NAIST-IS-DD1661014

Doctoral Dissertation

Fast Solution of Whole-body Inverse Kinematics and Generation of Target Movement Using Prior Knowledge for Humanoid Robots

Yuya Hakamata

March 17, 2021

Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Yuya Hakamata

Thesis Committee:

Professor Tsukasa Ogasawara	(Supervisor)
Professor Kenji Sugimoto	(Co-supervisor)
Associate Professor Jun Takamatsu	(Co-supervisor)
Assistant Professor Gustavo Alfonso Garcia Ricardez	(Co-supervisor)
Assistant Professor Satoki Tsuichihara	(University of Fukui)

Fast Solution of Whole-body Inverse Kinematics and Generation of Target Movement Using Prior Knowledge for Humanoid Robots*

Yuya Hakamata

Abstract

As robotics technology advances, we expect humanoid robots to perform tasks that need physical interaction instead of us. In daily life, we frequently conduct object manipulations. In particular, many household tasks involve pushing and pulling motions, e.g., opening a door and pulling a drawer.

To make the humanoid robots work in our daily-life environments, two requirements should be satisfied. First, the robots should manipulate objects including physical interaction. Second, the robots should conduct work in a short time, with similar speed to human movements. Several studies on opening a door using the humanoid robot have been conducted. However, due to the delays of sensor feedback, the generated motions are slower than human motions.

I propose two methods to speed up whole-body motion generation for humanoid robots. One method is to control the whole body's momentum using analytical inverse kinematics and Resolved Momentum Control. It is possible to reduce the computation time because no iterative calculation is required. The other is to derive the target trajectories of the Center of Mass (CoM) and hands using prior knowledge of the target object. As a prior knowledge, by configuring a reaction force of the object in a pushing motion, I derive target trajectories of the whole body with stability in a short time. Using prior knowledge of daily tools, such as a door or a chair, the robot can manipulate them quickly and safely in their daily life. Humans can learn the weight and frictional force of unknown

*Doctoral Dissertation, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD1661014, March 17, 2021.

tools through trial and error. On the other hand, for objects that were already known, the required force can be predicted, and thus the smooth manipulation is possible. In the proposed method, it is assumed that the applied force is known in advance to speed up the robot's motion at the same speed as a human being.

In this dissertation, the effectiveness of the proposed method is verified by using a humanoid robot, HRP-4, in dynamic simulations and a real robot. First, I conducted the experiment of the motion generation using the proposed momentum control in the dynamic simulation. During the kicking motion, I confirmed that the upper body movements decreased the lower body's momentum. The effectiveness of this method was confirmed by measuring the computation time and position errors of the foot and CoM compared to the previous method. Next, I conducted an experiment of the motion generation to push a 10 kg box. The target trajectories of the CoM and hands were calculated from the force required to push the box, which was measured in advance. The generated pushing motion was completed in 6 s. In order to show the versatility of this method, I conducted experiments on the pulling motion. The pulling motion in which the direction of force is backward from the pushing motion. It succeeded that the movement opening the refrigerator door was in 12 s. In addition, this method was verified in detail. I confirmed the behavior of the robot when pushing an object of different weight. In addition, the force exerted by the robot during the pushing motion was confirmed by pushing a force sensor.

Keywords:

humanoid robot, whole body motion, pushing motion, pulling motion, ZMP

ヒューマノイドロボットのための全身逆運動学の高速解法と予備知識を用いた運動目標の生成*

袴田　有哉

内容梗概

ロボット技術の進歩に伴い，人型ロボットは私たちに代わって力仕事をしてくれることを期待されている．日常生活の中では様々な物を動かす機会が多く存在し，特に，扉を開けたり引き出しを引いたりといった，物を押す，引くという動作が頻繁に行われている．人間の生活環境を有効に活用できるヒューマノイドロボットにとって，このように力を必要とする動作が可能になることが重要な課題となっている．

人間の生活環境でロボットを活躍させるためには，二つの要件を満たす必要がある．一つ目は，ロボットが身体的な相互作用を含めて対象物を操作できること．二つ目は，人間と同程度の速度で短時間に作業を行うことである．ヒューマノイドロボットを用いて，ドアの開閉や物体を押す研究はすでにいくつか行われているが，センサフィードバックの待ち時間が長いため，生成される動作は人間の動作に比べて遅くなっている．

動作を高速化するために，以下の二つの方法を提案する．一つは，解析的逆運動学と分解運動量制御を用いた全身運動量の制御方法である．手先と重心位置に対する反復計算を必要としないため，計算時間の削減を可能とした．もう一つは，対象物体の予備知識を用いた重心及び手先の目標軌道の導出方法である．予備知識として押し動作時の反力を与えることで，短時間で安定した動作を実現する目標軌道を導出する．ドアや椅子などの予備知識を用いることにより，日常生活の中でそれらを素早く安全に操作することが可能となる．人間は，未知の道具の重さや摩擦力を試行錯誤しながら覚えていくことが可能である．一方で，すでに既知となった物体に関しては必要な力を予測できるため，安定した動作が可能

*奈良先端科学技術大学院大学 情報科学研究科 博士論文, NAIST-IS-DD1661014, 2021 年 3 月 17 日.

となっている．提案手法では，ロボットに人間と同程度の速度で安定した動作をさせるために，印加される力が事前に分かっていることを前提としている．

本研究では，提案手法の有効性をヒューマノイドロボット HRP-4 を用いて動力学シミュレーションと実機で検証した．はじめに，全身運動量の制御に関し実験を行った．蹴り動作を行わせた際に，脚で発生した運動量を打ち消す動作が上半身で生成されることを確認した．その際に，従来法との比較として計算時間および足位置と重心位置の誤差を計測し，有効性を確認した．次に，10 [kg] の箱を押す動作を行わせた．事前に測定した箱を押すために必要な力の大きさから，重心及び手先の目標軌道を計算し，押し動作を 6 [s] で達成している．さらに，本手法に汎用性があることを示すため，押し動作と力の向きが逆となる引き動作の実験を行った．冷蔵庫の扉開けを 12 [s] で行うことに成功している．また，本手法の詳細な検証を行った．重さの異なる物体を押した際の挙動を実験により確かめた。さらに，押し動作中にロボットが発揮している力の大きさを，力センサを押すことで確認した．

キーワード

ヒューマノイドロボット, 全身動作生成, 押し動作, 引き動作, ZMP

Contents

List of Figures

List of Tables

1. Introduction

1.1 Research background

In recent years, advances in robotics have led to expectations for robots that can perform forceful tasks in daily life. Since everything in our daily life is designed for humans, humanoid robots are fit to perform such work instead of us, as they have a physical geometry and function similar to humans. Using prior knowledge of daily tools, such as a door or a chair, we can manipulate them in a quick and safe manner. Without the prior knowledge, there is a possibility to fall to the ground or hit the door. Usually, as the prior knowledge, we can learn the weight and friction of unknown tools by trial and error.

To make robots work in our daily-life environments, the robots need to meet two requirements. First, robots should manipulate objects including physical interaction. Second, robots should conduct work in a short time, with similar speed of that of a human. Among various possible robot manipulations, we focus on the motions to push or pull objects. In everyday life, we often perform push and pull objects, for example, opening or closing doors, refrigerators, and drawers.

Several studies on opening a door using a humanoid robot have been conducted [1, 2]. The studies were devoted to developing the criteria for balance and to using the criteria in the feedback strategy. Due to the high computational costs of the feedback strategy, the speed of the humanoid robot in opening a door is significantly slower than a human.

To accelerate the motion of the humanoid robot, we need to tackle two barriers. First, since the accelerations involved affect the balance of the robot, the control must consider the states of the robot in the near future, *e.g.*, a preview control [3]. Second, the computational cost to generate the whole-body motion should be considered. Usually, the methods used to generate stable motions first calculate the Center of Mass (CoM) trajectory of the whole body, *e.g.*, the inverse pendulum model, and then generate the whole-body motion needed to follow the CoM trajectory. This slow computation is problematic for the feedback strategy.

1.2 Related research

In this section, we discuss three key areas of related research. The first one includes several examples of studies on robot motion with physical interaction. The second area relates to various attempts to open and close a door with a robot. The third area comprises the use of a preview control for robotic motion.

Generally, generating a robot motion that includes physical interaction must consider dynamic effects, *e.g.*, force and acceleration. For example, Righetti *et al.* [4] used inverse dynamics to calculate optimized contact forces for legged robots. For stabilization of humanoids in multi-contact tasks, Ott *et al.* [5] proposed a framework for kinesthetic teaching and iterative refinement of whole body motions. Henze *et al.* [6] combined Model Predictive Control with optimization of the contact forces. Tassa *et al.* [7] proposed a modification of Differential Dynamic Programming which allowed them to incorporate control limits such as kinematic variables of the joint references. Their proposed methods are realized using a simulator or a robot with special functions such as joint torque sensors.

Several research studies have actually attempted to have a robot open or close a door, or to carry an object, either in simulations or with an actual robot. For example, Harada *et al.* [8] proposed to control the pushing motion by using a built-in walking generator and stabilizer. They used the contacting force of the hands to adjust the target Zero Moment Point (ZMP) for the stabilizer. Takubo *et al.* [9] also investigated the pushing motion. Using the RMC method and the force sensors in the hand, they controlled the ZMP with a feedback method. Murooka *et al.* [10] proposed a method to generate and execute pushing motions in various situations. Using a real humanoid robot, they succeeded in pushing large and heavy objects. To estimate the pushing force to be applied to an unknown object, they gradually increased the pushing force and planned the foot placement based on the Capture Point [11]. Arisumi *et al.* [12] analyzed a dynamic model of the door and succeeded in opening it using a hitting motion. Finally, Banerjee *et al.* [13] proposed a method for planning the motion to open a door. Using the humanoid robot ATLAS, they succeeded in pushing and walking through the door, but it took 7 minutes and 40 seconds.

Preview control is widely used for humanoid walking. The method proposed by Kajita *et al.* [14] is a seminal work in this stream of research. They tracked the ZMP using the future ZMP reference, and succeeded in generating a walking pattern on spiral stairs. Similar to our work, Wieber [15] proposed using a preview controller by assuming that the state (position, velocity, and acceleration) of the CoM after the perturbation is known. Since their method also required conversion from the CoM trajectory to whole body motion, they encountered the issue of the calculation speed of the conversion. They only showed the applicability of their method in simulation. Ibanez *et al.* [16] extended the preview control by integrating impedance control of the robot hands. Unfortunately, the simultaneous control of the hands and the CoM complicates the generation of the whole body motion much more.

1.3 Overview of the Proposed Method

In this dissertation, we propose a method to generate human like-speed stable motion for a humanoid robot to push or pull an object. To tackle the first barrier, we assume that the applied force profiles are known in advance. Based on this assumption, we calculate near-future states of the robot. We apply a preview control [3] to the motion which includes physical interactions by the assumed force direction and the displacement of the point of the effort which are perpendicular to gravity. This assumption is very similar to the assumption in walking control, where the height of the CoM is fixed [17].

To tackle the second barrier, we use analytical inverse kinematics to accelerate the calculation of the Resolved Momentum Control (RMC) [18] in every control cycle. RMC can control the robot momentum around the CoM and end-effector positions. To improve the stability of humanoid robots, high-frequency control is required. The original RMC uses Jacobian matrices for the inverse kinematics of legs and arms based on non-linear iterative optimization (*e.g.*, Newton's method). In contrast, the use of analytical inverse kinematics removes the iterative process and thus reduces the calculation burden.

1.4 Organization of this thesis

The rest of this dissertation is organized as follows. Section 2 presents the outline of the proposed method. Section 3 explains the calculation of the whole-body motion based on inverse kinematics and the RMC. Section 4 derives the conditions of the motion, such as the trajectory of the CoM and hands using prior knowledge. Section 5 explains the details of system implementation using actual objects and the humanoid robot HRP-4. Section 6 describes the experimental results for the two tasks: pushing and pulling an object. Section 7 concludes this dissertation with a brief summary and discussion of possible future work.

2. Outline of the proposed method

This section provides an overview of the proposed methodology. First, we describe the existing methods and then explain the improvements in the proposed method. The humanoid robot to be used in the experiment will also be explained.

2.1 Description of existing system

The existing system to generate whole-body motions for the humanoid robot to perform push and pull motions is shown in Fig. 1. Two things are required for the whole-body motion generation, calculating the joint angle from target values, such as position and posture of CoM, hands and feet, and calculating the target values for the motion. The calculation of joint angles is performed by inverse kinematics calculations. In order to move the robot as per the target values, it is necessary to perform the calculation within the control cycle. In addition, by calculating the target values of the poses for the hands and feet to achieve the movement, the target values are satisfied that the humanoid does not fall over. ZMP [19] or other indicators that guarantee stability must be considered. To achieve the target ZMP trajectory, the calculated target CoM trajectory is followed by the whole body of the humanoid robot. When a push action is performed, the humanoid robot can detect feedback from the object using a force sensor that is attached to the hand. The sensor values are used to update the target value to prevent the humanoid from falling over.

In the existing system, the barriers which I described in Section 1.1 can be explained in detail below:

1. Iterative calculation for the CoM and the joint angles could not be completed in the robot's control cycle.

2. Generated movement is slow because it relies on the responsiveness of the sensor feedback.

2.2 Differences from existing methods

I propose a method to generate human like-speed stable motion for a humanoid robot to push or pull an object. The proposed system is shown in Fig. 2. To accelerate the motion of the humanoid robot, we need to tackle two barriers. First, the computational cost to generate the whole-body motion should be considered. Usually, the methods used to generate stable motions first calculate the CoM trajectory of the whole body, *e.g.*, the inverse pendulum model, and then generate the whole-body motion to follow the CoM trajectory. Second, since the accelerations involved affect the balance of the robot, the control must consider the states of the robot in the near future, *e.g.*, a preview control [3].

2.2.1 Generating whole-body motion with low computational cost

To tackle the first barrier, I use analytical inverse kinematics to accelerate the calculation of the Resolved Momentum Control in every control cycle. RMC can control the robot momentum around the CoM and end-effector positions. To improve the stability of humanoid robots, high-frequency control is required. The original RMC [18] uses Jacobian matrices for the inverse kinematics of legs and arms based on non-linear iterative optimization (*e.g.*, Newton's method). In contrast, the use of analytical inverse kinematics removes the iterative process and thus reduces the calculation burden.

2.2.2 Generation of motion target using preview control

To tackle the second barrier, we assume that the applied force profiles are known in advance. Based on this assumption, I calculate near-future states of the robot. I apply a preview control [3] to the motion which includes physical interactions by the assumed force direction and the displacement of the point of the effort which are perpendicular to gravity. This assumption is very similar to the assumption in walking control, where the height of the CoM is fixed [17]. The inputs are the physical properties of the robot, such as the force and mass of the object and the coefficient of friction, which generate the target trajectory of the robot's CoM and the target position of the end-effector, such as hands or feet.

2.3 Description of the humanoid robot to be used

In this dissertation, I use a humanoid robot, HRP-4, as shown in Fig. 3. The HRP-4 is about the same size as a human. It was developed by Kawada Corporation and National Institute of Advanced Industrial Science and Technology. It is 1514 mm tall, weighs 39 kg, has 34 degrees of freedom throughout its body, and has a maximum payload of 0.5 kg on one arm [20].

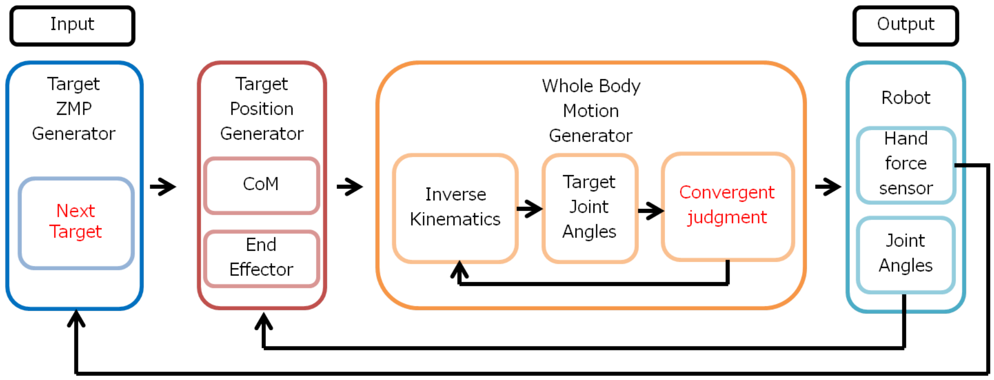


Figure 1. Motion generator for a humanoid robot of the existing system

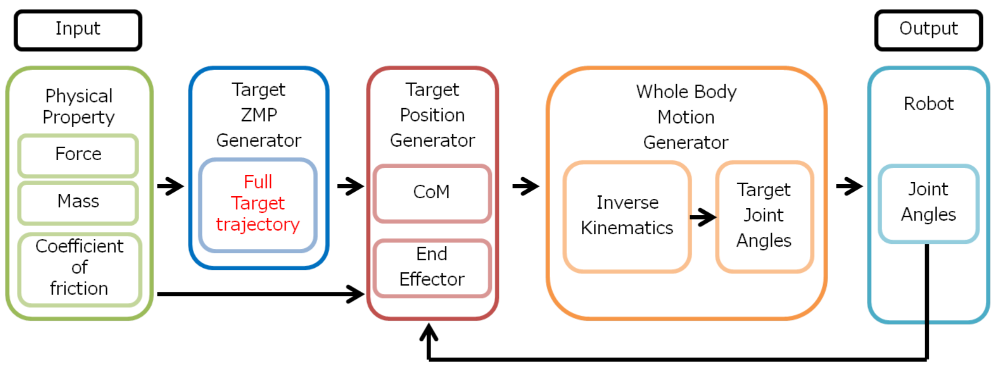


Figure 2. Motion generator for a humanoid robot of the proposed method

Figure 3. Humanoid robot HRP-4

3. Whole-body motion generator

In this section, I explain the calculation of joint angles from the target trajectory of the CoM and the hand. In order to finish the calculation of the CoM control within a robot control cycle (*e.g.*, 5 ms), I propose a method which uses analytical inverse kinematics based on a modification of the original RMC method [18]. In the RMC calculation, the numerical solution of the inverse kinematics cannot simply be replaced by the analytical inverse kinematics, so the calculation process has been modified to solve the computational issues using the analytical inverse kinematics.

3.1 Momentum equation

The RMC is a method to calculate the whole-body joint angles needed to satisfy the target positions of the end effectors and the CoM momentum. In this method, the total momentum is given by the product of the joint velocity vector and an inertial matrix which is determined by the physical and kinematic properties of the humanoid robot. Therefore, if a target value of the total momentum is given, I can calculate the joint velocities.

