PALADYN Journal of Behavioral Robotics

VERSITA

Research Article · DOI: 10.2478/pjbr-2013-0002 · JBR · 4(1) · 2013 · 23-33

Efficient Planning of Humanoid Motions by Modifying Constraints

ChangHyun Sung*, Takahiro Kagawa, Yoji Uno

Department of Mechanical Science and Engineering, Graduate School of Engineering, Nagoya University
Furo-cho Chikusa-ku, Nagoya 464-8603, Japan

Received 06-03-2013

Accepted 05-07-2013

Abstract

In this paper, we propose an effective planning method for whole-body motions of humanoid robots under various conditions for achieving the task. In motion planning, various constraints such as range of motion have to be considered. Specifically, it is important to maintain balance in whole-body motion. In order to be useful in an unpredictable environment, rapid planning is an essential problem. In this research, via-point representation is used for assigning sufficient conditions to deal with various constraints in the movement. The position, posture and velocity of the robot are constrained as a state of a via-point. In our algorithm, the feasible motions are planned by modifying via-points. Furthermore, we formulate the motion planning problem as a simple iterative method with a Linear Programming (LP) problem for efficiency of the motion planning. We have applied the method to generate the kicking motion of a HOAP-3 humanoid robot. We confirmed that the robot can successfully score a goal with various courses corresponding to changing conditions of the location of an obstacle. The computation time was less than two seconds. These results indicate that the proposed algorithm can achieve efficient motion planning.

Keywords

humanoid robot · motion planning · modifying via-point

1. Introduction

Humanoid robots with many degrees of freedom have the potential of general-purpose properties and are expected to be used in human daily life. However, many combinations of multi-joints movement, which satisfy various constraints, such as the maintenance of balance and the range of motion, must be considered in the planning of whole-body motion for humanoid robots. Furthermore, it is necessary to consider the various conditions that must be met in achieving the task according to the surrounding environment.

In the past, the motion of humanoid robots was planned by using simplified models. By reducing the many degrees of freedom and focusing the movement of center of mass, a planning method can rapidly yield the motion of the robot. For example, an inverted pendulum model [1, 2] and a cart-on-a-table model [3] were often used to generate the locomotion of a humanoid robot. Specifically, the use of a simplified model with footstep planning [32, 33] makes possible on-line control of a humanoid. However, the motion planning method that employs these models cannot deal with the constraint conditions such as the range of joint angle motion and collision avoidance with each joint. In motion planning, there are various constraint conditions that have to be satisfied over whole motion duration.

In recent years, the whole-body motion planning for humanoid robots has been solved as a constrained optimization problem [4]. Semi Infinite Programming (SIP) is an optimization problem with a finite number of variables to optimize and a set of continuous constraint functions that is equivalent to an infinite number of discrete constraints [5]. With the parameterization of joint angle trajectories, the motion planning problem can be transformed into SIP [6]. SIP is typically used to generate kicking motion [7–9], throwing motion [10] and multi-contact motion [11] which is the motion combining subtasks. Furthermore, parallel tasks [12], balance motion [13] and impact motion [14] are generated according to a constrained optimization problem. We can obtain motions that are satisfied with constraints by considering motion planning as the optimization problem.

The appropriate motions of the robot have to be planned in a short time to be effective in an unpredictable human environment. Usually, the constraint conditions over whole motion duration are discretized in the optimization problems including SIP [5, 15]. The satisfaction of constraint conditions in the motion planning can be guaranteed from the short discretized intervals. However, the shorter intervals used, the more the constraint conditions have to be considered in optimization. Furthermore, the constraint conditions of balance or collision avoidance are nonlinear. Therefore, a large amount of the computation cost is required to solve the optimization of SIP problems. Dealing with a small number of constraint conditions is one way of reducing the computation time. In addition, the period to plan the motions of the robots would be decreased by linear approximate calculation of nonlinear constraint conditions.

For achieving the task, the robot should have an ability to take actions against the various conditions of changing environment. Generally, the appropriate constraints in the movement of the robots lead to successful accomplishment of the task. Via-point representation is mentioned as one of the ways for specifying the sufficient condition of these constraints. Assigning some via-points, dynamical human arm movements can be formulated by optimization [16, 17]. Taking a specific position in human movements as via-points, the redundant robot manipulators performed a skillful task [18].

