTINE, "Telemanipulation with time delays," *The International Journal of Robotics Research*, vol. 23.9, pp. 873–890, 2004.

[27] W. Ryder C, S. Sean M, H. Elliot W, C. David L, J. Hao, C. Mark R, and O. Allison M, "Time-delayed teleoperation for interaction with moving objects

in space," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, June 2014.

[28] Y. Tsumaki, T. Goshozono, K. Abe, M. Uchiyama, R. Koeppe, and G. Hirzinger, "Verification of an advanced space teleoperation system using internet," in *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)*, vol. 2, pp. 1167–1172, 2000.

[29] Y. Tsumaki and M. Uchiyama, "A model-based space teleoperation system using mixed force and motion commands," in *Transaction of the JSME, Series C*, vol. 64(626), pp. 3901–3906, 1998.

[30] W.-K. Yoon, T. Goshozono, H. Kawabe, M. Kinami, Y. Tsumaki, M. Uchiyama, M. Oda, and T. Doi, "Model-based space robot teleoperation of ets-vii manipulator," *IEEE Trans. on Robotics and Automation*, vol. 20, no. 3, pp. 602–612, 2004.

[31] H. Arisumi, J.-R. Chardonnet, A. Kheddar, and K. Yokoi, "Dynamic lifting motion of humanoid robots," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 2661–2667, 2007.

[32] K. Harada, S. Kajita, H. Saito, M. Morisawa, F. Kanehiro, K. Fujiwara, K. Kaneko, and H. Hirukawa, "A humanoid robot carrying a heavy object," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 1712–1717, 2005.

[33] Y. Ohmura and Y. Kuniyoshi, "Humanoid robot which can lift a 30kg box by whole body contact and tactile feedback," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1136–1141, 2007.

[34] H.-Y. Lee, H.-P. Huang, and H.-K. Hsu, "Lifting motion planning for humanoid robots," in *Proceedings of IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 1174–1179, 2014.

[35] M. Grey, S. Joo, and M. Zucker, "Planning heavy lifts for humanoid robots," in *Proceedings of IEEE-RAS International Conference on Humanoid Robots*, pp. 640–645, 2014.

[36] M. Murooka, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba, "Whole-body pushing manipulation with contact posture planning of large and heavy object for humanoid robot," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5682–5689, 2015.

[37] H.-G. Kim, J.-U. An, J.-I. Moon, and D.-H. Kim, "Simulation based for intelligent control system of multi—humanoid robots for dynamic load carrying," in *ICCAS 2010*, pp. 101–104, 2010.

[38] S. A. A. Moosavian, A. Janati, and M. H. Ghazikhani, "Object manipulation by two humanoid robots using mtj control," in *Proceedings of IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 1286–1290, 2011.

[39] M.-H. Wu, S. Ogawa, and A. Konno, "Symmetry position/force hybrid control for cooperative object transportation using multiple humanoid robots," *Advanced Robotics*, vol. 30, no. 2, pp. 131–149, 2016.

[40] S. G. McGill and D. D. Lee, "Cooperative humanoid stretcher manipulation and locomotion," in *Proceedings of IEEE-RAS International Conference on Humanoid Robots*, pp. 429–433, 2011.

[41] M. S. Motahar, S. Veer, J. Huang, and I. Poulakakis, "Integrating dynamic walking and arm impedance control for cooperative transportation," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1004–1010, 2015.

[42] K. Otani, K. Bouyarmane, and S. Ivaldi, "Generating assistive humanoid motions for co-manipulation tasks with a multi-robot quadratic program controller," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3107–3113, 2018.

[43] J. Stückler and S. Behnke, "Following human guidance to cooperatively carry a large object," in *2011 11th IEEE-RAS International Conference on Humanoid Robots*, pp. 218–223, 2011.

[44] D. J. Agravante, A. Cherubini, A. Bussy, P. Gergondet, and A. Kheddar, "Collaborative human-humanoid carrying using vision and haptic sensing," in *2014 IEEE international conference on robotics and automation (ICRA)*, pp. 607–612, 2014.

[45] K. Yokoyama, H. Handa, T. Isozumi, Y. Fukase, K. Kaneko, F. Kanehiro, Y. Kawai, F. Tomita, and H. Hirukawa, "Cooperative works by a human and a humanoid robot," in *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, vol. 3, pp. 2985–2991, 2003.