The humanoid's translation momentum $\boldsymbol{P}$ and the rotation momentum $\boldsymbol{L}$ are expressed as

$$\begin{bmatrix} \boldsymbol{P} \\ \boldsymbol{L} \end{bmatrix} = \begin{bmatrix} m\boldsymbol{E} & -m\hat{\boldsymbol{r}}_{\mathrm{B}\to\tilde{\mathrm{C}}} & \boldsymbol{M}_{\dot{\boldsymbol{\theta}}} \\ \boldsymbol{0} & \tilde{\boldsymbol{I}} & \boldsymbol{H}_{\dot{\boldsymbol{\theta}}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\xi}_{\mathrm{B}} \\ \dot{\boldsymbol{\theta}} \end{bmatrix}, \tag{1}$$

$$\boldsymbol{\xi}_{\mathrm{B}} \equiv \left[\boldsymbol{v}_{\mathrm{B}}^{\mathrm{T}} \ \boldsymbol{\omega}_{\mathrm{B}}^{\mathrm{T}}\right]^{\mathrm{T}},$$

where $\boldsymbol{v}_{\mathrm{B}}$ is the translational velocity of the base (waist link), $\boldsymbol{\omega}_{\mathrm{B}}$ is the rotation velocity of the base, $\dot{\boldsymbol{\theta}}$ is a vector of all joint angular velocities with n elements, $\boldsymbol{E}$ is a 3×3 identity matrix, $\boldsymbol{r}_{\mathrm{B}\to\tilde{\mathrm{C}}}$ is the vector from the base to the CoM, $\tilde{\boldsymbol{I}}$ is the 3×3 inertia matrix with respect to the CoM, and $\boldsymbol{M}_{\dot{\boldsymbol{\theta}}}$ and $\boldsymbol{H}_{\dot{\boldsymbol{\theta}}}$ are the $3 \times n$ inertia matrices which express how the joint speeds affect the linear and the angular momentum. The symbol $\hat{}$ is an operator which translates a vector into a skew-symmetric matrix.

Next, I divide the joint velocity vector $\dot{\boldsymbol{\theta}}$ by the vectors of the end effectors being controlled using Inverse Kinematics (IK) $\dot{\boldsymbol{\theta}}_{\mathrm{IK}}$ and the other vectors $\dot{\boldsymbol{\theta}}_{\mathrm{free}}$ as

$$\dot{\boldsymbol{\theta}} = \begin{bmatrix} \dot{\boldsymbol{\theta}}_{\mathrm{IK}}^{\mathrm{T}} & \dot{\boldsymbol{\theta}}_{\mathrm{free}}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}. \tag{2}$$

To generate the pushing motion, I set the joint angles of the arms and legs to $\dot{\boldsymbol{\theta}}_{\mathrm{IK}}$, and the other joints (*e.g.*, chest and neck) are set to $\dot{\boldsymbol{\theta}}_{\mathrm{free}}$. The inertial matrix can be divided in the same way. Substituting these equations into (1), I obtain

$$\begin{bmatrix} \boldsymbol{P} \\ \boldsymbol{L} \end{bmatrix} = \begin{bmatrix} m\boldsymbol{E} & -m\hat{\boldsymbol{r}}_{\mathrm{B}\to\tilde{\mathrm{C}}} & \boldsymbol{M}_{\dot{\theta}_{\mathrm{free}}} \\ \boldsymbol{0} & \tilde{\boldsymbol{I}} & \boldsymbol{H}_{\dot{\theta}_{\mathrm{free}}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\xi}_{\mathrm{B}} \\ \dot{\boldsymbol{\theta}}_{\mathrm{free}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{M}_{\dot{\theta}_{\mathrm{IK}}} \\ \boldsymbol{H}_{\dot{\theta}_{\mathrm{IK}}} \end{bmatrix} \dot{\boldsymbol{\theta}}_{\mathrm{IK}}. \tag{3}$$

The joint angle velocity $\dot{\boldsymbol{\theta}}_{\mathrm{IK}}$ is approximated from the difference between the current and target configurations of the end effectors. By giving a target momentum $\boldsymbol{P}^{\mathrm{ref}}$ and $\boldsymbol{L}^{\mathrm{ref}}$, the whole-body motion ($\boldsymbol{\xi}_{\mathrm{B}}$ and $\dot{\boldsymbol{\theta}}_{\mathrm{free}}$) can be calculated using the Pseudo-inverse. In this case, I set the target linear momentum $\boldsymbol{P}^{\mathrm{ref}}$ and the angular momenum $\boldsymbol{L}^{\mathrm{ref}}$ of CoM as

$$\boldsymbol{P}^{\mathrm{ref}} = m\boldsymbol{\xi}_{\mathrm{com}}^{\mathrm{ref}}, \quad \boldsymbol{L}^{\mathrm{ref}} = \boldsymbol{0}\,.$$

To realize a pushing motion, controlling the limbs to a specified position is necessary. Using Forward Kinematics (FK), the target position and posture of the hands and feet $\boldsymbol{p}^{\mathrm{ref}}$ are expressed as

$$\boldsymbol{p}^{\mathrm{ref}} = f_{\mathrm{FK}}(\boldsymbol{\xi}_{\mathrm{B}}, \dot{\boldsymbol{\theta}}_{\mathrm{free}}, \dot{\boldsymbol{\theta}}_{\mathrm{IK}}). \tag{4}$$

However, when solving (3) and (4), the following dilemma poses a difficulty:

- The configuration of the base link is given by the joint angles $\boldsymbol{\theta}$.
- The joint angles $\boldsymbol{\theta}_{\mathrm{IK}}$ can only be solved if the configuration of the base link is given.

To solve this problem, the original method uses the Jacobian matrix and the proposed method uses the analytical solution of inverse kinematics.

3.1.1 Solution using Jacobian

To simplify the explanation, I assumed that the robot performs a kicking motion. In this case, the legs are given a target position and posture, and the upper body generates motions to counteract the momentum of the legs.

Using the Jacobian matrix, the angular velocity $\dot{\boldsymbol{\theta}}_{IK}$ is expressed as

$$\dot{\boldsymbol{\theta}}_{leg_i} = \boldsymbol{J}_{leg_i}^{-1}\boldsymbol{\xi}_{F_i} - \boldsymbol{J}_{leg_i}^{-1}\begin{bmatrix}\boldsymbol{E} & -\hat{\boldsymbol{r}}_{B\to F_i}\\ \boldsymbol{0} & \boldsymbol{E}\end{bmatrix}\boldsymbol{\xi}_B, \tag{5}$$

$$\boldsymbol{\xi}_{F_i} \equiv \left[\boldsymbol{v}_{F_i}^{\mathrm{T}} \;\; \boldsymbol{\omega}_{F_i}^{\mathrm{T}}\right]^{\mathrm{T}},$$

where $\boldsymbol{v}_{F_i}$ is the velocity of both leg tips, $\boldsymbol{\omega}_{F_i}$ is the angular velocity, and $i = 1, 2$ is used to distinguish the left and right legs, and the Jacobian matrix (6×6) for leg position and posture is $\boldsymbol{J}_{leg_i}$, $\boldsymbol{r}_{B\to F_i}$ is the vector from the waist link to the feet, and the foot velocity and angular velocity vectors are $\boldsymbol{\xi}_{F_i}$.

From Eq. (5), it is necessary to know the angular velocity of the waist joint in order to obtain the angular velocity of the leg joint. First, by substituting Eq. (5) into Eq. (1), I obtain the following equation,

$$\begin{bmatrix}\boldsymbol{P}\\ \boldsymbol{L}\end{bmatrix} = \begin{bmatrix}\boldsymbol{M}_B^* & \boldsymbol{M}_{free}\\ \boldsymbol{H}_B^* & \boldsymbol{H}_{free}\end{bmatrix}\begin{bmatrix}\boldsymbol{\xi}_B\\ \dot{\boldsymbol{\theta}}_{free}\end{bmatrix} + \sum_{i=1}^{2}\begin{bmatrix}\boldsymbol{M}_{F_i}^*\\ \boldsymbol{H}_{F_i}^*\end{bmatrix}\boldsymbol{\xi}_{F_i}, \tag{6}$$

$$\begin{bmatrix}\boldsymbol{M}_B^*\\ \boldsymbol{H}_B^*\end{bmatrix} \equiv \begin{bmatrix}\tilde{m}\boldsymbol{E} & -\tilde{m}\hat{\boldsymbol{r}}_{B\to\tilde{C}}\\ \boldsymbol{0} & \tilde{\boldsymbol{I}}\end{bmatrix} - \sum_{i=1}^{2}\begin{bmatrix}\boldsymbol{M}_{F_i}^*\\ \boldsymbol{H}_{F_i}^*\end{bmatrix}\begin{bmatrix}\boldsymbol{E} & -\hat{\boldsymbol{r}}_{B\to F_i}\\ \boldsymbol{0} & \boldsymbol{E}\end{bmatrix},$$

$$\begin{bmatrix}\boldsymbol{M}_{F_i}^*\\ \boldsymbol{H}_{F_i}^*\end{bmatrix} \equiv \begin{bmatrix}\boldsymbol{M}_{leg_i}\\ \boldsymbol{H}_{leg_i}\end{bmatrix}\boldsymbol{J}_{leg_i}^{-1}.$$

From Eq. (6), $\boldsymbol{\xi}_{\mathrm{B}}$ and $\dot{\boldsymbol{\theta}}_{\mathrm{free}}$ are solved as Eq. (7),

$$\begin{bmatrix}\boldsymbol{\xi_B}\\ \dot{\boldsymbol{\theta}}_{\boldsymbol{free}}\end{bmatrix} = \boldsymbol{A}^{\dagger}\boldsymbol{y}, \tag{7}$$

$$\boldsymbol{y} \equiv \boldsymbol{S}\left\{\begin{bmatrix}\boldsymbol{P}^{ref}\\ \boldsymbol{L}^{ref}\end{bmatrix} - \sum_{i=1}^{2}\begin{bmatrix}\boldsymbol{M}_{F_i}^*\\ \boldsymbol{H}_{F_i}^*\end{bmatrix}\boldsymbol{\xi}_{F_i}^{ref}\right\}, \tag{8}$$

$$\boldsymbol{A} \equiv \boldsymbol{S}\begin{bmatrix}\boldsymbol{M}_B^* & \boldsymbol{M}_{free}\\ \boldsymbol{H}_B^* & \boldsymbol{H}_{free}\end{bmatrix}, \tag{9}$$

$$\boldsymbol{S} \equiv \begin{bmatrix} \boldsymbol{e}_{s_1}^{\mathrm{T}} \\ \vdots \\ \boldsymbol{e}_{s_i}^{\mathrm{T}} \end{bmatrix},$$

where $\boldsymbol{P}^{ref}$ and $\boldsymbol{L}^{ref}$ are the target momentum, $\boldsymbol{A}^{\dagger}$ is the Pseudo-inverse matrix of $\boldsymbol{A}$, $\boldsymbol{S}$ is an $i \times 6$ matrix that selects the momentum components to be controlled ($0 < i \leq 6$), and $\boldsymbol{e}_{s_i}$ is a 6×1 column vector with the elements corresponding to the s_i component of the total momentum vector set to 1 and the rest to 0. The angular velocity of the leg joint can be obtained by substituting the velocity and angular velocity of the hip link obtained into the equation (5). Finally, the whole-body state is updated by integrating the calculated whole-body joint angles $\dot{\boldsymbol{\theta}}_{free}$, $\boldsymbol{\theta}_{IK}$ and the change in the base link $\boldsymbol{\xi}_B$. The whole body motion is generated by repeating the above calculations to satisfy the target momentum and the target position and posture of inverse kinematics. The calculation process is shown in Algorithm 1 and the explanation of the symbols is shown in Table 1.

3.1.2 Solution using analytical inverse kinematics

In the proposed method, the following points are considered for speeding up the calculation.

- The inverse kinematics of each limb is geometrically solvable, and each limb is connected to the body links.
- The displacement of the joint angles and the CoM in one control cycle is small since the robot control cycle is very short (*e.g.*, 5 ms).

About the first point, the HRP-4 [20] used in the experiments has 7-Degrees-of-Freedom (DoF) arms and 6-DoF legs, but it is possible to solve their inverse kinematics analytically [21, 22]. Therefore, by fixing $\boldsymbol{\xi}_{\mathrm{B}}$ and $\boldsymbol{\theta}_{\mathrm{free}}$, $\boldsymbol{\theta}_{\mathrm{IK}}$ can be solved quickly as

$$\boldsymbol{\theta}_{\mathrm{IK}} = f_{\mathrm{IK}}(\boldsymbol{\xi}_{\mathrm{B}}, \dot{\boldsymbol{\theta}}_{\mathrm{free}}, \boldsymbol{p}^{\mathrm{ref}}), \tag{10}$$

and it achieves the target hand position accurately. Using this, I can calculate the whole-body motion more quickly.

About the second point, the base displacement $\boldsymbol{\xi}_{\mathrm{B}}$ is similar to the moving distance of the CoM $\boldsymbol{\xi}_{\mathrm{com}}^{\mathrm{ref}}$, given by the preview control, $\dot{\boldsymbol{\theta}}_{\mathrm{free}}$ is similar to zero.

In the displacement of the CoM $\boldsymbol{\xi}_{\text{com}}^{\text{ref}}$, I used z to keep the CoM height and the angular velocity to zero.

From Eq. (3), $\boldsymbol{\xi}_{\text{B}}$ and $\dot{\boldsymbol{\theta}}_{\text{free}}$ are solved as Eq. (11), given $\dot{\boldsymbol{\theta}}_{\text{IK}}$.

$$\begin{bmatrix} \boldsymbol{\xi}_{\text{B}} \\ \dot{\boldsymbol{\theta}}_{\text{free}} \end{bmatrix} = \boldsymbol{A}^{\dagger}\boldsymbol{y}, \tag{11}$$

$$\boldsymbol{y} \equiv \begin{bmatrix} \boldsymbol{P}^{\text{ref}} \\ \boldsymbol{L}^{\text{ref}} \end{bmatrix} - \begin{bmatrix} \boldsymbol{M}_{\dot{\theta}_{\text{IK}}} \\ \boldsymbol{H}_{\dot{\theta}_{\text{IK}}} \end{bmatrix} \dot{\boldsymbol{\theta}}_{\text{IK}}, \tag{12}$$

$$\boldsymbol{A} \equiv \begin{bmatrix} m\boldsymbol{E} & -m\hat{\boldsymbol{r}}_{\text{B}\to\tilde{\text{C}}} & \boldsymbol{M}_{\dot{\theta}_{\text{free}}} \\ \boldsymbol{0} & \tilde{\boldsymbol{I}} & \boldsymbol{H}_{\dot{\theta}_{\text{free}}} \end{bmatrix}, \tag{13}$$

where $\boldsymbol{r}_{\text{B}\to\tilde{\text{C}}}$ is given by the current state of joint angles $\boldsymbol{\theta}$, $\boldsymbol{A}^{\dagger}$ is the Pseudo-inverse matrix of $\boldsymbol{A}$. I use the base velocity $\boldsymbol{\xi}_{\text{B}}$ to solve $\dot{\boldsymbol{\theta}}_{\text{IK}}$ again using Eq. (10). Since the change of the $\dot{\boldsymbol{\theta}}_{\text{IK}}$ is small, the whole-body momentum is almost satisfied.

The whole body motion is calculated as follows:

1. Calculate $\dot{\boldsymbol{\theta}}_{\text{IK}}$ using Eq. (10), assuming that $\boldsymbol{\xi}_{\text{B}} = \boldsymbol{\xi}_{\text{CoM}}^{\text{ref}}$ and $\dot{\boldsymbol{\theta}}_{\text{free}} = 0$.

2. Calculate $\boldsymbol{\xi}_{\text{B}}$ and $\dot{\boldsymbol{\theta}}_{\text{free}}$ from Eq. (11)

3. Recalculate $\dot{\boldsymbol{\theta}}_{\text{IK}}$ using $\boldsymbol{\xi}_{\text{B}}$ and $\dot{\boldsymbol{\theta}}_{\text{free}}$ obtained in step 2.

In step 1, I calculate the difference of the CoM $\xi_{\text{com}}^{\text{ref}}$ using the difference of target CoM position ($\boldsymbol{p}_{\text{current}}^{\text{ref}} - \boldsymbol{p}_{\text{old}}^{\text{ref}}$). Our algorithm, shown above, excludes the iteration. Step 3 precisely controls the hand position; this is very important, since the hand contacts an object. I verify the effect of this method experimentally.

The detailed computation process is shown in Algorithm 2.