In our previous work, we addressed the planning of suitable motions using the via-point representation [19]. Each joint angle trajectory of the robots was directly determined on the basis of minimum jerk with via-point constraints. Via-point parameters consist of the joint angle and angular velocity at the important points of the motion to accomplish the

*E-mail: {s_changhyun, kagawa, uno}@nuem.nagoya-u.ac.jp

task. The various conditions for achieving the task were determined by a state of via-point. In the motion planning, the position, posture and velocity of the robot were constrained as a state of via-point. We proposed a whole-body motion planning method (referred to as the previous method in this paper) by optimizing all parameters of the via-point as a SIP problem. Therefore, the computational cost was too expensive.

In this research, we propose a rapid motion-planning method for humanoid robots based on the via-point representation. Our policy of motion planning is to modify the via-point parameters from one of a planned motion to satisfy new conditions according to the surrounding environment. We formulate the motion planning problem based on a simple iterative method with Linear Programming (LP) problem [20]. By formulating the motion planning problem as LP, the proposed method can plan whole-body motions that satisfy a number of constraints efficiently. We applied the proposed method to generate the kicking motion of a HOAP-3 humanoid robot. The robot needs to maintain its balance on one support leg, and kicks the ball accurately while taking into account the change in surrounding conditions when controlling the swing leg. Furthermore, lower computation time is desirable to be effective in the real soccer game. We demonstrate the effectiveness of our motion planning algorithm in experiments in which a HOAP-3 humanoid robot performs the kicking motion with consideration of the location of an obstacle. Consequently, the computation time could be significantly reduced using our proposed algorithm.

This paper is organized as follows. Section 2 describes the motion planning problem by specifying the via-point. Section 3 introduces our algorithm of motion planning by modifying the via-point for the humanoid robot. Section 4 illustrates an application of the proposed method and experimental results in generating the kicking motion of a HOAP-3 humanoid robot. Section 5 presents discussion about modification of constraint conditions and compares the computation time with the previous method. We conclude the paper by describing the advantages and the limitations of our motion planning method, and by emphasizing possible developments of our method.

2. Motion Planning by Specifying Via-points

In this section, we describe the motion planning problem based on a via-point representation. We assume that there are some important instants for accomplishing the task in the motion planning. A state of important instant, such as position, posture and velocity of the end effector, is defined as a state of the via-point. Passing through the via-point in its motion, the robot can successfully complete the task. Each joint angle and angular velocity of the robot at the via-point is constrained in the motion planning process. The specification of the via-point is an essential problem, and in general whole-body motion, specifying appropriate via-points is a fairly difficult problem. Therefore we deal with the motion planning problem in which appropriate via-points to achieve the task can be obviously estimated. Specifically, we assume that the timing of passing through the via-point is appropriately given. In the motion planning problem, we calculate the appropriate joint angle and angular velocity parameters at the via-point.

2.1. Representation of Joint Angle Trajectory with Via-points

Minimum jerk is a typical criterion to represent the smoothness of a trajectory [21, 22]. In this research, the joint angle trajectories of the robot are generated by optimization for the jerk cost function. For simplicity, we describe the joint angle trajectory with one via-point. If necessary, a joint angle trajectory with two or more via-points could easily be formulated. Let t_{via} be the time that the joint angle trajectory passes the via-point. We assume in this research that the time t_{via} passing through the via-point is given beforehand so that the task succeeds. The minimum jerk trajectory with via-point constraints of angle $\theta(t_{via})$ and angular velocity $\dot{\theta}(t_{via})$ can be expressed using Lagrange multipliers π_1 and π_2 [16].

$$\theta(t) = \begin{cases} a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \quad (t_s \le t \le t_{via}) \\ a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \quad (t_{via} \le t \le t_f) \\ \quad + \frac{1}{24}(t - t_{via})^4 \pi_1 + \frac{1}{120}(t - t_{via})^5 \pi_2 \end{cases}, \tag{1}$$

where t_s and t_f are the start and end time of the movement.
Equation (1) has eight parameters, which are determined by the six conditions of each joint angle, angular velocity ($= 0$) and angular acceleration ($= 0$) at the initial and final states, and two conditions of joint angle $\theta(t_{via}) = \theta_{via}$ and angular velocity $\dot{\theta}(t_{via}) = \dot{\theta}_{via}$ corresponding to the via-point. Assuming that the time at the via-point and joint angles of initial and final states are given, it is necessary to determine the two conditions of the via-point to represent the joint angle trajectory. In other words, we can express various trajectories that satisfy the boundary by changing only the angle and angular velocity of the via-point. Let m be the number of controlled joints of the robot. The $2m$ parameters of the via-point must be determined to express the entire movement of the robot. These parameters can be expressed by,