[46] S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba, "Controlling the planar motion of a heavy object by pushing with a humanoid robot using dual-arm force control," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1428–1435, 2012.

[47] A. Rioux and W. Suleiman, "Autonomous slam based humanoid navigation in a cluttered environment while transporting a heavy load," *Robotics and Autonomous Systems*, vol. 99, pp. 50–62, 2018.

[48] T. Koolen, S. Bertrand, G. Thomas, T. De Boer, T. Wu, J. Smith, J. Englsberger, and J. Pratt, "Design of a momentum-based control framework and application to the humanoid robot atlas," *International Journal of Humanoid Robotics*, vol. 13, no. 01, p. 1650007, 2016.

[49] C. P. Sayers, P. Richard P, W. Louis L, and Y. Dana R, "Teleprogramming for subsea teleoperation using acoustic communication," *IEEE Journal of Oceanic Engineering*, vol. 23.1, pp. 60–71, 1998.

[50] K. Yoshida, "Experimental platforms for research and development of space robots," *Journal of the Robotics Society of Japan*, vol. 14-1, pp. 18–21, 1996.

[51] A. Farhad, N. Mehrzad, and V. George, "Satellite simulator with a hydraulic manipulator," in *Proceedings 2006 IEEE International Conference on Robotics and Automation(ICRA 2006). IEEE*, May 2006.

[52] X. Wenfu, L. Bin, X. Yangsheng, L. Cheng, and Q. Wenyi, "A ground experiment system of free-floating robot for capturing space target," *Journal of Intelligent and Robotic Systems*, vol. 48.2, pp. 187–208, 2007.

[53] G. Rouleau, I. Rekleitis, R. L'Archeveque, E. Martin, K. Parsa, and E. Dupuis, "Autonomous capture of a tumbling satellite," in *Proceedings 2006 IEEE International Conference on Robotics and Automation(ICRA 2006). IEEE*, May 2006.

[54] T. Boge, T. Wimmer, O. Ma, and M. Zebenay, "Epos- a robotics-based hardware-in-the-loop simulator for simulating satellite rvd operations," in *10th International Symposium on Artificial Intelligence Robotics and Automation in Space*, Sep 2010.

[55] K. Osaki, A. Konno, and M. Uchiyama, "Delay time compensation for a hybrid simulator," *Advanced Robotics*, vol. 24, no. 8-9, pp. 1081–1098, 2010.

[56] R. Takahashi, H. Ise, A. Konno, M. Uchiyama, and D. Sato, "Hybrid simulation of a dual-arm space robot colliding with a floating object," in *2008 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1201–1206, 2008.

[57] M. Uchiyama, K.-i. Iimura, F. Pierrot, K. Unno, and O. Toyama, "Design and control of a very fast 6-dof parallel robot," *Robotics Mechatronics and Manufacturing Systems*, pp. 473–478, 1992.

[58] Barrett Advanced Robotics, https://advanced.barrett.com/.

[59] Y. Tsumaki, H. Naruse, D. N. Nenchev, and M. Uchiyama, "Design of a compact 6-dof haptic interface," in *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2580–2585, 1998.

[60] R. Kikuuwe, S. Yasukouchi, H. Fujimoto, and M. Yamamoto, "Proxy-based sliding mode control: a safer extension of pid position control," *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 670–683, 2010.

[61] R. Kikuuwe, T. Yamamoto, and H. Fujimoto, "A guideline for low-force robotic guidance for enhancing human performance of positioning and trajectory tracking: It should be stiff and appropriately slow," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 38, no. 4, pp. 945–957, 2008.

[62] M.-H. Wu, A. Konno, and M. Uchiyama, "Cooperative object transportation by multiple humanoid robots," in *2011 IEEE/SICE International Symposium on System Integration (SII)*, pp. 779–784, 2011.

[63] A. Rioux and W. Suleiman, "Humanoid navigation and heavy load transportation in a cluttered environment," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2180–2186, 2015.

[64] QuadProg++, https://github.com/liuq/QuadProgpp.

[65] Fujitsu HOAP, http://home.comcast.net/jtechsc/HOAP-3-Spec-Sheet.pdf.

[66] Robotis OP2, https://www.asratec.co.jp/portfolio-page/robotis-op2/.

[67] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura, "The intelligent asimo: System overview and integration," in *IEEE/RSJ international conference on intelligent robots and systems*, vol. 3, pp. 2478–2483, 2002.