Algorithm 1 Resolved Momentum Control using Jacobian

Require: n, ε, K_p, $\boldsymbol{\xi}_{com}^{ref}$, $\boldsymbol{\xi}_{IK}^{ref}$, $\boldsymbol{p}_{B_k}$, $\boldsymbol{R}_{B_k}$, $\boldsymbol{\theta}_k$
Ensure: $\boldsymbol{p}_{B_{k+1}}$, $\boldsymbol{R}_{B_{k+1}}$, $\boldsymbol{\theta}_{k+1}$
  for $i = 0$ to n **do**
    $\boldsymbol{J}, \boldsymbol{T}_{all} \Leftarrow Forward\ Kinematics(\boldsymbol{\theta}_k, \boldsymbol{p}_{B_k}, \boldsymbol{R}_{B_k})$
    $\boldsymbol{Error} \Leftarrow Calculate\ Error\ Vector(\boldsymbol{T}_{all})$
    if $\boldsymbol{Error} < \varepsilon$ **then**
      $break$
    end if
    $\boldsymbol{M_\theta}, \boldsymbol{H_\theta}, \tilde{\boldsymbol{I}} \Leftarrow Inertia\ Matrix\ Calculation(\boldsymbol{\theta}_k, \boldsymbol{T}_{all})$
    $\boldsymbol{A} \Leftarrow \boldsymbol{p}_B, \boldsymbol{R}_B, \boldsymbol{J}, \boldsymbol{M_\theta}, \boldsymbol{H_\theta}, \tilde{\boldsymbol{I}}$
    $\boldsymbol{P}^{ref}, \boldsymbol{L}^{ref} \Leftarrow MomentumTarget(\boldsymbol{\xi}_{com}^{ref})$
    $\boldsymbol{y} \Leftarrow \boldsymbol{P}^{ref}, \boldsymbol{L}^{ref}, \boldsymbol{M_\theta}, \boldsymbol{H_\theta}, \boldsymbol{J}, \boldsymbol{\xi}_{IK}^{ref}$
    $\boldsymbol{\xi}_B, \dot{\boldsymbol{\theta}}_{free} \Leftarrow \boldsymbol{A}^\dagger \boldsymbol{y}$
    $\dot{\boldsymbol{\theta}}_{IK} \Leftarrow Numerical\ solution\ of Inverse\ Kinematics(\boldsymbol{\xi}_B, \boldsymbol{\xi}_{IK}^{ref}, \boldsymbol{Error}, K_p, \boldsymbol{J})$
    $\boldsymbol{p}_{B_{k+1}}, \boldsymbol{R}_{B_{k+1}} \Leftarrow Update(\boldsymbol{p}_{B_k}, \boldsymbol{R}_{B_k}, \boldsymbol{\xi}_B)$
    $\boldsymbol{\theta}_{k+1} \Leftarrow Update(\boldsymbol{\theta}_k, \dot{\boldsymbol{\theta}}_{IK}, \dot{\boldsymbol{\theta}}_{free})$
  end for

Algorithm 2 Resolved Momentum Control using Analytical Inverse Kinematics

Require: $\boldsymbol{\xi}_{com}^{ref}$, $\boldsymbol{\xi}_{IK}^{ref}$, $\boldsymbol{p}_{B_k}$, $\boldsymbol{R}_{B_k}$, $\boldsymbol{\theta}_k$
Ensure: $\boldsymbol{p}_{B_{k+1}}$, $\boldsymbol{R}_{B_{k+1}}$, $\boldsymbol{\theta}_{k+1}$
  $\boldsymbol{T}_{all} \Leftarrow Forward\ Kinematics(\boldsymbol{\theta}_k, \boldsymbol{p}_{B_k}, \boldsymbol{R}_{B_k})$
  $\boldsymbol{M_\theta}, \boldsymbol{H_\theta}, \tilde{\boldsymbol{I}} \Leftarrow Inertia\ Matrix\ Calculation(\boldsymbol{\theta}_k, \boldsymbol{T}_{all})$
  $\boldsymbol{A} \Leftarrow \boldsymbol{M_\theta}, \boldsymbol{H_\theta}, \tilde{\boldsymbol{I}}$
  $\boldsymbol{P}^{ref}, \boldsymbol{L}^{ref} \Leftarrow MomentumTarget(\boldsymbol{\xi}_{com}^{ref})$
  $\dot{\boldsymbol{\theta}}_{IK} \Leftarrow Analytical\ Inverse\ Kinematics(\boldsymbol{\xi}_{com}, \boldsymbol{\xi}_{IK}^{ref})$
  $\boldsymbol{y} \Leftarrow \boldsymbol{P}^{ref}, \boldsymbol{L}^{ref}, \boldsymbol{M_\theta}, \boldsymbol{H_\theta}, \dot{\boldsymbol{\theta}}_{IK}$
  $\boldsymbol{\xi}_B, \dot{\boldsymbol{\theta}}_{free} \Leftarrow \boldsymbol{A}^\dagger \boldsymbol{y}$
  $\dot{\boldsymbol{\theta}}_{IK} \Leftarrow Analytical\ Inverse\ Kinematics(\boldsymbol{\xi}_B, \boldsymbol{\xi}_{IK}^{ref})$
  $\boldsymbol{p}_{B_{k+1}}, \boldsymbol{R}_{B_{k+1}} \Leftarrow Update(\boldsymbol{p}_{B_k}, \boldsymbol{R}_{B_k}, \boldsymbol{\xi}_B)$
  $\boldsymbol{\theta}_{k+1} \Leftarrow Update(\boldsymbol{\theta}_k, \dot{\boldsymbol{\theta}}_{IK}, \dot{\boldsymbol{\theta}}_{free})$

Table 1. Parameters of Resolved Momentum Control

n	Max Iteration
ε	Tolerance
K_p	Coefficient for repeated computation
$\boldsymbol{\xi}_{com}^{ref}$	Target velocity of COM
$\boldsymbol{\xi}_{IK}^{ref}$	Target velocity for IK (Inverse Kinematics)
$\boldsymbol{p}_B$	Base position vector
$\boldsymbol{R}_B$	Base rotation matrix
$\boldsymbol{\theta}$	Angular vector
k	The number of iterations
$\boldsymbol{P}^{ref}$, $\boldsymbol{L}^{ref}$	COM's target Momentum
$\boldsymbol{J}$	Jacobian of the arms and legs
$\boldsymbol{T}_{all}$	Transition matrix of all joints
$\boldsymbol{Error}$	Position and rotation Error vector
$\boldsymbol{M_\theta}, \boldsymbol{H_\theta}, \tilde{\boldsymbol{I}}$	Inertia matrix
$\boldsymbol{\xi}_B$	Base link's velocity
$\dot{\boldsymbol{\theta}}_{IK}$	Angular velocity vector of using IK joints
$\dot{\boldsymbol{\theta}}_{free}$	Angular velocity vector of free joints

3.2 Analytical Inverse Kinematic Solutions for the Arm

An analytical inverse kinematics solution for the HRP-4 is presented to control the position and orientation of end-effectors such as hands and feet. The HRP-4 has 7 DOFs in its arms, which means that it has one redundant DOF. In order to solve for inverse kinematics analytically, I need a parameter to represent this redundant degree of freedom. Kreutz-Delgado *et al.* proposed a parameter called arm angle [21]. As shown in Fig. 4, the arm angle ψ is expressed as the angle formed by the arm plane and the reference plane, which consists of the intersection of the three shoulder joints P_s, the elbow joint P_e, and the intersection of the three wrist joints P_w. Shimizu *et al.* proposed an analytical inverse kinematics solution using arm angles and a method to obtain arm angles that satisfy the range of motion of joints for the PA-10 robot arm manufactured by Mitsubishi Heavy Industries, Ltd. [22].

In this section, I describe the method of analytical inverse kinematics using arm angles by Shimizu *et al.* , which is applied to the arm of the HRP-4.

3.2.1 Parameters of the HRP-4 arm

The axes of rotation at each joint of the arms of the HRP-4 are shown in Fig. 5. In addition, the arm parameters and rotation axis of the HRP-4 are represented

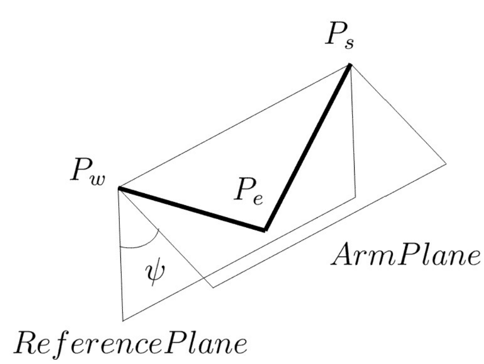


Figure 4. Arm angle ψ. P_s, P_e and P_w are the position of the shoulder, the elbow and the wrist, respectively.

using the Denavit-Hartenberg notation [23] and are shown in Table 2. The coordinate system at each joint position is set to $\Sigma_i (i = 1, 2, \ldots, 7)$. When all joint angles θ_i become 0, the orientation of each coordinate axis matches the world coordinate system Σ_0. ${}^i\boldsymbol{l}_p$ is the vector from the origin to position p in the Σ_i coordinate system, then the vector from the shoulder to the elbow ${}^3\boldsymbol{l}_{se}$, from the elbow to the wrist ${}^4\boldsymbol{l}_{ew}$, and from the wrist to the tip of the hand ${}^7\boldsymbol{l}_{wt}$ are as follows.

$$
\begin{aligned}
{}^3\boldsymbol{l}_{se} &= \begin{bmatrix} 0 & 0 & -d_{se} \end{bmatrix}^{\mathrm{T}}, \\
{}^4\boldsymbol{l}_{ew} &= \begin{bmatrix} 0 & 0 & -d_{ew} \end{bmatrix}^{\mathrm{T}}, \\
{}^7\boldsymbol{l}_{wt} &= \begin{bmatrix} 0 & 0 & -d_{wt} \end{bmatrix}^{\mathrm{T}}.
\end{aligned}
$$

The number at the top left of the vector indicates the reference coordinate system. The rotation matrix ${}^{i-1}\boldsymbol{R}_i$, corresponding to θ_i, is given by the rotation axis shown

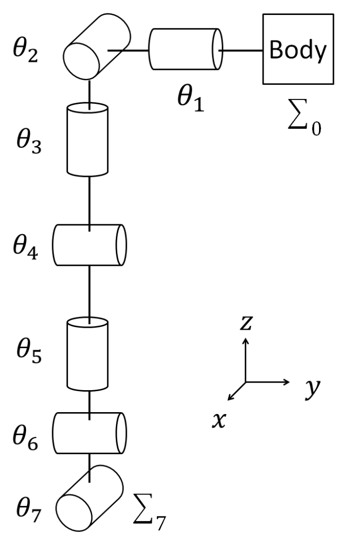


Figure 5. Rotation axes of the arm

Table 2. HRP-4's arm parameters

i	θ_i	α_i [rad]	d_i	a_i	axis
1	θ_1	$-\pi/2$	0	0	pitch
2	$\theta_2 - \pi/2$	$\pi/2$	0	0	roll
3	θ_3	$\pi/2$	d_{se}	0	yaw
4	θ_4	$-\pi/2$	0	0	pitch
5	θ_5	$\pi/2$	d_{ew}	0	yaw
6	θ_6	$\pi/2$	d_{wt}	0	pitch
7	θ_7	0	0	0	roll

in Table 2 as follows

$$
{}^{i-1}\boldsymbol{R}_i = \begin{cases} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_i & -S_i \\ 0 & S_i & C_i \end{bmatrix} & (roll) \\ \\ \begin{bmatrix} C_i & 0 & S_i \\ 0 & 1 & 0 \\ -S_i & 0 & C_i \end{bmatrix} & (pitch) \\ \\ \begin{bmatrix} C_i & -S_i & 0 \\ S_i & C_i & 0 \\ 0 & 0 & 1 \end{bmatrix} & (yaw) \end{cases},
$$

where S_i and C_i represent $\sin\theta_i$ and $\cos\theta_i$, respectively.

When the arm angle is $\theta_3 = 0$, the Arm Plane and the Reference Plane are the same. In other words, when $\theta_3 = 0$, the arm angle $\psi = 0$.

3.2.2 Derivation of the elbow angle

First, I derive the angle of the elbow joint θ_4. The vector from the shoulder to the wrist ${}^0\boldsymbol{x}_{sw}$ is expressed as

$$
{}^0\boldsymbol{x}_{sw} = {}^0\boldsymbol{x}_7^d - {}^0\boldsymbol{l}_{bs} - {}^0\boldsymbol{R}_7^d\,{}^7\boldsymbol{l}_{wt}, \tag{14}
$$

where ${}^0\boldsymbol{x}_7^d \in \boldsymbol{R}^3$ and $\boldsymbol{R}_7^d \in SO(3)$ are the target position and the target posture of the hand respectively. The vector from the shoulder to the wrist is represented by the following equation using the joint angles of the shoulder and elbow

$$
{}^0\boldsymbol{x}_{sw} \equiv {}^2\boldsymbol{R}_3^o\left({}^3\boldsymbol{l}_{se} + {}^3\boldsymbol{R}_4{}^4\boldsymbol{l}_{ew}\right). \tag{15}
$$

By calculating the sum of the squared norm on both sides, we obtain

$$
\|{}^0\boldsymbol{x}_{sw}\|^2 = \|{}^3\boldsymbol{l}_{se}\|^2 + \|{}^4\boldsymbol{l}_{ew}\|^2 + 2\left({}^3\boldsymbol{l}_{se}^{\mathrm{T}}\,{}^3\boldsymbol{R}_4{}^4\boldsymbol{l}_{ew}\right). \tag{16}
$$

From Eq. (14) and Eq. (16), the elbow angle θ_4 is expressed as

$$
\theta_4 = \cos^{-1}\left(\frac{\|{}^0\boldsymbol{x}_{sw}\|^2 - d_{se}^2 - d_{ew}^2}{2d_{se}d_{ew}}\right). \tag{17}
$$

3.2.3 Derivation of shoulder angle

Next, I derive the angle of the shoulder joint θ_1, θ_2, θ_3. In the robot coordinate system with the waist link as the origin, the vector ${}^{B}\boldsymbol{x}_{sw}$ from the shoulder to the wrist is written as

$$ {}^{B}\boldsymbol{x}_{sw} = {}^{B}\boldsymbol{R}_0 \, {}^{0}\boldsymbol{R}_1 \, {}^{1}\boldsymbol{R}_2 \, {}^{2}\boldsymbol{R}_3 \left({}^{3}\boldsymbol{l}_{se} + {}^{3}\boldsymbol{R}_4 {}^{4}\boldsymbol{l}_{ew} \right) . \tag{18} $$

Find θ_1^o and θ_2^o when the arm angle is 0. Since the arm angle ψ is zero, $\theta_3 = 0$. Multiplying both sides of the Eq. (18) by ${}^{B}\boldsymbol{R}_0^{\mathrm{T}}$, I obtain the Eq. (19).

$$ {}^{B}\boldsymbol{R}_0^{\mathrm{T}} \, {}^{B}\boldsymbol{x}_{sw} = {}^{0}\boldsymbol{R}_1^o \, {}^{1}\boldsymbol{R}_2^o \left({}^{3}\boldsymbol{l}_{se} + {}^{3}\boldsymbol{R}_4 {}^{4}\boldsymbol{l}_{ew} \right) , \tag{19} $$

$$ {}^{B}\boldsymbol{R}_0^{\mathrm{T}} \, {}^{B}\boldsymbol{x}_{sw} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} . $$

By rearranging Eq. (19), we obtain the following equation.

$$ \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} d_{ew} C_1 S_4 + \left(d_{se} + d_{ew} C_4 \right) S_1 C_2 \\ -\left(d_{se} + d_{ew} C_4 \right) S_2 \\ -d_{ew} S_1 S_4 + \left(d_{se} + d_{ew} C_4 \right) C_1 C_2 \end{bmatrix} . \tag{20} $$

From Eq. (20), since θ_4 is known, θ_2^o is calculated as

$$ x_2 = -\left(d_{se} + d_{ew} C_4 \right) S_2, $$

$$ \theta_2^o = \sin^{-1} \left(\frac{-x_2}{d_{se} + d_{ew} C_4} \right) . \tag{21} $$

In the case of $\theta_4 = 0$, from $S_4 = 0$ the following equation is obtained

$$ x_1 = \left(d_{se} + d_{ew} C_4 \right) S_1 C_2, \tag{22} $$

$$ x_3 = \left(d_{se} + d_{ew} C_4 \right) C_1 C_2. \tag{23} $$

By dividing Eq. (22) by Eq. (23), θ_1^o is calculated as

$$ \frac{x_1}{x_3} = \frac{S_1}{C_1} = \tan \theta_1^o, \tag{24} $$

$$ \theta_1^o = \tan^{-1} \left(\frac{x_1}{x_3} \right) . \tag{25} $$

In the case of $\theta_4 \neq 0$, the following equation is obtained from Eq. (20).

$$x_1 = d_{ew} C_1 S_4 + (d_{se} + d_{ew} C_4) S_1 C_2, \tag{26}$$

$$x_3 = -d_{ew} S_1 S_4 + (d_{se} + d_{ew} C_4) C_1 C_2. \tag{27}$$

From Eq. (26), C_1 is expressed as

$$C_1 = \frac{x_1 - (d_{se} + d_{ew} C_4) S_1 C_2}{d_{ew} S_4}. \tag{28}$$

Substituting into Eq. (27), θ_1^o is obtained by the following equation

$$S_1 = \frac{d_{ew} S_4 x_3 - (d_{se} + d_{ew} C_4) C_2 x_1}{-d_{ew}^2 S_4^2 - (d_{se} + d_{ew} C_4)^2 C_2^2},$$

$$\theta_1^o = \sin^{-1}\left(\frac{d_{ew} S_4 x_3 - (d_{se} + d_{ew} C_4) C_2 x_1}{-d_{ew}^2 S_4^2 - (d_{se} + d_{ew} C_4)^2 C_2^2}\right). \tag{29}$$

Then, I solve for the shoulder joint angle when the arm angle is ψ. The posture matrix by shoulder angle is expressed as

$${}^0\boldsymbol{R}_3 = \boldsymbol{A}_s \sin\psi + \boldsymbol{B}_s \cos\psi + \boldsymbol{C}_s, \tag{30}$$

where $\boldsymbol{A}_s \in \boldsymbol{R}^{3\times3}$, $\boldsymbol{B}_s \in \boldsymbol{R}^{3\times3}$ and $\boldsymbol{C}_s \in \boldsymbol{R}^{3\times3}$ are constant matrices, respectively, given by

$$\begin{aligned}
\boldsymbol{A}_s &= \hat{({}^B\boldsymbol{u}_{sw})}\ {}^0\boldsymbol{R}_3^o, \\
\boldsymbol{B}_s &= -\hat{({}^B\boldsymbol{u}_{sw})}^2\ {}^0\boldsymbol{R}_3^o, \\
\boldsymbol{C}_s &= {}^B\boldsymbol{u}_{sw}\ {}^B\boldsymbol{u}_{sw}^{\mathrm{T}}\ {}^0\boldsymbol{R}_3^o.
\end{aligned}$$

${}^B\boldsymbol{u}_{sw}$ is the unit vector from the shoulder to the wrist and $\hat{}$ represents the transformation to a skewed symmetry matrix. Where the strain target matrix $\hat{\boldsymbol{\omega}}$ of vector $\boldsymbol{\omega} = [\omega_x\ \omega_y\ \omega_z]^{\mathrm{T}}$ is given by

$$\hat{\boldsymbol{\omega}} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix},$$

and ${}^0\boldsymbol{R}_3$ is given by

$${}^0\boldsymbol{R}_3 = \begin{bmatrix} - & - & S_1 C_2 \\ C_2 S_3 & C_2 C_3 & -S_2 \\ - & - & C_1 C_2 \end{bmatrix}.$$