$$\begin{aligned} \boldsymbol{\theta}_{via} &= \{\theta_1, \cdots, \theta_m\} \\ \dot{\boldsymbol{\theta}}_{via} &= \{\dot{\theta}_1, \cdots, \dot{\theta}_m\}, \end{aligned} \tag{2}$$

where θ_i and $\dot{\theta}_i$ $(i = 1, \cdots, m)$ are the joint angle and angular velocity of the i-th joint at the via-point. Here, $\boldsymbol{\theta}_{via}$ and $\dot{\boldsymbol{\theta}}_{via}$ are combined and represented by $\boldsymbol{X}$:

$$\boldsymbol{X} = \{\boldsymbol{\theta}_{via}, \dot{\boldsymbol{\theta}}_{via}\}. \tag{3}$$

Because the motion of the robot can be determined by optimization of the parameter set $\boldsymbol{X}$, the joint angle trajectory of the robot can be expressed by a function of the parameter set $\boldsymbol{X}$ as follows.

$$\boldsymbol{\theta}(\boldsymbol{X}, t), \quad \forall t \in [t_s, t_f]. \tag{4}$$

2.2. Constraint Conditions

The condition for accomplishment of the task is the most important in the motion planning. We specify the target information for achieving the task as "position $\boldsymbol{P}$, posture $\boldsymbol{R}$ and velocity $\boldsymbol{V}$ of the end effector at the time t_{via}". We assume that the information that depends on the surrounding environment is appropriately given. We can deal with these conditions using equality constraints. These conditions can be expressed by the via-point parameters $\boldsymbol{X}$, because the trajectories of

the joint angle and angular velocity are functions of the parameters $\boldsymbol{X}$. Therefore, we can define the conditions for achieving the task as

$$\boldsymbol{p}(\boldsymbol{X}) = \boldsymbol{P}_d, \quad \boldsymbol{r}(\boldsymbol{X}) = \boldsymbol{R}_d, \quad \boldsymbol{v}(\boldsymbol{X}) = \boldsymbol{V}_d, \tag{5}$$

where $\boldsymbol{P}_d$, $\boldsymbol{R}_d$ and $\boldsymbol{V}_d$ are the desired target information for achieving the task. Equation (5) can be formulated by forward kinematics equations [25].
There are a number of constraints that have to be satisfied throughout movement duration. We consider the range of the motion, collision avoidance and so on in the motion planning for humanoid robots. These conditions can be represented by inequality constraints with motion planning parameters $\boldsymbol{X}$.

$$\underline{\boldsymbol{\Theta}} \leq \boldsymbol{\theta}(\boldsymbol{X}, t) \leq \overline{\boldsymbol{\Theta}}, \quad \underline{\boldsymbol{P}} \leq \boldsymbol{p}(\boldsymbol{X}, t) \leq \overline{\boldsymbol{P}}, \tag{6}$$

where $\overline{\boldsymbol{\Theta}}$ and $\underline{\boldsymbol{\Theta}}$ are upper and lower limits of the joint angle. $\overline{\boldsymbol{P}}$ and $\underline{\boldsymbol{P}}$ are upper and lower limits for collision avoidance.
Furthermore, it is also important to maintain balance in whole-body motion planning. Usually, the balance of a humanoid robot can be evaluated using the zero moment point (ZMP). The ZMP is the point at which the moment of the ground reaction force is equal to zero in the supporting area [23]. If the ZMP of the planned motion is located within the base of the support with sufficient margins, the humanoid robot can maintain its balance. These conditions are represented based on the via-point representation as follows.

$$\underline{\boldsymbol{Z}} \leq \boldsymbol{z}(\boldsymbol{X}, t) \leq \overline{\boldsymbol{Z}}, \tag{7}$$

where $\overline{\boldsymbol{Z}}$ and $\underline{\boldsymbol{Z}}$ are upper and lower limits of the ZMP.