[68] PLEN, https://plen.jp/wp/plen2/.

[69] H. Inoue and H. Hirukawa, "Hrp: humanoid robotics project of miti," *Journal of the Robotics Society of Japan*, vol. 19, no. 1, pp. 2–7, 2001.

[70] Atlas - The Agile Anthropomorphic Robot, https://www.bostondynamics.com/atlas.

[71] PETMAN, https://www.army-technology.com/projects/petman/.

[72] TALOS, http://pal-robotics.com/en/products/talos/.

[73] REEM-C, http://reemc.pal-robotics.com/en/reemc/.

[74] F. Kanehiro, H. Hirukawa, and S. Kajita, "Openhrp: Open architecture humanoid robotics platform," *The International Journal of Robotics Research*, vol. 23, no. 2, pp. 155–165, 2004.

[75] S. K. H. H. T. K. M. H. K. A. Kenji Kaneko, Fumio Kanehiro and T. Isozumi, "Humanoid robot hrp-2," in *IEEE International Conference on Robotics and Automation*.

[76] K. Yoshida, R. Kurazume, and Y. Umetani, "Dual arm coordination in space free-flying robot," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2516–2521, 1991.

[77] Y. Zhang, S. S. Ge, and T. H. Lee, "A unified quadratic-programming-based dynamical system approach to joint torque optimization of physically constrained redundant manipulators," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 5, pp. 2126–2132, 2004.

[78] S. Kuindersma, F. Permenter, and R. Tedrake, "An efficiently solvable quadratic program for stabilizing dynamic locomotion," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2589–2594, 2014.

[79] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous Robots*, vol. 40, no. 3, pp. 429–455, 2016.

[80] L. Saab, O. E. Ramos, F. Keith, N. Mansard, P. Soueres, and J.-Y. Fourquet, "Dynamic whole-body motion generation under rigid contacts and other unilateral constraints," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 346–362, 2013.

List of Publications

Journal Publications

[1] **Jaesung Yang**, Atsushi Konno, Satoko Abiko and Masaru Uchiyama, "Hardware-in-the-loop simulation of massive-payload manipulation on orbit," *ROBOMECH Journal*, vol. 5, No. 19 (12 pages), DOI: 10.1186/s40648-018-0116-8, August 31, 2018.

International Conferences (Full Review)

[1] **Jaesung Yang**, Shuhei Ogawa, Teppei Tsujita, Shunsuke Komizunai, and Atsushi Konno, "Massive object transportation by a humanoid robot," in *Proceedings of the IFAC Symposium on Robot Control (SYROCO)*, Paper No. 017 (6 pages), Budapest, Hungary, August 27-30, 2018.

[2] **Jaesung Yang**, Teppei Tsujita, Shunsuke Komizunai and Atsushi Konno, "Cooperative massive object transportation by two humanoid robots," in *Proceedings of ACA/JSME Asian Control Conference 2019 (ASCC 2019)*, pp. 498-503, Kitakyushu, Japan, June 9 -12, 2019.

Appendix Dynamics of the Virtual Arm

In the case of the high-ratio reduction gears are used in the joint, the joint angles and torques at the before and after of the gear reduction are expressed as

$$\boldsymbol{G}_r \boldsymbol{q} = \boldsymbol{q}_a, \tag{1}$$

$$\boldsymbol{\tau} = \boldsymbol{G}_r^T \boldsymbol{\tau}_a \quad (\because \Delta \boldsymbol{q}^T \boldsymbol{\tau} = \Delta \boldsymbol{q}_a^T \boldsymbol{\tau}_a = \Delta \boldsymbol{q}^T \boldsymbol{G}_r^T \boldsymbol{\tau}_a), \tag{2}$$

where $\boldsymbol{q}_a$ is the angle vector of the actuators; $\boldsymbol{q}$ is the joint angle vector of an arm; $\boldsymbol{\tau}$ is the joint torque vector; $\boldsymbol{\tau}_a$ is the torque vector of the actuators; and $\boldsymbol{G}_r$ is a diagonal reduction ratio matrix.

The dynamics of the arm is given by

$$\boldsymbol{\tau} = \boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{h} + \boldsymbol{J}^T \boldsymbol{F}, \tag{3}$$

where $\boldsymbol{J}$ is the Jacobian matrix, $\boldsymbol{M}$ is the inertia matrix of the arm, $\boldsymbol{F}$ is the external force, and $\boldsymbol{h}$ represents the Coriolis force and centrifugal force and gravitational force.