The components represented by - are omitted here because they are not needed in subsequent calculations. From the correspondence between the two sides of Eq. (30), we obtain the following equation

$$S_1 C_2 = a_{s13} \sin\psi + b_{s13} \cos\psi + c_{s13}, \tag{31}$$

$$C_1 C_2 = a_{s33} \sin\psi + b_{s33} \cos\psi + c_{s33}. \tag{32}$$

a_{sij}, b_{sij} and c_{sij} represent the (i, j) components of $\boldsymbol{A}_s$, $\boldsymbol{B}_s$ and $\boldsymbol{C}_s$, respectively. By dividing Eq. (31) by Eq. (32), θ_1 is expressed as

$$\begin{aligned} \tan\theta_1 &= \frac{a_{s13} \sin\psi + b_{s13} \cos\psi + c_{s13}}{a_{s33} \sin\psi + b_{s33} \cos\psi + c_{s33}}, \\ \theta_1 &= \tan^{-1}\left(\frac{a_{s13} \sin\psi + b_{s13} \cos\psi + c_{s13}}{a_{s33} \sin\psi + b_{s33} \cos\psi + c_{s33}}\right). \end{aligned} \tag{33}$$

Similarly, θ_2 and θ_3 are calculated as

$$\theta_2 = \sin^{-1}\left(-a_{s23} \sin\psi - b_{s23} \cos\psi - c_{s23}\right), \tag{34}$$

$$\theta_3 = \tan^{-1}\left(\frac{a_{s21} \sin\psi + b_{s21} \cos\psi + c_{s21}}{a_{s22} \sin\psi + b_{s22} \cos\psi + c_{s22}}\right). \tag{35}$$

3.2.4 Derivation of wrist angle

Finally, I derive the wrist angles θ_5, θ_6 and θ_7. Since the target pose is represented by the posture matrix of all the joints, the following equation can be derived.

$$\begin{aligned} {}^B\boldsymbol{R}_0\, {}^0\boldsymbol{R}_3\, {}^3\boldsymbol{R}_4\, {}^4\boldsymbol{R}_7 &= {}^0\boldsymbol{R}_7^d, \\ {}^4\boldsymbol{R}_7 &= {}^3\boldsymbol{R}_4^{\mathrm{T}}\, {}^0\boldsymbol{R}_3^{\mathrm{T}}\, {}^B\boldsymbol{R}_0^{\mathrm{T}}\, {}^0\boldsymbol{R}_7^d. \end{aligned} \tag{36}$$

Substituting Eq. (30) into Eq. (36), we obtain

$${}^4\boldsymbol{R}_7 = \boldsymbol{A}_w \sin\psi + \boldsymbol{B}_w \cos\psi + \boldsymbol{C}_w, \tag{37}$$

where $\boldsymbol{A}_w \in \boldsymbol{R}^{3\times3}$, $\boldsymbol{B}_w \in \boldsymbol{R}^{3\times3}$ and $\boldsymbol{C}_w \in \boldsymbol{R}^{3\times3}$ are constant matrices, respectively, given by

$$\begin{aligned} \boldsymbol{A}_w &= {}^3\boldsymbol{R}_4^{\mathrm{T}}\, \boldsymbol{A}_s^{\mathrm{T}}\, {}^B\boldsymbol{R}_0^{\mathrm{T}}\, {}^0\boldsymbol{R}_7^d, \\ \boldsymbol{B}_w &= {}^3\boldsymbol{R}_4^{\mathrm{T}}\, \boldsymbol{B}_s^{\mathrm{T}}\, {}^B\boldsymbol{R}_0^{\mathrm{T}}\, {}^0\boldsymbol{R}_7^d, \\ \boldsymbol{C}_w &= {}^3\boldsymbol{R}_4^{\mathrm{T}}\, \boldsymbol{C}_s^{\mathrm{T}}\, {}^B\boldsymbol{R}_0^{\mathrm{T}}\, {}^0\boldsymbol{R}_7^d, \end{aligned}$$

and ${}^4\boldsymbol{R}_7$ is given by

$$ {}^4\boldsymbol{R}_7 = \begin{bmatrix} C_5C_6 & \text{-} & \text{-} \\ S_5C_6 & \text{-} & \text{-} \\ -S_6 & C_6S_7 & C_6C_7 \end{bmatrix}. \tag{38} $$

From Eq. (37) and (38), the wrist angle is calculated as

$$ \theta_5 = \tan^{-1}\left(\frac{a_{w21}\sin\psi + b_{w21}\cos\psi + c_{w21}}{a_{w11}\sin\psi + b_{w11}\cos\psi + c_{w11}}\right), \tag{39} $$

$$ \theta_6 = \sin^{-1}\left(-a_{w31}\sin\psi - b_{w31}\cos\psi - c_{w31}\right), \tag{40} $$

$$ \theta_7 = \tan^{-1}\left(\frac{a_{w22}\sin\psi + b_{w22}\cos\psi + c_{w22}}{a_{w23}\sin\psi + b_{w23}\cos\psi + c_{w23}}\right), \tag{41} $$

where a_{wij}, b_{wij} and c_{wij} are the (i, j) components of $\boldsymbol{A}_w$, $\boldsymbol{B}_w$ and $\boldsymbol{C}_w$, respectively.

From the above, it can be seen that the joint angle of the arm can be calculated by specifying the arm angle ψ.

3.3 Analytical Inverse Kinematic Solutions for Legs

The HRP-4 has six degrees of freedom in the legs. The analytical inverse kinematics solution for legs is shown below. The axis of rotation at each joint of the leg of HRP-4 is shown in Fig. 6. The parameters using the Denavit-Hartenberg notation for the legs of the HRP-4 and the axis of rotation are shown in Table 3.

The vector from the hip position to the knee position ${}^3\boldsymbol{l}_{hk}$, the vector from the knee position to the ankle position ${}^4\boldsymbol{l}_{ka}$, and the vector from the ankle position to the foot position ${}^6\boldsymbol{l}_{af}$ are given by

$$
\begin{aligned}
{}^3\boldsymbol{l}_{hk} &= \begin{bmatrix} 0 & d_{hk_y} & -d_{hk_z} \end{bmatrix}^{\mathrm{T}}, \\
{}^4\boldsymbol{l}_{ka} &= \begin{bmatrix} 0 & 0 & -d_{ka} \end{bmatrix}^{\mathrm{T}}, \\
{}^6\boldsymbol{l}_{af} &= \begin{bmatrix} 0 & 0 & -d_{af} \end{bmatrix}^{\mathrm{T}}.
\end{aligned}
$$

3.3.1 Derivation of the hip angle θ_1

I calculate the angle of the hip joint θ_1 at the base of a leg. The target position/posture of the toe is given by ${}^0\boldsymbol{x}_6^d \in \boldsymbol{R}^3$ and ${}^0\boldsymbol{R}_6^d \in SO(3)$. As for θ_1, it is the only joint that performs Yaw axial rotation on the leg, so the Yaw axial

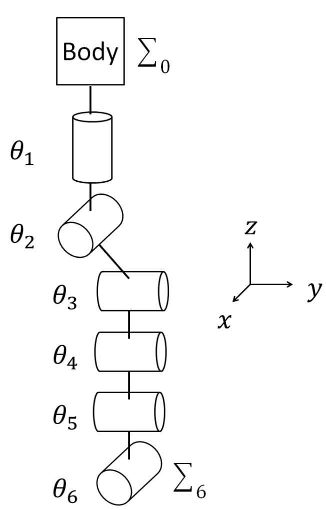


Figure 6. Rotation axes of the leg

Table 3. HRP-4's leg parameters

i	θ_i	α_i [rad]	d_i	a_i	axis
1	θ_1	$-\pi/2$	d_{fh}	0	yaw
2	θ_2	$\pi/2$	0	0	roll
3	θ_3	0	d_{hk_x}	d_{hk_y}	pitch
4	θ_4	0	0	0	pitch
5	$\theta_5 - \pi/2$	$-\pi/2$	d_{ka}	0	pitch
6	θ_6	0	d_{af}	0	roll

rotation component from the initial to the target posture is given as it is. The angular velocity vector $\boldsymbol{\omega}$ corresponding to the rotation matrix ${}^0\boldsymbol{R}_6^d$ from the initial posture to the target posture is given by

$$\boldsymbol{\omega} = \begin{cases} \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}} & (\boldsymbol{R} = \boldsymbol{E}) \\ \frac{\theta}{2\sin\theta} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix} & (\boldsymbol{R} \neq \boldsymbol{E}) \end{cases},$$

$$\boldsymbol{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix},$$

$$\theta = \cos^{-1}\left(\frac{r_{11} + r_{22} + r_{33} - 1}{2}\right).$$

The angular velocity vector $\boldsymbol{\omega}_1$ in the θ_1 coordinate system is expressed as

$$\boldsymbol{\omega}_1 = {}^B\boldsymbol{R}_0^{\mathrm{T}}\ {}^0\boldsymbol{R}_6\boldsymbol{\omega}. \tag{42}$$

Since the Yaw axis rotation component from the θ_1 coordinate system is given as the angle, the following equation is obtained

$$\theta_1 = \omega_{1z}, \tag{43}$$

where ω_{1z} is the z-axis rotational component of $\boldsymbol{\omega}_1$.

Applying Eq. (14), which finds the vector from the shoulder to the wrist, to the leg as well, the vector ${}^0\boldsymbol{x}_{ha}$ from the hip to the ankle can be expressed as

$${}^0\boldsymbol{x}_{ha} = {}^0\boldsymbol{x}_6^d - {}^0\boldsymbol{l}_{bh} - {}^0\boldsymbol{R}_6^d\ {}^6\boldsymbol{l}_{af}. \tag{44}$$

3.3.2 Derivation of the hip angle θ_2

The θ_2 is calculated using ${}^0\boldsymbol{x}_{ha}$. Let $\boldsymbol{l}_2$ be the value of ${}^0\boldsymbol{x}_{ha}$ from the θ_2 coordinate system, which is calculated as

$$\boldsymbol{l}_2 = {}^0\boldsymbol{R}_1^{\mathrm{T}}\ {}^B\boldsymbol{R}_0^{\mathrm{T}}\ {}^0\boldsymbol{x}_{ha}, \tag{45}$$

where ${}^{0}\boldsymbol{R}_1$ is obtained from the previously determined θ_1. Using the values in Table 3, θ_2 and $\boldsymbol{l}_2$ are expressed as shown in Fig. 7. θ_2 is expressed as

$$\theta_2 + \theta_a = \tan^{-1}\left(\frac{l_{2y}}{-l_{2z}}\right),$$

$$\theta_a = \sin^{-1}\left(\frac{l_{hk_y}}{\sqrt{l_{2y}^2 + l_{2z}^2}}\right),$$

$$\theta_2 = \tan^{-1}\left(\frac{l_{2y}}{-l_{2z}}\right) - \sin^{-1}\left(\frac{l_{hk_y}}{\sqrt{l_{2y}^2 + l_{2z}^2}}\right). \tag{46}$$

where l_{2y} and l_{2z} are the y and z components of $\boldsymbol{l}_2$, respectively, and l_{hky} is the y component of $\boldsymbol{l}_{hk}$.

3.3.3 Derivation of the hip angle θ_3 and knee joint θ_4

Let $\boldsymbol{l}_3$ denote ${}^{0}\boldsymbol{x}_{ha}$ from the θ_3 coordinate system, which is calculated as

$$\boldsymbol{l}_3 = {}^{1}\boldsymbol{R}_2^{\mathrm{T}}\, {}^{0}\boldsymbol{R}_1^{\mathrm{T}}\, {}^{B}\boldsymbol{R}_0^{\mathrm{T}}\, {}^{0}\boldsymbol{x}_{ha}, \tag{47}$$

where ${}^{1}\boldsymbol{R}_2$ is obtained from the previously determined θ_2 . The relationship between θ_3, θ_4 and $\boldsymbol{l}_3$ is shown in Fig. 8. Here, from Heron's formula, h is

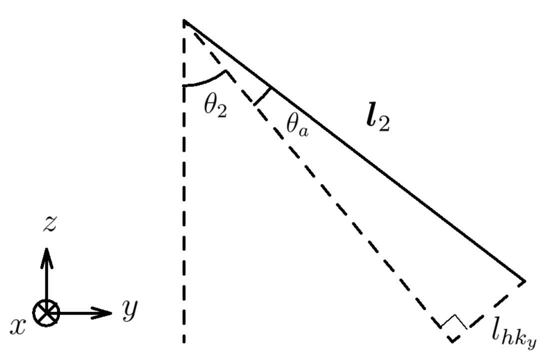


Figure 7. Hip roll angle θ_2

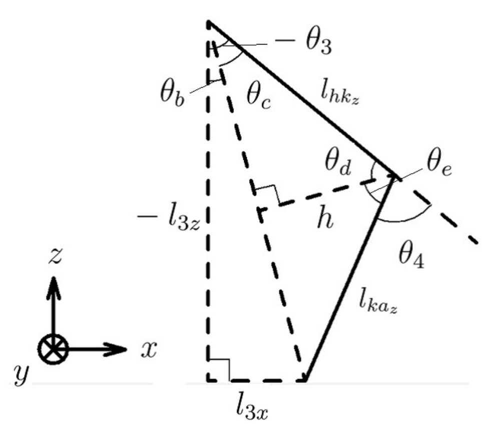


Figure 8. Hip pitch angle θ_3, Knee angle θ_4

obtained as

$$
\begin{aligned}
a &= l_{hk_z}, \\
b &= l_{ka_z}, \\
c &= \sqrt{l_{2x}^2 + l_{2z}^2}, \\
s &= \frac{a+b+c}{2}, \\
S &= \sqrt{s(s-a)(s-b)(s-c)}, \\
h &= \frac{2S}{c}.
\end{aligned}
$$

From the above, θ_3 and θ_4 are given by

$$
\begin{aligned}
\theta_3 &= -(\theta_b + \theta_c) \\
&= -\tan^{-1}\left(\frac{l_x}{-l_z}\right) - \sin^{-1}\left(\frac{h}{l_{hk_z}}\right), \qquad (48) \\
\theta_4 &= \pi - \theta_d - \theta_e \\
&= \pi - \cos^{-1}\left(\frac{h}{l_{hk_z}}\right) - \cos^{-1}\left(\frac{h}{l_{ka_z}}\right). \qquad (49)
\end{aligned}
$$

3.3.4 Derivation of the ankle joint θ_5 and θ_6

Since the target pose is represented by the posture matrix of all the joints, the following equation can be derived.

$$
\begin{aligned}
{}^B\boldsymbol{R}_0\, {}^0\boldsymbol{R}_4\, {}^4\boldsymbol{R}_6 &= {}^0\boldsymbol{R}_6^d, \\
{}^4\boldsymbol{R}_6 &= {}^0\boldsymbol{R}_4^{\mathrm{T}}\, {}^B\boldsymbol{R}_0^{\mathrm{T}}\, {}^0\boldsymbol{R}_6^d = \begin{bmatrix} r_{11}^d & r_{12}^d & r_{13}^d \\ r_{21}^d & r_{22}^d & r_{23}^d \\ r_{31}^d & r_{32}^d & r_{33}^d \end{bmatrix}. \qquad (50)
\end{aligned}
$$

${}^4\boldsymbol{R}_6$ is given by

$$
{}^4\boldsymbol{R}_6 = \begin{bmatrix} C_5 & - & - \\ - & C_6 & -S_6 \\ -S_5 & - & - \end{bmatrix}. \qquad (51)
$$

From Eq. (50) and Eq. (51), θ_5 and θ_6 are determined by

$$
\theta_5 = \tan^{-1}\left(\frac{-r_{31}^d}{r_{11}^d}\right), \qquad (52)
$$

$$\theta_6 = \tan^{-1}\left(\frac{-r_{23}^d}{r_{22}^d}\right). \tag{53}$$

4. Deriving the conditions of the motion

This section explains our proposed method to calculate the target trajectory of the CoM and the end-effector (hands and feet) using physical quantities such as the force applied to the object, the mass of the object, and the friction coefficient. First, using the preview control, I generate the target CoM trajectory from the pushing force and target ZMP. Then, using the target force needed to move the object, I calculate the target acceleration of the hands. I assume that the humanoid robot does not take a step while pushing an object.

4.1 Preview control in pushing motion

The preview control provides the optimal control for a linear system [3]. This means that the dynamics of pushing an object must be formulated in a linear form. Assuming that the humanoid has a simple mass point as shown in Fig. 9, the ZMP in the pushing direction ZMP_x is expressed as follows [9]:

$$ZMP_x = \frac{-z_{\mathrm{com}}\dot{P}_x - \tau_{\mathrm{hand}} + x_{\mathrm{com}}mg + x_{\mathrm{hand}}F_{\mathrm{hand_z}} + z_{\mathrm{hand}}F_{\mathrm{hand_x}}}{\dot{P}_z - F_{\mathrm{hand_z}} + mg}, \tag{54}$$

where P is the translational momentum of the CoM, τ is the moment, m is the total mass of the humanoid robot, g is the gravitational acceleration, F is the reaction force, and x, y, z are the position in the corresponding axis. I only consider linear (over the x axis) pushing/pulling motions. In this case, I safely assume that the height of the hands and the CoM is constant, that the mass is concentrated at one point (*i.e.*, no inertial moment), and that no torque is exerted by the hands. In summary, the following equations are satisfied.

$$F_{\mathrm{hand_z}} = 0, \tag{55}$$

$$P_{\mathrm{z}} = 0, i.e., \dot{P}_{\mathrm{z}} = 0, \tag{56}$$

$$\tau_{\mathrm{hand}} = 0, \tag{57}$$

By substituting these conditions into Eq. (54), I obtain Eq. (58),

$$ZMP_x = -\frac{z_{\mathrm{com}}}{g}\ddot{x}_{\mathrm{com}} + x_{\mathrm{com}} + \frac{z_{\mathrm{hand}}}{mg}F_{\mathrm{hand}}, \tag{58}$$

where $\dot{P}_x = m\ddot{x}_{\text{com}}$ and I simply describe F_{hand_x} as F_{hand}. From these conditions, the denominator of the right hand side of Eq. (54) becomes constant, *i.e.*, mg. Thus, Eq. (54) becomes linear.

This formalization tells us one important thing. The hand position in the pushing direction does not appear in this equation. This means that the position of the hand does not affect the whole body balance. This relaxes the choice of the hand control, *e.g.*, using sensor feedback control.