2.3. Motion Planning Problem

The motion planning problem is usually solved by representing the set of joint angle trajectories. We can determine the joint angle trajectories by calculating the parameter set $\boldsymbol{X}$. In our previous work [19], this problem is solved as the Semi Infinite Programming (SIP) problem by setting some criteria function $C(\boldsymbol{X})$.

$$\begin{aligned} &\arg\min_{\boldsymbol{X}} \quad C(\boldsymbol{X}) \\ &\text{s.t.} \quad\quad \boldsymbol{eq}(\boldsymbol{X}) = \boldsymbol{0} \\ &\text{and} \quad\quad \boldsymbol{ieq}(\boldsymbol{X}, t) \leq \boldsymbol{0}, \quad \forall t \in [t_s, t_f], \end{aligned} \tag{8}$$

where t_s and t_f are the start and end times of movement, respectively. The equality constraint function ($\boldsymbol{eq}(\boldsymbol{X}) = \boldsymbol{0}$) involves the conditions for achieving the task as Equation (5). The inequality constraint function ($\boldsymbol{ieq}(\boldsymbol{X}, t) \leq \boldsymbol{0}$) deals with the Equations (6) and (7).

3. Efficient Planning by Modifying Via-points

In this section, we describe a proposed algorithm to rapidly solve the motion planning problem for humanoid robots. Considering the efficiency of computation time, it is important to satisfy the constraints for achieving the task rather than to minimize the cost function. Generally, if the parameters satisfy the constraint conditions, the motions are regarded to be feasible [34–36]. From this point of view, the solution satisfies the constraints of task achievement in our method, although the solution obtained is not a proper optimal solution under a criterion, Usually, the criterion for motion planning takes into account the features of the robot. Since the trajectory is already formulated according to the minimum jerk criterion in the via-point representation of joint angle trajectory, the smoothness of the trajectory is guaranteed for any parameter values.
In our research, we describe a fast method of planning feasible motions for humanoid robots. Feasible motions according to the surrounding environment are planned by modifying the via-point parameters of a base motion. We define the modified parameters as $\boldsymbol{X}'$ for a given state of via-point $\boldsymbol{P}'$, $\boldsymbol{R}'$ and $\boldsymbol{V}'$.

$$\boldsymbol{X}' = \{\boldsymbol{\theta}'_{via}, \dot{\boldsymbol{\theta}}'_{via}\}, \tag{9}$$

where $\boldsymbol{\theta}'_{via}$ and $\dot{\boldsymbol{\theta}}'_{via}$ are the modified parameters of joint angle and angular velocity at the via-point. In our method, we do not calculate the motion planning parameters $\boldsymbol{X}'$ at one time.
Generally, the computation time is significantly increased with the number of optimization variables. In our method, we expect that computation time would be decreased by separating the complex nonlinear optimization problem into two sequential problems. Each process for the problems corresponds to solving inverse kinematics equations and Linear Programming (LP) problem, respectively. In each process, the number of variables becomes fewer than the previous method. In addition, by solving the inverse kinematics, the SIP problem can be transformed into a LP problem which is effective in decreasing the computation time. The specific calculation is described as follows.

Step A.

Calculate the joint angle parameters $\boldsymbol{\theta}'_{via}$ of the via-point

Step B.

Optimize the joint angular velocity parameters $\dot{\boldsymbol{\theta}}'_{via}$ of the via-point

In our motion planning algorithm, the new parameters are sequentially calculated by the two steps. In each step, we consider the various constraints including the conditions for achieving the task. First, we calculate the joint angle parameter $\boldsymbol{\theta}'_{via}$ of the via-point that satisfies the new conditions $\boldsymbol{P}'$ and $\boldsymbol{R}'$, solving the inverse kinematics equations (**Step A**). Next, we optimize the joint angular velocity parameters $\dot{\boldsymbol{\theta}}'_{via}$ of the via-point satisfying the new conditions $\boldsymbol{V}'$ as a simple iterative method with Linear Programming problem (**Step B**). Obviously, the other constraints that have to be satisfied over the whole motion duration are considered. After calculation of **Step A**, the motion can be represented by joint angular velocity parameters $\dot{\boldsymbol{\theta}}'_{via}$ at the via-point. Linearization of the nonlinear constraints with respect to the $\dot{\boldsymbol{\theta}}'_{via}$ allows to application of a Linear Programming problem in **Step B**.