The following equation is created by substituting (1) and (2) into (3):

$$\boldsymbol{\tau}_a = \boldsymbol{G}_r^{-T} \boldsymbol{M} \boldsymbol{G}_r^{-1} \ddot{\boldsymbol{q}}_a + \boldsymbol{G}_r^{-T}(\boldsymbol{h} + \boldsymbol{J}^T \boldsymbol{F}). \tag{4}$$

The actuator torque is calculated as following equation by assuming that the actuator is the velocity command type:

$$\boldsymbol{\tau}_a = \boldsymbol{K}_{va}(\dot{\boldsymbol{q}}_{ca} - \dot{\boldsymbol{q}}_a) = \boldsymbol{K}_{va} \boldsymbol{G}_r (\dot{\boldsymbol{q}}_c - \dot{\boldsymbol{q}}) \quad (\because \dot{\boldsymbol{q}}_{ca} - \dot{\boldsymbol{q}}_a = \boldsymbol{G}_r(\dot{\boldsymbol{q}}_c - \dot{\boldsymbol{q}})), \tag{5}$$

where $\dot{\boldsymbol{q}}_{ca}$ is the joint angle velocity command. $\boldsymbol{K}_{va}$ matches to the gain of the hardware velocity servo.

The following equation is obtained by substituting (5) into (4):

$$\dot{\boldsymbol{q}}_c - \dot{\boldsymbol{q}} = \boldsymbol{G}_r^{-1} \boldsymbol{K}_{va}^{-1} \boldsymbol{G}_r^{-T} \boldsymbol{M} \boldsymbol{G}_r^{-1} \ddot{\boldsymbol{q}}_a + \boldsymbol{G}_r^{-1} \boldsymbol{K}_{va}^{-1} \boldsymbol{G}_r^{-T} (\boldsymbol{h} + \boldsymbol{J}^T \boldsymbol{F}). \tag{6}$$

The following equation is obtained by multiplying both sides of (6) by $\boldsymbol{J}$:

$$\boldsymbol{J}(\dot{\boldsymbol{q}}_c - \dot{\boldsymbol{q}}) = \boldsymbol{J}\boldsymbol{G}_r^{-1}\boldsymbol{K}_{va}^{-1}\boldsymbol{G}_r^{-T}\boldsymbol{M}\boldsymbol{G}_r^{-1}\ddot{\boldsymbol{q}}_a + \boldsymbol{J}\boldsymbol{G}_r^{-1}\boldsymbol{K}_{va}^{-1}\boldsymbol{G}_r^{-T}(\boldsymbol{h} + \boldsymbol{J}^T\boldsymbol{F}). \quad (7)$$

Therefore, by using $\boldsymbol{v}_c = \boldsymbol{J}\dot{\boldsymbol{q}}_c$ and $\boldsymbol{v} = \boldsymbol{J}\dot{\boldsymbol{q}}$, (7) is rewritten as

$$\boldsymbol{v}_c - \boldsymbol{v} = \boldsymbol{J}\boldsymbol{G}_r^{-1}\boldsymbol{K}_{va}^{-1}\boldsymbol{G}_r^{-T}\boldsymbol{M}\boldsymbol{G}_r^{-1}\ddot{\boldsymbol{q}}_a + \boldsymbol{J}\boldsymbol{G}_r^{-1}\boldsymbol{K}_{va}^{-1}\boldsymbol{G}_r^{-T}(\boldsymbol{h} + \boldsymbol{J}^T\boldsymbol{F}). \quad (8)$$

In the case of the reduction ratio is large enough, due to the first term is proportional to $\boldsymbol{G}^{-3}$ and the second term is proportional to $\boldsymbol{G}^{-2}$, the two terms on the right hand side of (8) can be omitted. Therefore, $\boldsymbol{v}$ can be written as $\boldsymbol{v}_c$. $\boldsymbol{F}$ in the second term may become very large in massive-payload manipulation simulation, and it may not be possible to ignore them. However, the second term of the right-hand side of (8) will be sufficiently small enough that the assumption about $\boldsymbol{v} \approx \boldsymbol{v}_c$ will be satisfied, if $\boldsymbol{F}$ is appropriately mitigated by force feedback term (e.g. $\boldsymbol{K}_s\boldsymbol{F}_s$ in (2.2)).