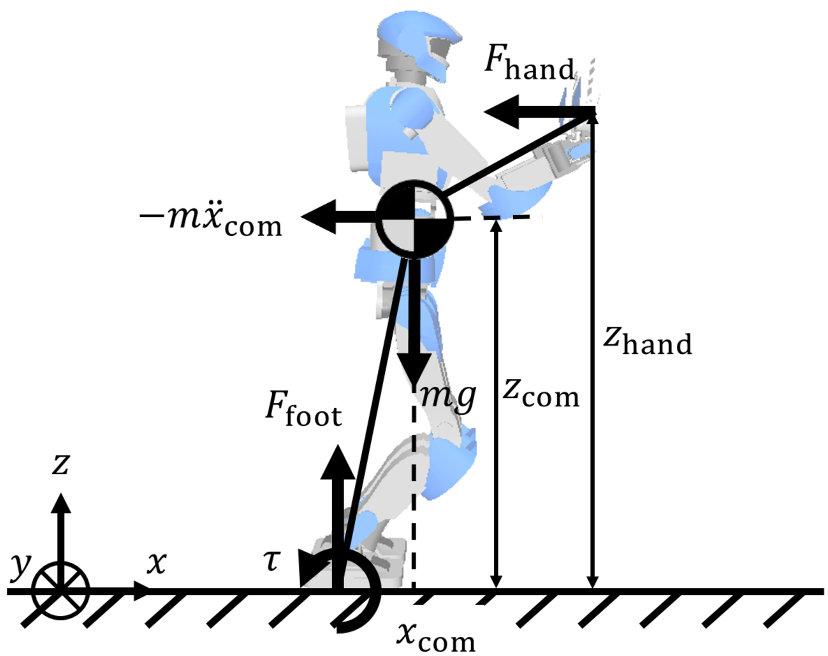


Figure 9. Simple physical model for the horizontal pushing and pulling motion: F_{hand} and F_{foot} are reaction forces generated at the hand and the two feet, respectively. τ_{foot} is the moment generated at the two feet, x_{com} is the CoM position in the x axis direction, z_{com} and z_{hand} are the height of the CoM and the hand, respectively. m is the total mass of the humanoid, and g is the gravitational acceleration.

4.2 Generate CoM trajectory using preview control

To use preview control, Eq. (58) is transformed as

$$
\begin{aligned}
y &\equiv ZMP_x - \frac{z_{\text{hand}}}{mg} F_{\text{hand}}\,, \\
&= -\frac{z_{\text{com}}}{g} \ddot{x}_{\text{com}} + x_{\text{com}}.
\end{aligned} \tag{59}
$$

If y is given, the optimal x_{com} control is derived from the preview controller. In this dissertation, I assume the robot does not take a step, *i.e.*, $ZMP_x = 0$, that is, the center of foot sole. Also, I assume that F_{hand} is planned in advance. Note that it is theoretically possible to apply our proposed method in the case where the robot takes a step.

I apply the preview control to the pushing motion. The input u of the control is

$$
u = \dddot{x}_{\text{com}}, \tag{60}
$$

where $\dddot{x}_{\text{com}}$ is the jerk of the CoM. Eq. (61) then formulates a system of simplified model's dynamics.

$$
\begin{cases}
\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{b}u_k \\
y = \boldsymbol{c}\boldsymbol{x}_k
\end{cases}, \tag{61}
$$

where

$$
\boldsymbol{x}_k \equiv \begin{bmatrix} x_{\text{com}}(k\Delta t) & \dot{x}_{\text{com}}(k\Delta t) & \ddot{x}_{\text{com}}(k\Delta t) \end{bmatrix}^{\mathrm{T}},
$$

$$
u_k \equiv u(k\Delta t),
$$

$$
F_{\text{hand},k} \equiv F_{\text{hand}}(k\Delta t),
$$

$$
\boldsymbol{A} \equiv \begin{bmatrix} 1 & \Delta t & \Delta t^2/2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{b} \equiv \begin{bmatrix} \Delta t^3/6 \\ \Delta t^2/2 \\ \Delta t \end{bmatrix},
$$

$$
\boldsymbol{c} \equiv \begin{bmatrix} 1 & 0 & -z_{\text{com}}/g \end{bmatrix},
$$

where k is the number of control steps, and Δt is the sampling time. The performance index J is defined as

$$
J = \sum_{j=1}^{\infty} \left\{ Q(F^{\text{ref}}_{\text{hand},j} - F_{\text{hand},j})^2 + Ru_j^2 \right\}, \tag{62}
$$

where Q and R are positive weights. The $F^{\mathrm{ref}}_{\mathrm{hand},k}$ is input from the reference force's trajectory. The index J can be minimized by using the controller, which gives us the optimal solution. I obtain the target trajectory of the CoM x_{com} and the force F_{hand} from the sequential computation of Eq. (61).

4.3 Generate hand motion

The hand trajectory needs to satisfy the equilibrium of forces. The target object follows simple dynamics (Newton's law) based on the pushing force and friction. In this dissertation, I assumed the static and dynamic friction are the same and calculated the force to move an object to simplify the pushing model for the whole-body motion generation. After moving an object, the equilibrium of the forces is expressed by

$$F_{\mathrm{hand}} = -\mu M g - M a, \tag{63}$$

where M is the mass of the target object and μ is the coefficient of friction. From Eq. (61), I can obtain the force exerted by the hand at time k, as

$$F_{\mathrm{hand},k} = -\frac{mg}{z_{\mathrm{hand}}}\boldsymbol{c}\boldsymbol{x}_k. \tag{64}$$

Finally, I obtain the hand acceleration at time k, a_k as follows:

$$a_k = -\mu g - \frac{F_{\mathrm{hand},k}}{M}. \tag{65}$$

I can safely assume the initial state of the target object, that is, that the position is known and the velocity is zero.

5. Implementation

5.1 System configuration

I conducted experiments using the humanoid robot HRP-4 [20]. The HRP-4 is 1.514 m height and 39 kg weight. Its software runs on a PC with a 1.6 GHz Intel Pentium M CPU, with ARTLinux as the operating system. The control system is built on the OpenRTM-aist [24] middleware, which our system also uses. This middleware follows the component-oriented programming paradigm which connects blocks called RT-components with specific functionality to form complex systems.

Fig. 10 shows the system configuration used for the experiments. Before the experiment starts, I calculate the target trajectory of the CoM x_{com} from the predefined force profile $F^{\mathrm{ref}}_{\mathrm{hand}}$ using the preview control. During the operation, I calculate the target position of the hand $\boldsymbol{p}^{\mathrm{ref}}_{hand}$, and generate the whole body motion within a robot control cycle. The target position of the hands $\boldsymbol{p}^{\mathrm{ref}}_{hand}$, is calculated from the acceleration a^{ref} and the velocity v (Eq. 12). Due to small deviations caused by the control, I cannot assume that the robot follows the ideal hand trajectory. The acceleration is directly related to the force applied by the hand. Because of this, I decided to control the robot hands with respect to the acceleration. The velocity can be calculated from the difference between two time-sequential hand positions $\boldsymbol{p}_{hand}$, which are obtained from the forward kinematics of the HRP-4. Finally, I update the joint angle of the whole body using these target values.

5.2 Physical properties of the target objects

The physical properties of the target objects used for the input are the force required to push the object F_{max}, the mass M and the coefficient of static friction μ. Before conducting experiments with a real robot, I measured the physical properties of the object used in the experiments.

To avoid generating excessive torques in the motors of the robot, I set the profile of the force to change the force smoothly as follows

$$F^{\mathrm{ref}}_{\mathrm{hand}} = F_{\mathrm{max}} \sin \frac{\pi}{2t_{\mathrm{max}}} t, \tag{66}$$

where $F_{\text{hand}}^{\text{ref}}$ is the target force exerted by the hands, t is the elapsed time, and the value $t_{\max}$ is set experimentally.

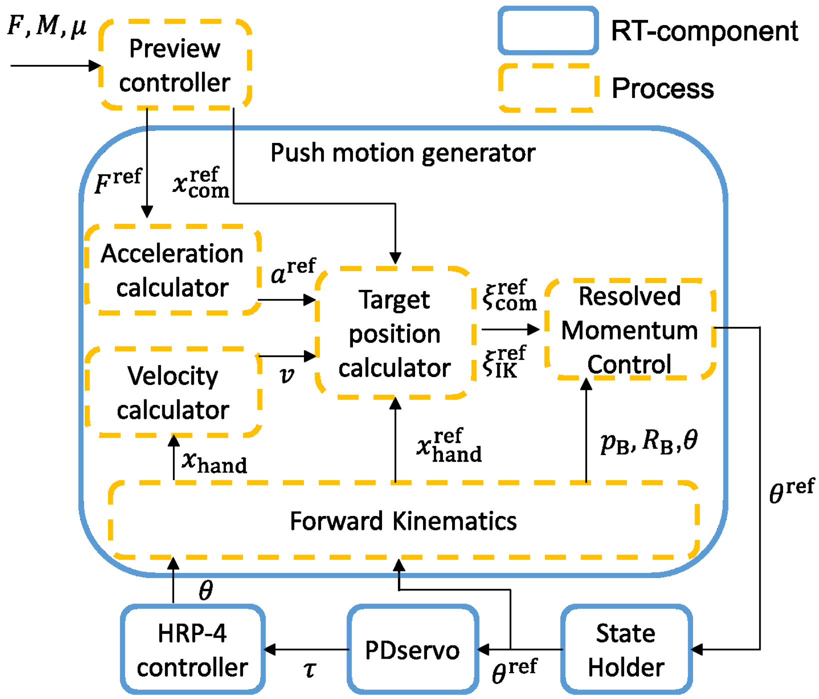


Figure 10. System to generate whole-body motion. F is the force exerted by the hand, M is the mass, μ is the static friction coefficient, a is acceleration, v is velocity, $\boldsymbol{p}$ is the pose, $\boldsymbol{\tau}$ is the torque, $\boldsymbol{\theta}$ represents the joint angles, and ref denotes the reference value.

6. Experiments

In this section, I present the conducted experiments where I apply the proposed method to the tasks of pushing and pulling an object. In these two tasks, the forces exerted by the hand are in opposite directions. First, I compared the computational time of the original RMC and our proposed method using analytical inverse kinematics. In these experiments with the real robot, I did not use the ZMP feedback control. I only used the ZMP to evaluate the stability of the robot. I experimented both with pushing a box and with pulling a door. Furthermore, I analyzed in more detail the generated motion. I confirmed the generated motion and the ZMP trajectory for pushing objects of different weights using the simulator, and we measured the force generated by the pushing motion using a force sensor.

6.1 Experiments of Whole-body motion generator

Experiments on whole-body motion generation using the proposed method was conducted. To calculate $\dot{\boldsymbol{\theta}}_{\mathrm{IK}}$ using Eq. (10), we assume that $\boldsymbol{\xi}_{\mathrm{B}} = \boldsymbol{\xi}^{\mathrm{ref}}_{\mathrm{CoM}}$ and $\dot{\boldsymbol{\theta}}_{\mathrm{free}} = 0$. The effect of this assumption is confirmed by experiment. First, I compared kicking motion with the numerical solutions of inverse kinematics and the proposed method. Next, I compared the original RMC and the proposed method using analytical inverse kinematics.

I compared the two methods using a kicking motion, as was done in [18]. This motion reduces the dimension of $\dot{\boldsymbol{\theta}}_{\mathrm{IK}}$ and thus the computation of the original RMC becomes stable. The flow of the motion is shown in Table 4. The translational components of the target momentum $\boldsymbol{P}^{ref}$ and the rotational component $\boldsymbol{L}^{ref}$ are given by

$$P^{ref}_{x,y} = \tilde{m} K_p (\tilde{c}^{\,ref}_{x,y} - \tilde{c}_{x,y}), \tag{67}$$

$$P^{ref}_{z} = \tilde{m} K_p (z^{ref}_{B} - z_B), \tag{68}$$

$$\boldsymbol{L}^{ref} = \boldsymbol{0}, \tag{69}$$

where K_p is the feedback gain, $\tilde{c}$ is the position of the CoM, z_B is the waist, ^{ref} is the target value, and the subscripts x. y and z are components in each direction. z^{ref}_B is the initial height of the base link, $\tilde{c}^{\,ref}$ is given to move onto the axial foot between 0-1 s and then keep it there. In addition, the initial position and orientation of the feet were given as target values to maintain contact with the ground.

To calculate the kicking motion, the joint velocity vector $\dot{\boldsymbol{\theta}}_{\mathrm{IK}}$ and $\dot{\boldsymbol{\theta}}_{\mathrm{free}}$ are set to

$$\dot{\boldsymbol{\theta}}_{\mathrm{IK}} = \left[\dot{\boldsymbol{\theta}}^{\mathrm{T}}_{\mathrm{leg_R}} \dot{\boldsymbol{\theta}}^{\mathrm{T}}_{\mathrm{leg_L}}\right]^{\mathrm{T}},$$

$$\dot{\boldsymbol{\theta}}_{\mathrm{free}} = \left[\dot{\boldsymbol{\theta}}^{\mathrm{T}}_{\mathrm{arm_R}} \dot{\boldsymbol{\theta}}^{\mathrm{T}}_{\mathrm{arm_L}} \dot{\boldsymbol{\theta}}^{\mathrm{T}}_{\mathrm{body}}\right]^{\mathrm{T}},$$

where $\dot{\boldsymbol{\theta}}_{\mathrm{leg}}$ is the vector of the leg joint angle velocity, $\dot{\boldsymbol{\theta}}_{\mathrm{arm}}$ is the vector of the arm joint angle velocity, the subscripts R and L mean the right side and left side, and $\dot{\boldsymbol{\theta}}_{\mathrm{body}}$ denotes the other joint angle velocities, *i.e.*, those of the chest and the neck.

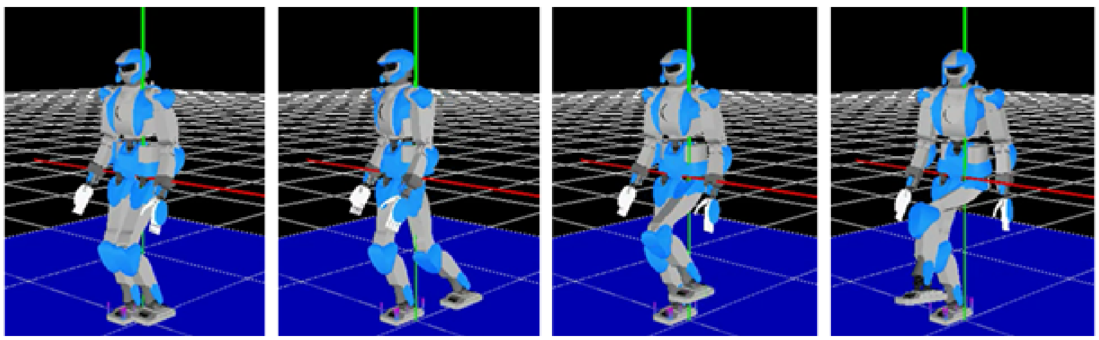
Table 4. Kicking motion procedure

Time [s]	Motion
$0 \sim 1$	Move the COM on the right foot
$1 \sim 2$	Raise the left leg vertically
$2 \sim 3$	Move the left leg behind
$3 \sim 4$	Kick using the left leg

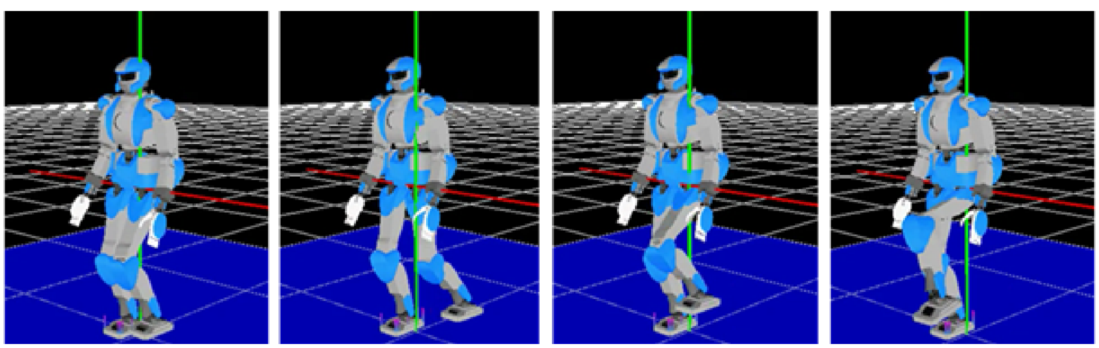
6.1.1 Comparison to the numerical solutions of inverse kinematics

Fig. 11 is the result of the kicking motion. In the case of RMC, we can confirm that the upper body generates a twisting motion in the direction to counteract the recoil of the kick. In order to compare the stability of the two operations, the change in ZMP during operation is shown in Fig. 12.

With respect to 4 s on wards after the kicking motion, the ZMP deflection is about 0.08 m in the x direction and about 0.04 m in the y direction for RMC, whereas when only inverse kinematics is used, it is about 0.15 m in the x direction and about 0.07 m in the y direction. It can be seen that the deflection of the ZMP is smaller when the RMC is used. Although the ZMP was not directly controlled, it was confirmed that the ZMP oscillation range was reduced and the stability was improved.