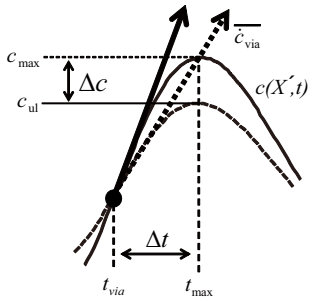


Figure 1. Conceptual sketch of calculating upper limit of velocity at the via-point ($\overline{\dot{c}_{via}}$). $\overline{\dot{c}_{via}}$ can be calculated so that the maximum value of $c(X', t)$ is equal to the upper boundary of inequality constraint (c_{ul}).

3.1. Linearization of Constraint Conditions

Before describing **Step A** and **Step B**, we explain our linearization method to represent the linear approximation function of the nonlinear inequality constraints to computation time. Our idea is that nonlinear constraints over whole motion duration are transformed into linear constraints at the via-point. The maximum and minimum values of constraint over whole motion duration are considered by the upper and lower bounds of the constraint at the via-point. We employ the certain constraint function $c(X', t)$ that has to be satisfied with the upper and lower limits c_{uL} and c_{lL} during whole motion duration.

$$c_{lL} \leq c(X', t) \leq c_{uL}, \quad \forall t \in [t_s, t_f]. \tag{10}$$

The joint angle trajectories are the function of two kinds of parameters; the joint angle parameters θ'_{via} and joint angular velocity parameters $\dot{\theta}'_{via}$ at the via-point. Assuming that θ'_{via} have already been determined in **Step A**, the shape of the nonlinear constraint $c(X', t)$ depend on $\dot{\theta}'_{via}$.
Figure 1 is a conceptual sketch for setting the appropriate condition of the upper limit $\overline{\dot{c}_{via}}$ at the via-point. As shown Figure 1, the large velocity of $c(X', t)$ at the via-point results in a higher maximum value which violates a constraint condition. If we could set the appropriate upper limit $\overline{\dot{c}_{via}}$, we can obtain the motion that is satisfied with the given constraint over whole motion duration. In our method, the constraint condition is expressed using the conditions related to the velocity at the via-point as follows.

$$\underline{\dot{c}_{via}} \leq J_c(\theta'_{via})\, \dot{\theta}'_{via} \leq \overline{\dot{c}_{via}}, \tag{11}$$

where $\overline{\dot{c}_{via}}$ and $\underline{\dot{c}_{via}}$ are upper and lower limits of the via-point, which satisfy the constraint condition of Equation (11). The $J_c(\theta'_{via})$ is a Jacobian related to the velocity of $c(X, t)$ around the via-point.
In addition, our method includes the calculation of the desired upper limit $\overline{\dot{c}_{via}}$ and lower limit $\underline{\dot{c}_{via}}$. The calculation of $\overline{\dot{c}_{via}}$ and $\underline{\dot{c}_{via}}$ is based on the Newton method [24]. Δc and Δt of Figure 1 are defined as

$$\begin{aligned} \Delta c &= |c_{max} - c_{ul}| \\ \Delta t &= |t_{max} - t_{via}| \ , \end{aligned} \tag{12}$$

where c_{max} is the maximum value of the constraint function $c(X, t)$. t_{max} is the time that the function $c(X, t)$ reaches the maximum value. The appropriate upper limit $\overline{\dot{c}_{via}}$ can be calculated with the following simple algorithm.

P1.
Set the starting value of $\overline{\dot{c}_{via}}$.

P2.
Calculate current Δc and Δt.

P3.
If Δc is zero, halt computation.

P4.
Else, update $\overline{\dot{c}_{via}} = \overline{\dot{c}_{via}} \pm \lambda \Delta \overline{\dot{c}_{via}}$, and go to **P2**.

If Δc is higher than zero in **P3**, it means that the constraint is not satisfied. By linear approximation as shown in Figure 1, the calculation for updating upper limit $\overline{\dot{c}_{via}}$ in **P4** can be derived as follows.

$$\Delta \overline{\dot{c}_{via}} \equiv \frac{\Delta c}{\Delta t}. \tag{13}$$

The lower limit $\underline{\dot{c}_{via}}$ can also be calculated in the same way. In the proposed method, the nonlinear constraints over whole motion duration are considered as the linear constraints at the via-point. From this transformation, the number of constraints to be considered is greatly reduced because a time discretization technique is not required. This method enables not only reduction in the number of inequality constraints but also the application of a Linear Programming problem.

3.2. Step A: Calculate the Joint Angle Parameter of the Via-point

In **Step A**, we calculate the joint angle parameters θ'_{via} by solving the inverse kinematics equations. There are a number of solutions θ'_{via} that are satisfied with given target information. In our research for the solution of inverse kinematics, we apply an iterative method consisting of the forward kinematics calculation and generalized inverse of the Jacobian [25]. This method is often used to determine the position and posture of redundant manipulators [26]. We can obtain the minimum value of the squared norm by performing the following algorithm.

Step A1.
Prepare the new target conditions P' and R' for achieving the task.

Step A2.
Calculate current conditions about P and R.

Step A3.
Calculate differences (ΔP, ΔR), alternatively referred to as errors, between the new target

conditions and current ones.

$$err(\Delta \boldsymbol{P}, \Delta \boldsymbol{R}) = \Delta \boldsymbol{P}^2 + \Delta \boldsymbol{R}^2, \tag{14}$$

Step A4.

If $err(\Delta \boldsymbol{P}, \Delta \boldsymbol{R})$ is smaller, halt the computation

Step A5.

Else, update as $\boldsymbol{\theta}'_{via} = \boldsymbol{\theta}'_{via} + \mu \Delta \boldsymbol{\theta}'_{via}$, and go to **Step A3**

$$\Delta \boldsymbol{\theta}'_{via} = \boldsymbol{J}(\boldsymbol{\theta}'_{via})^{\dagger} \left(\begin{bmatrix} \Delta \boldsymbol{P} \\ \Delta \boldsymbol{R} \end{bmatrix} \right) + (\boldsymbol{E} - \boldsymbol{J}^{\dagger}(\boldsymbol{\theta}'_{via}) \boldsymbol{J}(\boldsymbol{\theta}'_{via})) \boldsymbol{k}, \tag{15}$$

In Equation (14) in **Step A4**, $\Delta \boldsymbol{P} = \boldsymbol{P}' - \boldsymbol{P}$ is the error of the position of the end effector and $\Delta \boldsymbol{R}$ is the error of the posture of the end effector [25]. In Equation (15) in **Step A6**, $\boldsymbol{J}^{\dagger}(\boldsymbol{\theta}'_{via})$ is a pseudo inverse matrix of Jacobian $\boldsymbol{J}(\boldsymbol{\theta}'_{via})$ for end effector's position $\boldsymbol{P}$ and posture $\boldsymbol{R}$, and $\boldsymbol{k}$ is an arbitrary vector of null space. Furthermore, $\boldsymbol{E}$ is a unit matrix, and μ is a weight coefficient. Furthermore, this method can deal with constraints about the range of the joint angle motion from the null space $\boldsymbol{k}$ of the generalized inverse matrix [27]. If the solution of inverse kinematics does not exist or the constraints are violated, we regard the motion as infeasible.

3.3. Step B: Optimize the Joint Angular Velocity Parameter of the Via-point

Next, we calculate the joint angular velocity parameters $\dot{\boldsymbol{\theta}}'_{via}$ of the via-point based on the result obtained in **Step A**. In this step, we consider not only the new conditions $\boldsymbol{V}'$ for achieving the task but also various constraints through the motion. This problem is solved by the following algorithm with a Linear Programming (LP) problem. This simple iterative method include the calculation of appropriate upper and lower limits ($\overline{\boldsymbol{L}_{ieq}}$ and $\underline{\boldsymbol{L}_{ieq}}$) of the linear constraint function at the via-point.

Step B1.

Set the starting limit values $\overline{\boldsymbol{L}_{ieq,0}}$ and $\underline{\boldsymbol{L}_{ieq,0}}$.

Step B2.

Calculate the $\dot{\boldsymbol{\theta}}'_{via}$ with LP problem.

$$\arg\min_{\dot{\boldsymbol{\theta}}'_{via}} \quad \boldsymbol{F} \dot{\boldsymbol{\theta}}'_{via} \tag{16}$$

$$\text{s.t.} \quad \boldsymbol{J}_{eq}(\boldsymbol{\theta}'_{via})\, \dot{\boldsymbol{\theta}}'_{via} = \boldsymbol{V}_{eq} \tag{17}$$

$$\text{and} \quad \underline{\boldsymbol{L}_{ieq}} \leq \boldsymbol{J}_{ieq}(\boldsymbol{\theta}'_{via})\, \dot{\boldsymbol{\theta}}'_{via} \leq \overline{\boldsymbol{L}_{ieq}} \tag{18}$$

Step B3.