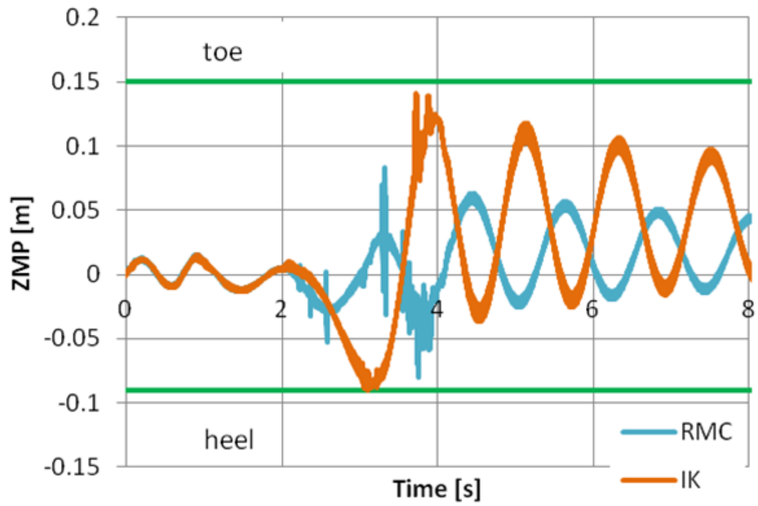


(a) Kicking motion using Resolved Momentum Control

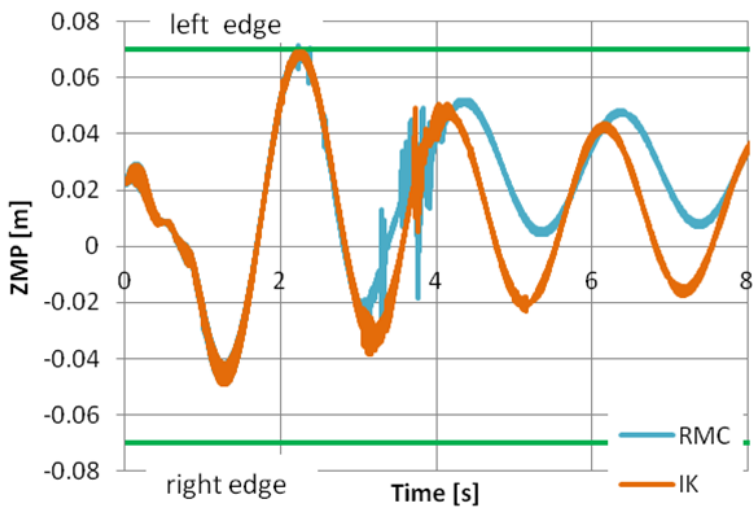


(b) Kicking motion only using Inverse Kinematics

Figure 11. Kicking motion



(a) ZMP position in x direction (sagittal direction)



(b) ZMP position in y direction (lateral direction)

Figure 12. ZMP position during kicking

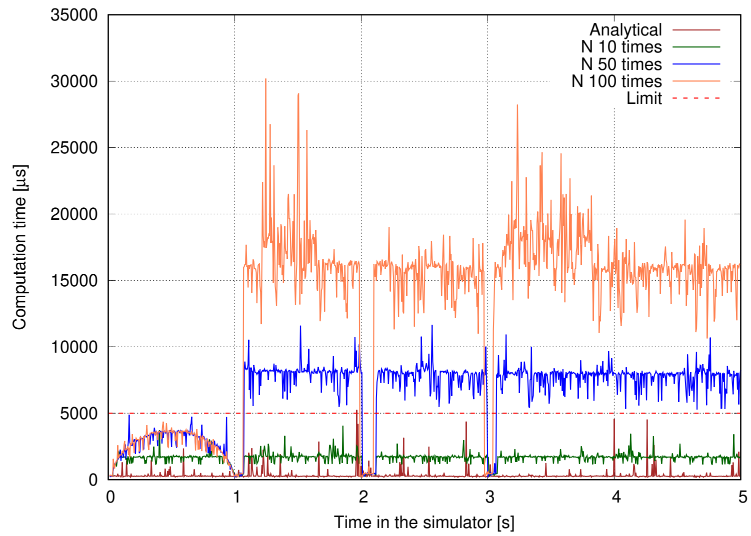
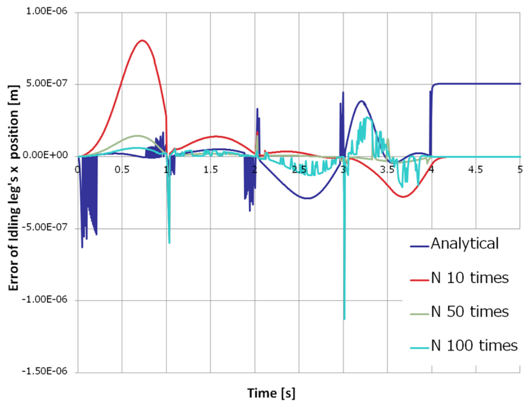


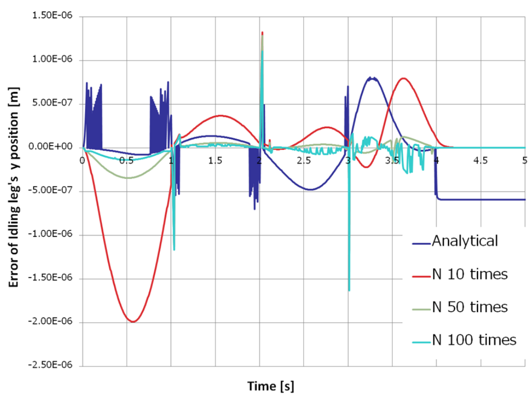
Figure 13. Comparison of computational time between analytical and numerical solutions of inverse kinematics in a kicking motion. In "N x times," x is the maximum number of iterations used in the numerical solution of inverse kinematics.

6.1.2 Comparison to the original RMC method

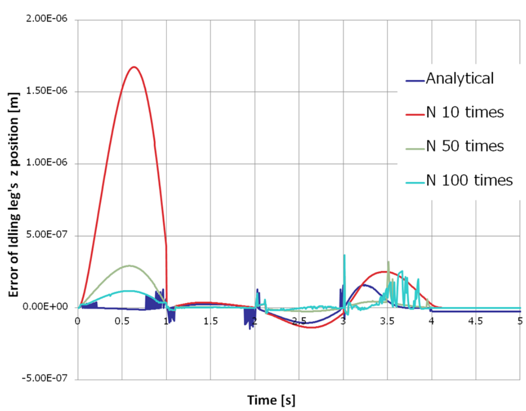
Fig. 13 shows the computation times for the proposed and the original method of RMC. For the original method, I set a feedback gain of 0.1 and the number of maximum iterations of 10, 50, and 100. The computation time of the proposed method was shorter than the original method. The HRP-4's control cycle is 5 ms, and the average computation time with our method is about 0.3 ms. Therefore, the calculation is finished with sufficient margin left to be used in the control cycle, showing that this method can be stably applied to the actual robot.



(a) Error of swing leg position in x direction (sagittal direction)



(b) Error of swing leg position in y direction (lateral direction)



(c) Error of swing leg position in z direction (vertical direction)

Figure 14. Error of swing leg position during kicking

Table 5. Maximum error of swing leg position

(a) Maximum error of swing leg position in x direction (sagittal direction)

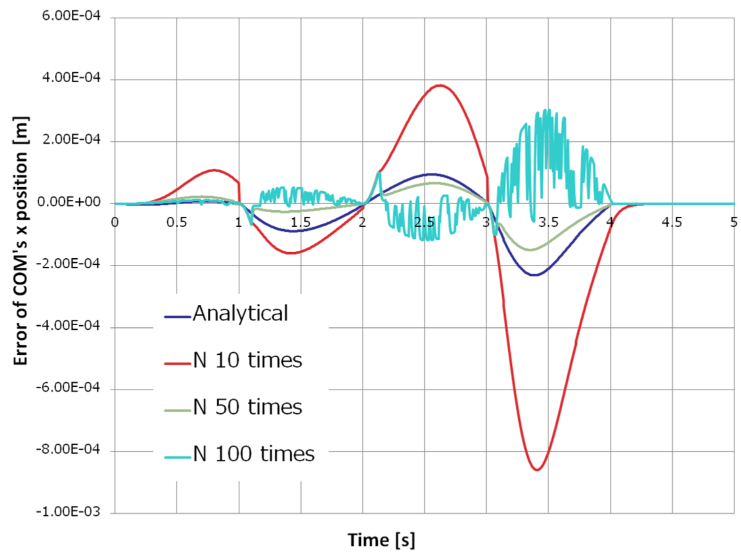
		Time [s]			
		0-1	1-2	2-3	3-4
Error [nm]	Analytical IK	628.6	377.0	370.1	452.8
	Numerical 10 times	805.6	583.2	172.9	797.9
	Numerical 50 times	144.6	597.1	144.3	789.3
	Numerical 100 times	60.7	599.2	128.7	1127.0

(b) Maximum error of swing leg position in y direction (lateral direction)

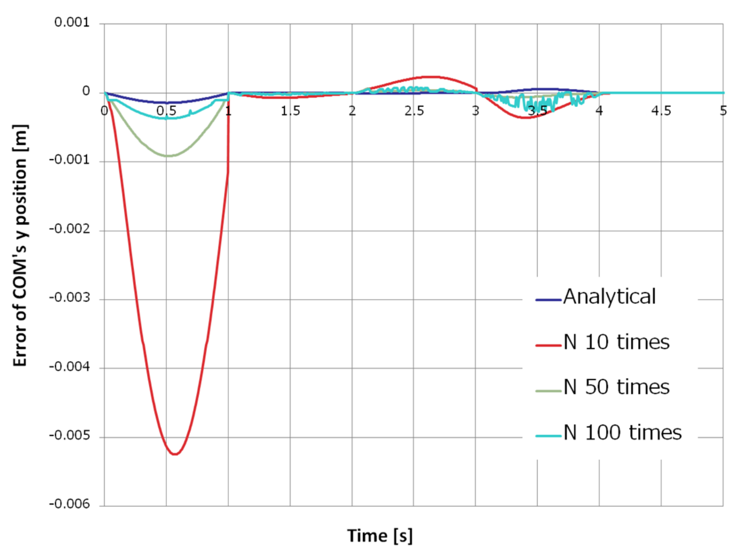
		Time [s]			
		0-1	1-2	2-3	3-4
Error [nm]	Analytical IK	756.6	701.2	616.6	807.3
	Numerical 10 times	1989.0	1130.6	1325.8	1175.4
	Numerical 50 times	344.5	1160.6	1282.8	1174.2
	Numerical 100 times	139.1	1164.7	1105.7	1631.9

(c) Maximum error of swing leg position in z direction (vertical direction)

		Time [s]			
		0-1	1-2	2-3	3-4
Error [nm]	Analytical IK	130.1	147.2	129.6	161.0
	Numerical 10 times	1673.9	429.4	137.1	252.3
	Numerical 50 times	292.1	37.3	63.3	322.1
	Numerical 100 times	118.6	37.5	100.4	366.6



(a) Error of COM position in x direction (sagittal direction)



(b) Error of COM position in y direction (lateral direction)

Figure 15. Error of COM position during kicking

Table 6. Maximum error of COM position

(a) Maximum error of COM position in x direction (sagittal direction)

		Time [s]			
		0-1	1-2	2-3	3-4
Error [μm]	Analytical IK	7.9	88.6	94.4	231.1
	Numerical 10 times	107.7	160.2	381.9	859.4
	Numerical 50 times	22.9	67.8	99.1	149.3
	Numerical 100 times	11,1	67.9	117.9	302.0

(b) Maximum error of COM position in y direction (lateral direction)

		Time [s]			
		0-1	1-2	2-3	3-4
Error [μm]	Analytical IK	142.7	4.0	6.2	57.2
	Numerical 10 times	5248.0	1149.9	235.1	359.3
	Numerical 50 times	915.9	98.9	61.2	63.8
	Numerical 100 times	372.6	98.9	80.8	271.7

Fig. 14 and Fig. 15 show the difference between the target position of the CoM and the position of the kicking foot in the robot's coordinate system and the actual position, respectively. In addition, The maximum error values per second, which is the interval between actions, are summarized in Table 5 and Table 6. The maximum difference in the x-direction of the CoM is reached between 3-4 s when the leg is kicked out. The value of 231.1 μm using the analytical solution method and 859.4 μm with 10 iterations using the numerical solution method were found to be more stable. Only when the number of iterations was 50 was it more accurate than the analytical solution method. For the entire operation, the same accuracy as that of 50 iterations was achieved with a computation time of about 1/30. In addition, when the calculation was repeated 100 times, there were some oscillations in the calculation results, and it was not always possible to calculate accurately. For the difference in the y direction of the center of gravity, the proposed method shows the best accuracy. As for the position of the foot, the difference is in units of nm and it can be said to have a better tracking ability than the center of gravity. In this method, it is considered that a difference of about mm in pushing motions is sufficient. Therefore, it is considered to be sufficiently accurate to perform the operation.

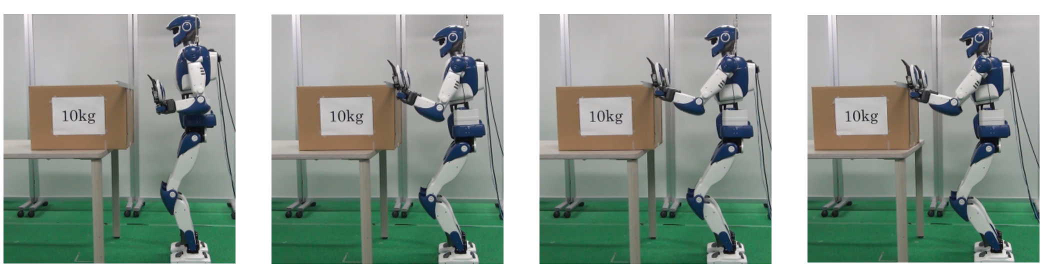
6.2 Experiments with real robot

6.2.1 Pushing task with a real robot

In our experiment with a real robot, I used a 10 kg box with a coefficient of static friction of 0.3 and it required 3 kgf force to move the box. I set the following target values:

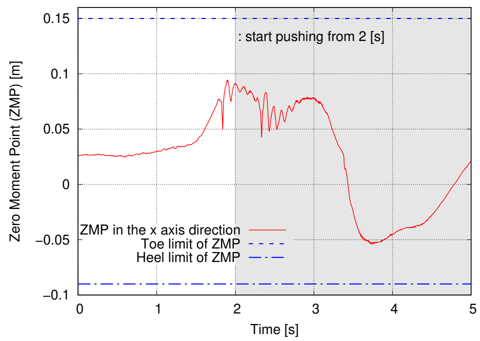
- Maximum force $F_{\max} = 0.3$ kgf.
- Time until the object moved $t_{\max} = 2.0$ s.
- Target height of the CoM $z_{\mathrm{com}} = 0.76$ m.
- Height of the hands $z_{\mathrm{hand}} = 1.23$ m.

Fig. 16 shows the generated motion of the real robot, which succeeded in pushing the box without falling. From $t = 0$ s to $t = 2$ s, the HRP-4 moved the hands toward the box and then started pushing. At $t = 4$ s, the HRP-4 is still pushing the box, and, at $t = 6$ s, finished the motion. Fig. 17 shows the ZMP trajectory of this motion. In the pushing direction, the x axis direction, there was an oscillation of the ZMP at around $t = 2$ s. This oscillation was caused by the impact of the collision with the box. However, the ZMP stays inside the feet. The target of the CoM is to control the CoM trajectory following the planned trajectory. Fig. 18 shows the CoM trajectory of this motion. From $t = 0$ s to $t = 2$ s, the CoM moved to the starting position for the pushing motion. After $t = 2$ s, the CoM followed the target trajectory until the motion was completed.

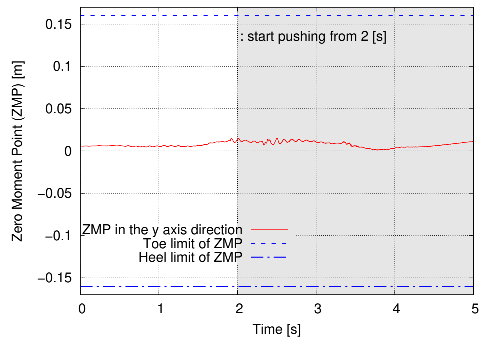


0 s: Set the hands 2 s: Start the pushing 4 s: Push the box 6 s: Finish motion

Figure 16. Generated motion for pushing a 10 kg box using the HRP-4. From $t = 0$ s to $t = 2$ s, the HRP-4 moves the hands toward the box. From $t = 2$ s to $t = 6$ s, the HRP-4 pushes the box.



(a) ZMP position in the x axis direction



(b) ZMP position in the y axis direction

Figure 17. ZMP trajectory during the task of pushing a 10 kg box

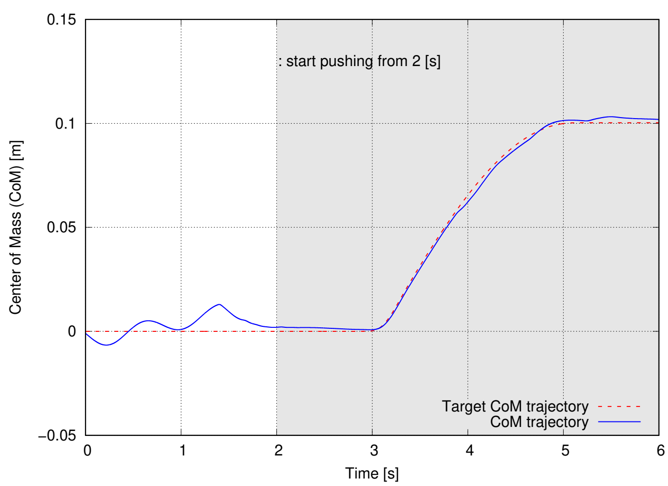


Figure 18. CoM trajectory during the pushing motion in the x axis direction

6.2.2 Pulling task with a real robot

I also experimented with opening a refrigerator door which requires a pulling motion. For this experiment, I attached a simple hook to the HRP-4's hands because the power of the finger motors is too weak to open the door. The refrigerator used in the experiments is shown in Fig. 19. In addition, a simple hook is shown in Fig. 20

Since the difference between the pushing and pulling motions is only the direction of the motion, the pulling motion can be easily generated by reversing the force direction of the pushing motion. The door of the refrigerator was closed using magnetic force and it required 1.5 kgf to open the door. This is about the same force as that needed to push a 5 kg box with a coefficient of static friction of 0.3. The force is needed only at the moment of opening it. To open the door, I generated a 0.03 m pulling motion. Since the distance of this motion is short, the motion can be assumed to be almost straight, so the influence of the rotation on the CoM can be neglected. After opening the door, I also solved the inverse kinematics of the arm to follow the door radius trajectory, and controlled the center of gravity to not move because the door can be opened using only a small force.

Figure 19. Refrigerator

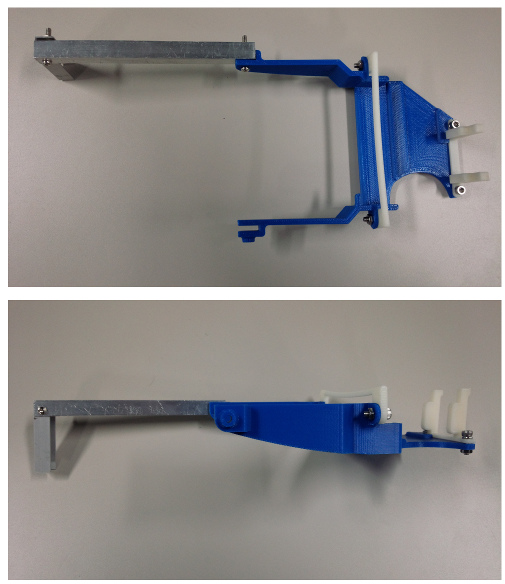
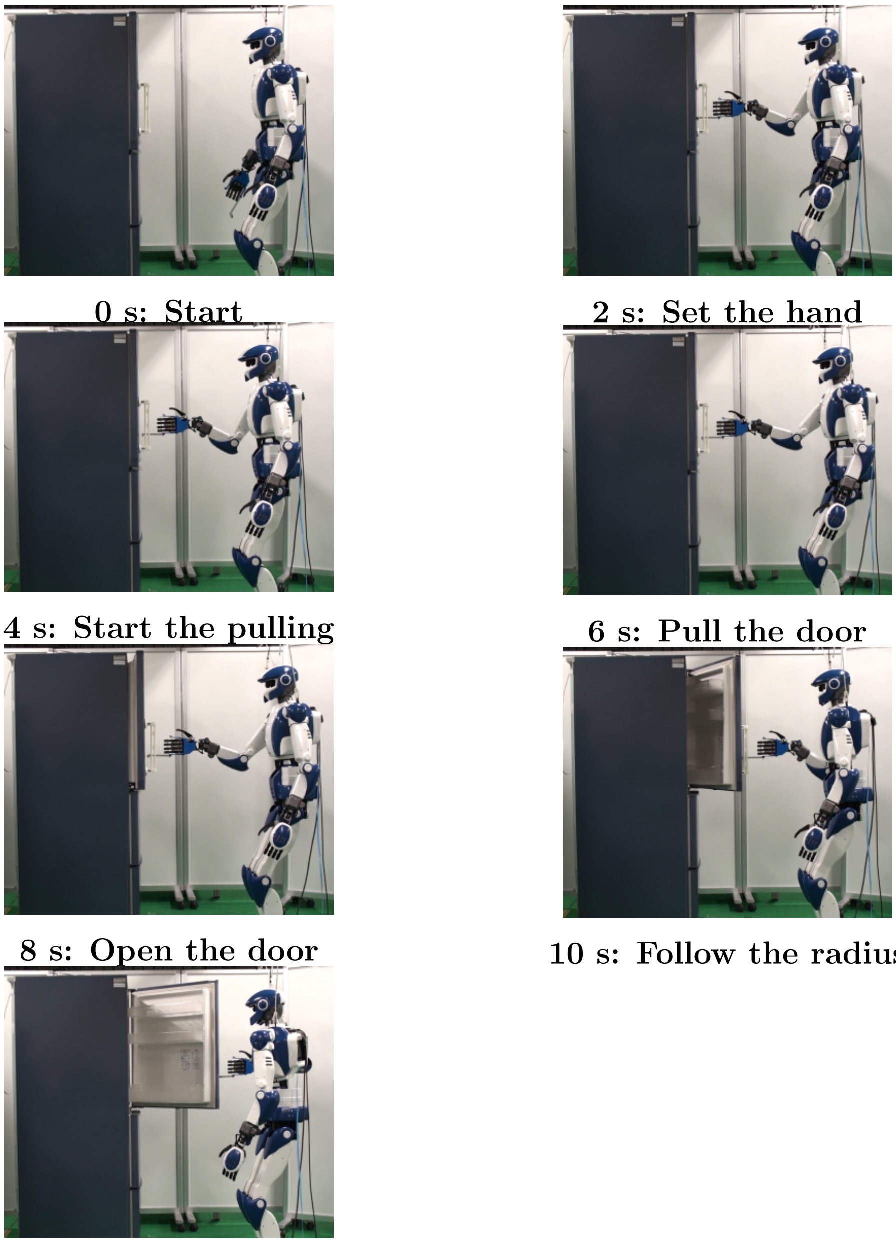


Figure 20. Hook attached to HRP-4 to open the door.

Fig. 21 shows the generated motion, and Fig. 22 shows the ZMP trajectory in the x axis direction. As shown in Fig. 21, the robot succeeded in opening the door. To compare the stability, I also generated the motion only controlling the pose of the hands and feet, and attempted to fix the CoM position during the motion (Conventional method). The ideal ZMP trajectory obtained from the preview control is also shown. At $t = 7$ s, $F_{\max}$ was input. For the proposed method, at about $t = 7$ s, the ZMP moved forward, the door was opened, and the reaction force of the hand was reduced to near zero. In comparison, the motion generated only from kinematics failed to complete the task. This result shows that the door opening task cannot be done without considering the balance of the robot.



0 s: Start **2 s: Set the hand**

4 s: Start the pulling **6 s: Pull the door**

8 s: Open the door **10 s: Follow the radius**

12 s: Finish

Figure 21. Motion to open the door. From $t = 0$ s to $t = 4$ s, move the hand to the handle of the refrigerator. From $t = 4$ s to $t = 8$ s, pull straight 0.03 m using the proposed method. From $t = 8$ s to $t = 12$ s, open the door using inverse kinematics of the arm to follow the door radius trajectory.

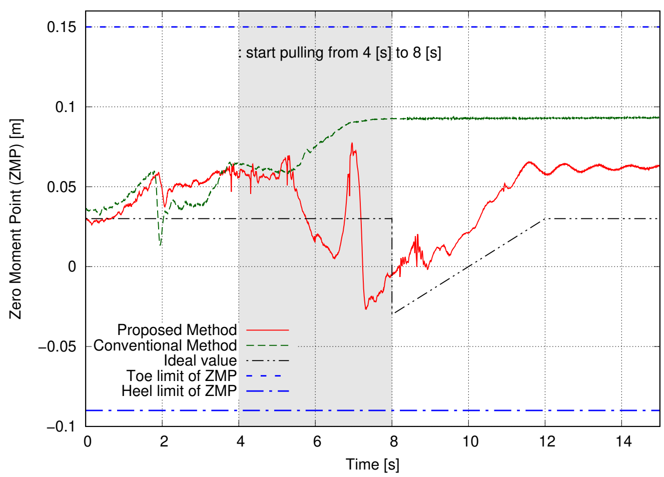


Figure 22. ZMP trajectory during the task of opening the door in the x axis direction

6.3 Detailed motion verification

6.3.1 Pushing objects with different weights

For the validation of the proposed method when the weight of the box is unexpected, I experimented with pushing the box with the mass set to 10 kg, 20 kg, 30 kg, and infinity, *i.e.*, a fixed box. The robot pushed the boxes using the motion fitted for pushing a 20 kg box. In this experiment, I used a simulator and set the following target values:

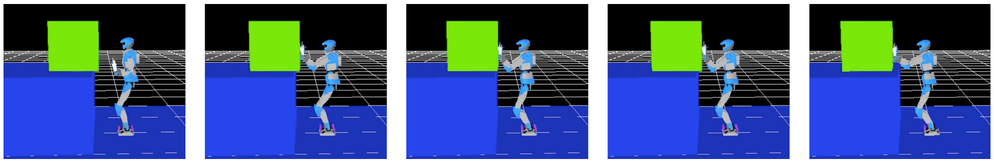
- Coefficient of static friction $\mu = 0.3$.
- Maximum force $F_{\text{max}} = 0.6$ kgf.
- Time until the object moved $t_{\text{max}} = 2.0$ s.
- Target height of the CoM $z_{\text{com}} = 0.80$ m.
- Height of the hands $z_{\text{hand}} = 1.30$ m.

I selected the value of t_{max} from a simulation experiment based on moving the target object quickly and avoid falling.

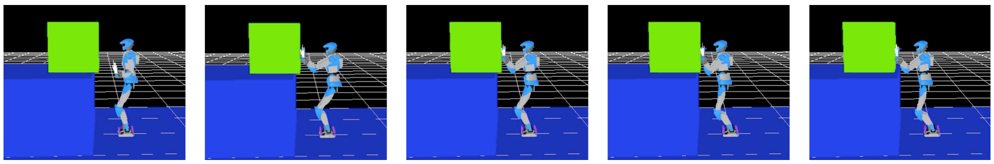
I hypothesize that even when applying the maximum force to a box with a weight greater than 20 kg, the box does not move.

Fig. 23 shows the results. In all cases, first from $t = 0$ s to $t = 3$ s, the robot moved the CoM and hands to the initial position for starting the motion to push the box. In the robot coordinate system, the CoM moved above the center point of both feet. Next, the robot pushes the box. In each case, the final positions of the hands are different. When the box was 10 kg, the robot pushed the box faster than when the box was 20 kg, and it can be seen that the box is pushed slightly backward. This is because the force is larger than the force necessary to move the box. When the box was 20 kg, the HRP-4 was most stable when pushing, and successfully pushed the box without falling down. The robot leaned toward the box and gradually increased the speed until it completed the movement. When pushing the 30 kg box and the fixed box, the distance moved by the hands was shortened. This was due to feedback on the speed of the hands. In both cases, the robot leaned against the box as when pushing a 20 kg box but the toes of the robot floated. Since the box was heavier than expected, the robot was not able to push the box.

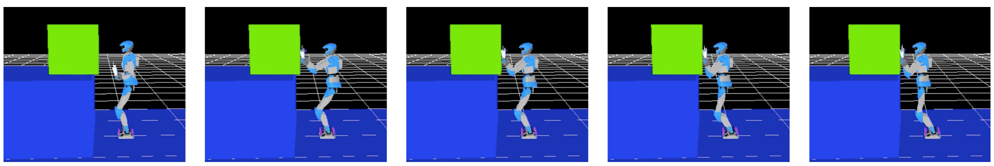
Fig. 24 shows the ZMP trajectory for each motion. I assumed that the ZMP trajectory would be zero during the pushing motion. When pushing a 20 kg box, the ZMP trajectory remained inside of the convex hull of the feet supporting area. This shows that an appropriate motion was generated for the assumed situation. However, when pushing the 10 kg box, after the motion, the ZMP moved forward because the reaction force from the hands was smaller than for the 20 kg case. The ZMP, in this case, moved to the toes. Similarly, for the 30 kg and fixed box cases, the ZMP moved to the heels. In these three cases, the ZMP was located at the boundary of the foot sole. Although the robot did not fall down, it is preferable that the ZMP is not located at the toe or heel for a long time because that increases the risk of loosing the balance.



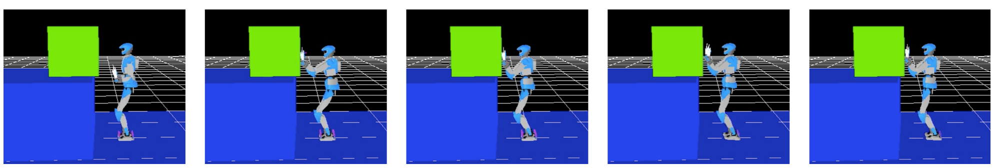
(a) 10 kg (lighter than that the method expects)



(b) 20 kg (the weight that the method assumes)



(c) 30 kg (heavier than that the method expects)



(d) Fixed box

Figure 23. Generated pushing motion. From left to right, $t = 0$ starting position, $t = 3$ s initial position for the pushing motion, $t = 5$ s during pushing motion, $t = 6$ s pushing motion completed and $t = 8$ s after the motion.

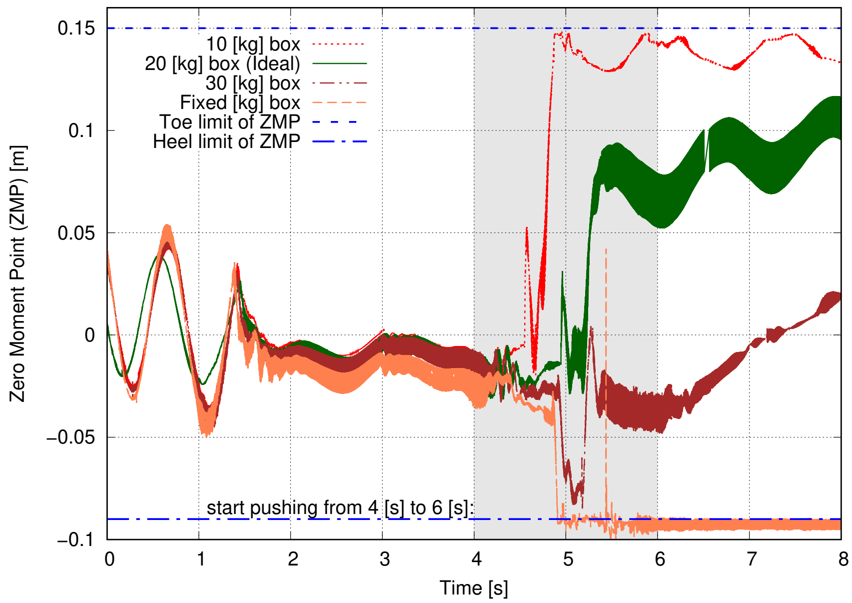


Figure 24. ZMP trajectory during the task of pushing the box with the mass set to 10 kg, 20 kg, 30 kg and infinity in the x axis direction

6.3.2 Avoiding falls due to speed limits

If the weight of the box is different from the expected weight, excessive velocity will be generated in the hands, which may cause the robot to fall over. An example of such a case is shown in Fig. 25. Therefore, I used the final target position of the hands in pushing an object x_{hand}^{ref} to use the hand speed limit V_{limit} m/s according to

$$V_{limit} = \frac{x_{hand}^{ref} - x_{hand}}{t}, \tag{70}$$

where $t = 1$ s and the upper limit is the speed to reach the target position from the current position x_{hand} to the target value during 1 s. The target acceleration of the hand is limited by V_{limit} at the stage of calculating the target speed, which prevents excessive input values from being applied to the robot. Considering the experiments on the actual machine, I decided to push a box of 10 kg within the force that the HRP-4 can exert, and created the corresponding input and checked the operation. The results are shown in Fig. 26. I succeeded in pushing the 10 kg box without any problems. The 5 kg box could be pressed in the same way. 30 kg could not be pushed, but the hand did not experience any unusual acceleration, so it did not fall over.

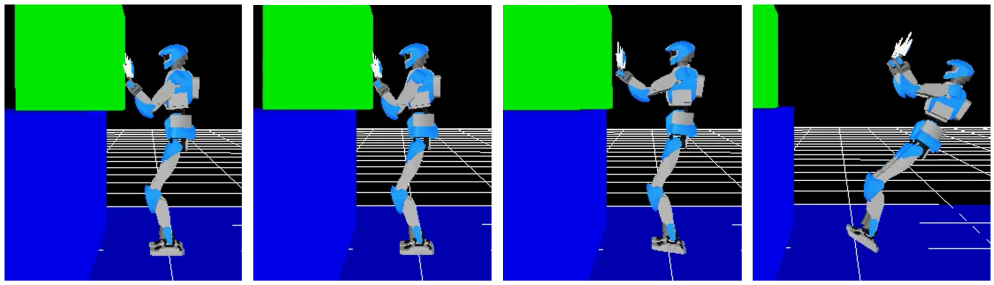
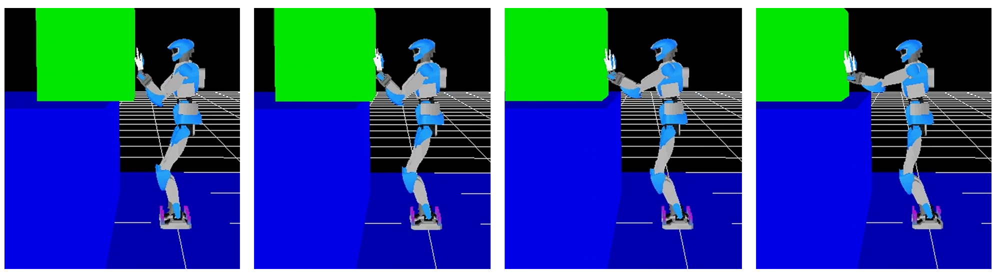
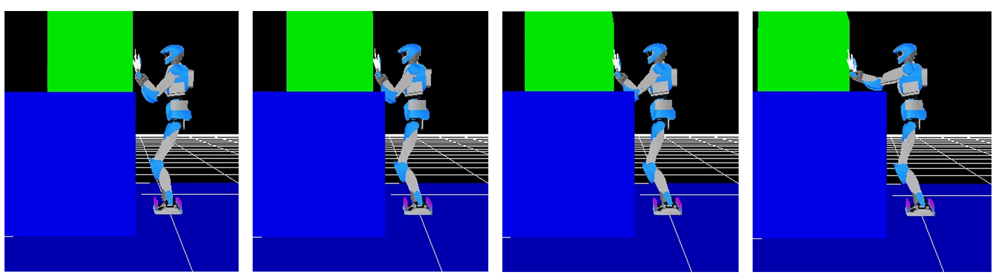


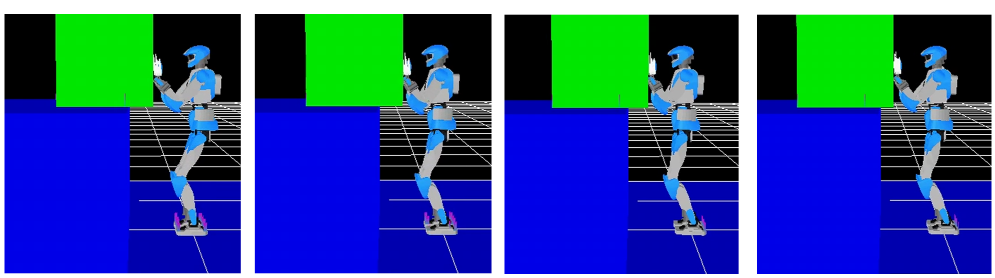
Figure 25. Pushing a box of unexpected weight and falling over. The input value is for pushing a 10kg box, and the actual weight of the box is 30kg.



(a) 5 kg



(b) 10 kg (Weight of box as expected.)



(c) 30 kg

Figure 26. Generated pushing motion using speed limiter

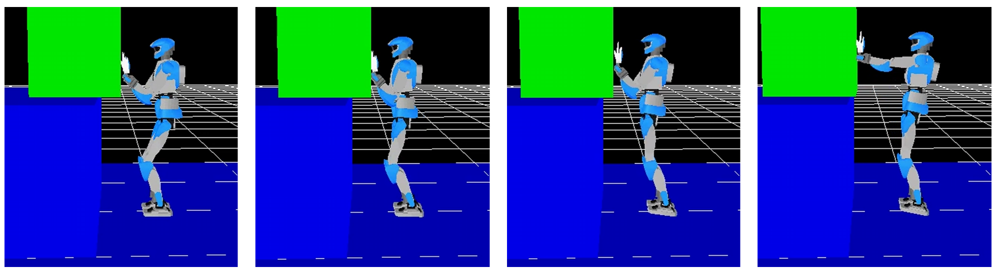
6.3.3 Experiments with heavier objects

I conducted experiments to see if it is possible to push heavier objects using this method. The weights of the boxes used were 30 kg, 40 kg and 50 kg. No speed limit was set to allow for greater power. The inputs for 30 kg, 40 kg, and 50 kg were determined by multiplying the force required to push the 10 kg box by a factor of 3, 4, and 5.

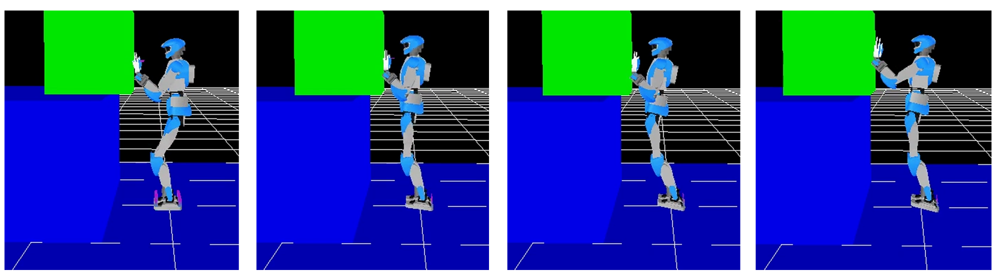
The results are shown in Fig. 27. In all cases, the HRP-4 were not able to push the box during the pushing motion because the toes were lifted. As the weight increased, the arms did not expand or contract, and in the case of 50 kg, the body was pushed forward more than the hands.

One of the possible causes of the immobility of the arm is that the arm is not capable of exerting the necessary force. As the cause of the failure to show the force, the position of the center of gravity is controlled by feed-forward control, while the hand uses acceleration control, which feeds back the velocity.

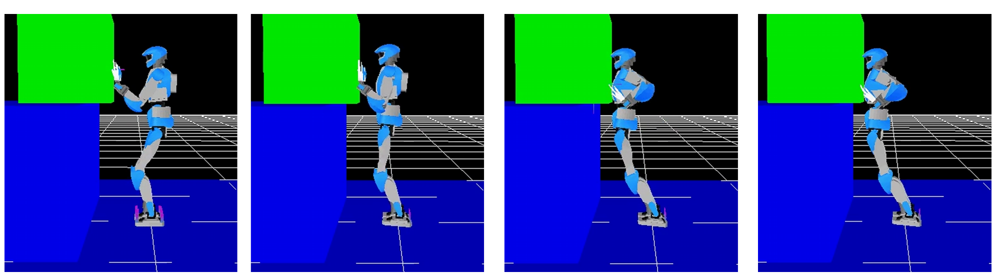
In addition, the center of gravity always moves forward, whereas the hand cannot move forward unless the object is moved. In the experimental results of pushing a 50 kg box, the body contacted the box due to the shift of the CoM, and the hand posture could not be maintained due to the limitation of the joint angle. Thus, even if the target force is exerted by hand, it is necessary to assume an initial position so that the body does not contact the box.



(a) 30 kg



(b) 40 kg



(c) 50 kg

Figure 27. Generated motion to push heavy boxes

6.3.4 Generated force

Using a force sensor, I measured the force generated by the pushing motion. I built a measuring box which consists of an aluminum frame, a force sensor and a weight. Since the payload of the robot was limited, I selected to equip the sensor on the object. Fig. 28 shows the robot and the measuring box. In this experiment, the sum of the masses, including the sensor and the frame, is 10 kg. The force required to move the measuring box is 30 N.

Fig. 29 shows the force sensor data. The target force value is set to increase from 3 s to 5 s. After that, the hand is gradually slow and stop. For 3 s to 5 s, the generated force is smaller than the target value, while from 5 s to 7 s it has exerted more than the target value which is necessary to move the box. In this case, at the beginning of the motion, the reaction force tilted the body of the robot toward the back. In our method, I do not use feedback control of the ZMP, and I confirmed to achieve the target force.

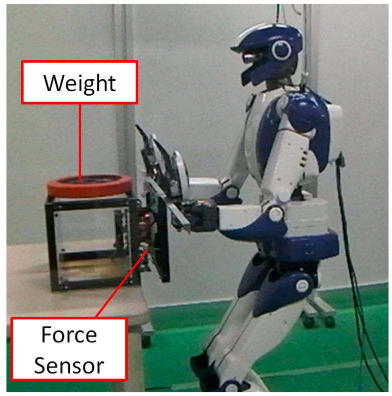


Figure 28. Generated motion for pushing a force sensor using the HRP-4. The mass is 10 kg, including the sensor and the frame.

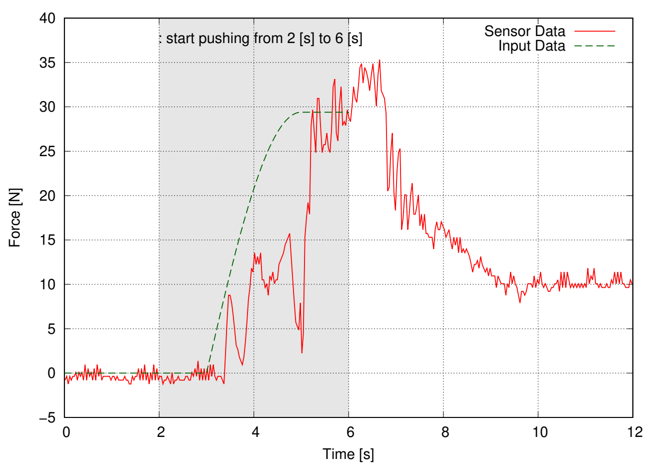


Figure 29. Force trajectory during the pushing motion in the x axis direction

6.4 Discussion

6.4.1 Automatic acquisition of prior knowledge

The proposed method achieves high speed of pushing and pulling motions, but the weight, friction coefficient, and force required to push the object must be known as prior knowledge in order to realize the motion without falling over. As mentioned earlier, the experiment uses pre-measured values. The weight of the target object and the force required to push it are measured using a scale and a spring scale, respectively, and the coefficient of friction is calculated in advance from these values. As for the dynamic simulation, the weight of the box and the coefficient of friction are also set to these measured values. Therefore, one of the future tasks is the automatic acquisition of prior knowledge.

In the proposed method, the force required to push the target object is necessary to calculate the target CoM trajectory, and the weight of the target object and the coefficient of friction are necessary to calculate the target acceleration of the hands. However, if target acceleration of the hands is calculated from the torque control of the joints with the force as the target value, it is not necessary to measure the weight of the object and the coefficient of friction.

Therefore, I consider measuring only the force required to push the target object. Since the proposed method is not designed to push an unknown object, I consider using an existing method that uses a force sensor on the hands to push the object. The flow of operation is as follows.

1. Unknown objects are pushed slowly using force sensors in order to acquire prior knowledge

2. For objects that are known, fast motion can be performed with the proposed method.

I expect that this method will allow us to automatically acquire prior knowledge and adapt the proposed method.

6.4.2 Improved stability during operation

When the kicking motion is performed in section 6.1, the ZMP is changing despite the fact that the momentum is controlled. This is because the floor reaction force is not taken into account. As for the floor reaction force, the original equation does not have a term for the floor reaction force, because it deals with momentum control in the air or in space [18]. Therefore, the control of rotational momentum other than in the vertical direction becomes unstable because it is affected by the floor reaction force during actual operation [9].

Takubo *et al.* achieved the pushing motion by adding the target ZMP to the translational components of the target momentum $\boldsymbol{P}^{ref}$. Similarly, by adding the floor reaction force and ZMP values to the target rotational momentum $\boldsymbol{L}^{ref}$, it is considered that more stable motion can be achieved.

6.4.3 Scope of applicability of this method

In order to adapt the proposed method, it is necessary to satisfy Eq. (59). Since the robot does not walk in the experiment and the target ZMP is set to 0, the following should be satisfied

$$\frac{z_{\mathrm{hand}}}{mg} F_{\mathrm{hand}} = \frac{z_{\mathrm{com}}}{g} \ddot{x}_{\mathrm{com}} - x_{\mathrm{com}}. \tag{71}$$

The robot's mass m and gravity acceleration g are fixed values, the reaction force F_{hand} is the input value, and the CoM position x_{com} is the output value, so the motion is affected by changing the height of the hands and the CoM. To achieve a large force, the force section should be small, and the acceleration section should be large. In other words, lowering the hand height and higher the CoM height allows for large forces to be exerted. However, motion is limited by the following factors

1. Maximum torque of the motor.

2. Motion range of arms and legs.

Regarding the motor torque, the weight of the object that can be moved is limited by the upper limit of torque that a real robot can exert. In the simulator, this limitation can be eliminated, making it possible to push objects that are heavier than actually possible. In addition, the actual HRP-4 is designed to stop when it is overloaded, so I need to adjust its operation to keep it within the limits.

Regarding the motion range, the robot model limits the range in which the hand height and CoM height can be changed. In addition, the legs and arms must not interfere with each other. The problem is that the motion range of robots is narrower than that of humans. Fig. 30 shows the HRP-4's hip joint pitch axis moved to its limit. Humanoid robots are often unable to assume the same posture as humans because their joints have a narrower range of motion and less freedom than those of humans. Fig. 31 shows the HRP-4 posture of sitting down to do pushing motion. Since the ankle pitch axis of the right leg (back leg) has reached the limit of its range, the HRP-4 is unable to move its CoM forward from this posture. Here, the object to be pushed is located in front of the knee. The hand is positioned slightly forward of the knee, but it can only be pushed for a fairly short distance, indicating that it is not practical. As described above, it is necessary to consider the physical characteristics of the robot to determine the movement posture. In the ankle pitch axis, there is a possibility that it can be improved by standing on the toes, but care must be taken to ensure stability and load on the ankle joint.

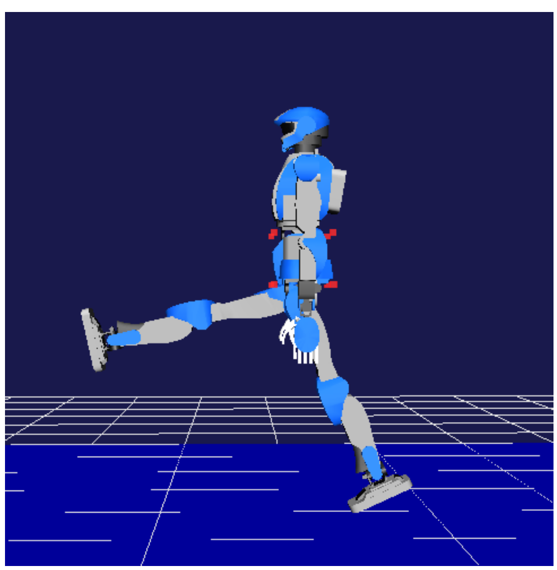


Figure 30. Range of hip pitch axis movement of the HRP-4.

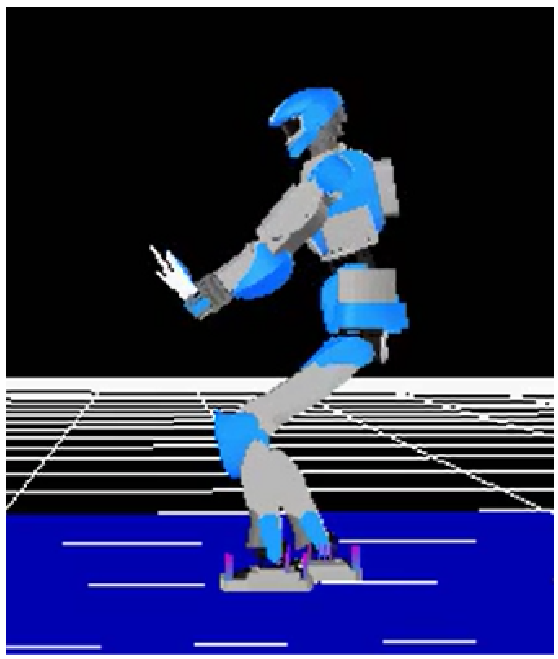


Figure 31. Pushing motion with lowered waist: The HRP-4 cannot move forward because the pitch axis of the right ankle is at the joint limit.

7. Conclusions

In this dissertation, I proposed a method for generating a whole-body motion for physical interaction and verified it experimentally using a simulator and the real robot HRP-4. By assuming that the expected force needed for the interaction is known, the proposed method increases the speed of the motion. The force is calculated from physical properties of the object, such as its mass and coefficient of friction.

First, I accelerated the calculation of the whole-body motion using Resolved Momentum Control (RMC) and analytical inverse kinematics. Then, I proposed a method to calculate the target trajectory of the CoM using a preview control to generate the target force.

Next, I conducted both pushing and pulling experiments using the humanoid robot HRP-4 to verify the effectiveness of the proposed method. The proposed method succeeded in pushing a 20 kg box in 6 s and opening a refrigerator door in 12 s. Using the proposed method, the time required to calculate the whole body motion is about 0.3 ms, which is shorter than the control cycle of the HRP-4 (5 ms) and similar humanoid robots.

Finally, I further verified the effectiveness of the proposed method for pushing motions in simulation. Specifically, I tested the motion to push boxes with different weights under the inaccurate input value that the boxes weighed 20 kg. As a result, the robot was the most stable when pushing a 20 kg box, and I were able to confirm that the proposed method generates a suitable motion for the assumed situation. Furthermore, I also confirmed the behavior when the weight of the box was different from the assumed physical properties. Furthermore, using a force sensor and the HRP-4, I verified that the required force was generated.

For humanoid robots to support our daily life activities, the robots must be able to carry out physical interactions as humans do. For example, in daily life, opening and closing doors, refrigerators, and drawers, or moving boxes are usual motions. This research helps to move us closer to the goal of having robots that can support our daily life. In our future work, I plan to combine these results and apply our proposed method to the humanoid pushing while walking. For example, I might have our humanoid robot push a cart.

Acknowledgements

I would like to express my special appreciation and thanks to my advisor Professor Tsukasa Ogasawara. He provided me with valuable research opportunities and gave me many kind comments on student life and research topics. I also received many comments on the content of my dissertation works, which helped me to deepen my understanding of the research. I would also like to thank my committee member Professor Kenji Sugimoto for serving as my committee member even at hardship. I also want to thank you for letting my defense be an enjoyable moment, and for your brilliant comments and suggestions, thanks to you. I really appreciate to Professor Jun Takamatsu, he kindly gave me a lot of the opportunity for discussion and helps. For my master's thesis and doctoral dissertation, I received precise comments on how to think about research and write papers, which helped me to clarify the direction of my research. Particular thank you go to Assistant Professor Gustavo Alfonso Garcia Ricardez. He has advised me many times in my research. I was fortunate to have him proofread my English many times. I am grateful to Assistant professor Satoki Tsuichihara for his great support in writing this dissertation. His advice was very helpful to me in my daily life as a student and in my research. Also, I would like to thank my family for their support and warm encouragement.

References

[1] J. Lee, A. Ajoudani, E. M. Hoffman, A. Rocchi, A. Settimi, M. Ferrati, A. Bicchi, N. G. Tsagarakis, and D. G. Caldwell, "Upper-body impedance control with variable stiffness for a door opening task," in *2014 IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 713–719, 2014.

[2] M. Prats, S. Wieland, T. Asfour, A. P. del Pobil, and R. Dillmann, "Compliant interaction in household environments by the armar-iii humanoid robot," in *2008 IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 475–480, 2008.

[3] T. B. Sheridan, "Three models of preview control," *IEEE Transactions on Human Factors in Electronics*, no. 2, pp. 91–102, 1966.

[4] L. Righetti, J. Buchli, M. Mistry, M. Kalakrishnan, and S. Schaal, "Optimal distribution of contact forces with inverse-dynamics control," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 280–298, 2013.

[5] C. Ott, B. Henze, and D. Lee, "Kinesthetic teaching of humanoid motion based on whole-body compliance control with interaction-aware balancing," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4615–4621, 2013.

[6] B. Henze, C. Ott, and M. A. Roa, "Posture and balance control for humanoid robots in multi-contact scenarios based on model predictive control," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3253–3258, 2014.

[7] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1168–1175, 2014.

[8] K. Harada, S. Kajita, K. Kaneko, and H. Hirukawa, "Pushing manipulation by humanoid considering two-kinds of zmps," in *2003 IEEE Interna-*

tional Conference on Robotics and Automation (ICRA), vol. 2, pp. 1627–1632, 2003.

[9] T. Takubo, K. Inoue, and T. Arai, "Pushing an object considering the hand reflect forces by humanoid robot in dynamic walking," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1706–1711, 2005.

[10] 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 *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5682–5689, 2015.

[11] T. Koolen, T. De Boer, J. Rebula, A. Goswami, and J. Pratt, "Capturability-based analysis and control of legged locomotion, part 1: Theory and application to three simple gait models," *The international journal of robotics research*, vol. 31, no. 9, pp. 1094–1113, 2012.

[12] H. Arisumi, J.-R. Chardonnet, and K. Yokoi, "Whole-body motion of a humanoid robot for passing through a door-opening a door by impulsive force," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 428–434, 2009.

[13] N. Banerjee, X. Long, R. Du, F. Polido, S. Feng, C. G. Atkeson, M. Gennert, and T. Padir, "Human-supervised control of the atlas humanoid robot for traversing doors," in *2015 IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 722–729, 2015.

[14] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *2003 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, pp. 1620–1626, 2003.

[15] P.-B. Wieber, "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," in *2006 6th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 137–142, 2006.

[16] A. Ibanez, P. Bidaud, and V. Padois, "Unified preview control for humanoid postural stability and upper-limb interaction adaptation," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1801–1808, 2012.

[17] S. Kajita, M. Morisawa, K. Harada, K. Kaneko, F. Kanehiro, K. Fujiwara, and H. Hirukawa, "Biped walking pattern generator allowing auxiliary zmp control," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2993–2999, 2006.

[18] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Resolved momentum control: Humanoid motion planning based on the linear and angular momentum," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 2, pp. 1644–1650, 2003.

[19] M. Vukobratović and J. Stepanenko, "On the stability of anthropomorphic systems," *Mathematical biosciences*, vol. 15, no. 1-2, pp. 1–37, 1972.

[20] K. Kaneko, F. Kanehiro, M. Morisawa, K. Akachi, G. Miyamori, A. Hayashi, and N. Kanehira, "Humanoid robot hrp-4-humanoid robotics platform with lightweight and slim body," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4400–4407, 2011.

[21] K. Kreutz-Delgado, M. Long, and H. Seraji, "Kinematic analysis of 7-dof manipulators," *The International journal of robotics research*, vol. 11, no. 5, pp. 469–481, 1992.

[22] M. Shimizu, H. Kakuya, W.-K. Yoon, K. Kitagaki, and K. Kosuge, "Analytical inverse kinematic computation for 7-dof redundant manipulators with joint limits and its application to redundancy resolution," *IEEE Transactions on robotics*, vol. 24, no. 5, pp. 1131–1142, 2008.

[23] H. Asada and J.-J. Slotine, *Robot analysis and control.* John Wiley & Sons, 1986.

[24] N. Ando, T. Suehiro, K. Kitagaki, T. Kotoku, and W.-K. Yoon, "Rt-middleware: distributed component middleware for rt (robot technology)," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3933–3938, 2005.