If the constraint conditions are satisfied, halt the computation.

Step B4.

Else, update the values of upper and lower limits $\overline{\boldsymbol{L}_{ieq}}$ and $\underline{\boldsymbol{L}_{ieq}}$, and go to **Step B2**.

where $\boldsymbol{F}$ in the calculation of **Step B2** is the matrix for representing the cost function. The Jacobian $\boldsymbol{J}_{eq}(\boldsymbol{\theta}'_{via})$ and the vector $\boldsymbol{V}_{eq}$ are related to the equality constraint conditions. The inequality constraint conditions are represented by the Jacobian $\boldsymbol{J}_{ieq}(\boldsymbol{\theta}'_{via})$, and the vectors of $\overline{\boldsymbol{L}_{ieq}}$ and $\underline{\boldsymbol{L}_{ieq}}$. From appropriate values of $\overline{\boldsymbol{L}_{ieq}}$ and $\underline{\boldsymbol{L}_{ieq}}$, we can obtain the motions that satisfy various constraints over whole motion duration. According to the process **P4** in Section 3.1, we can appropriately update the values of $\overline{\boldsymbol{L}_{ieq}}$ and $\underline{\boldsymbol{L}_{ieq}}$.

Indeed, the cost function and equality constraints are related to the new conditions $\boldsymbol{V}'$ for achieving the task. Furthermore, the various constraints that have to be satisfied over whole motion duration are represented by Equation (18) which corresponds to the linear inequality constraint at the via-point. We can express various linear constraints at the via-point corresponding to the constraints over whole motion duration; it is based on the transformation of Equation (11). The linear conditions for the range of joint angle motion and collision avoidance can be represented as follows.

$$\underline{\dot{\boldsymbol{\theta}}'_{via}} \leq \dot{\boldsymbol{\theta}}'_{via} \leq \overline{\dot{\boldsymbol{\theta}}'_{via}} \tag{19}$$

$$\underline{\boldsymbol{V}_{via}} \leq \boldsymbol{J}_{via}(\boldsymbol{\theta}'_{via})\, \dot{\boldsymbol{\theta}}'_{via} \leq \overline{\boldsymbol{V}_{via}}, \tag{20}$$

where $\overline{\dot{\boldsymbol{\theta}}'_{via}}$ and $\underline{\dot{\boldsymbol{\theta}}'_{via}}$ are upper and lower limits of $\dot{\boldsymbol{\theta}}'_{via}$. $\boldsymbol{J}_{via}(\boldsymbol{\theta}'_{via})$ is a Jacobian related to velocities, and $\overline{\boldsymbol{V}_{via}}$ and $\underline{\boldsymbol{V}_{via}}$ are upper and lower limits of end effector's velocities.

Furthermore, the conditions of ZMP for maintaining balance are represented by linear approximation on the basis of the centroidal angular momentum (described in **Appendix 1**). These conditions are also expressed by a linear function of the via-point.

$$\underline{\boldsymbol{L}_{G,via}} \leq \boldsymbol{J}_{G,via}(\boldsymbol{\theta}'_{via})\, \dot{\boldsymbol{\theta}}'_{via} \leq \overline{\boldsymbol{L}_{G,via}}, \tag{21}$$

where $\overline{\boldsymbol{L}_{G,via}}$ and $\underline{\boldsymbol{L}_{G,via}}$ are upper and lower limits of centroidal angular momentum at the via-point, and $\boldsymbol{J}_{G,via}(\boldsymbol{\theta}'_{via})$ is Jacobian related to the centroidal angular momentum. The inequality constraint conditions involve Equations (19), (20) and (21) in the motion planning for humanoid robots.

4. Application to Kicking Motion

We applied our method to the generation of a kicking motion for a HOAP-3 humanoid robot. In the planning of the kicking motion, even if the constraint conditions of maintaining balance and the range of joint angle motion are satisfied during the motion, the motion seems to be fail if the robot does not score a goal. In an actual soccer game, the robot must have the ability to deal with changes in the conditions for scoring a goal; i.e., direction and speed of the ball.

In the kicking motion, the most important point is the instant that the robot kicks the ball. In this research, we consider the target information for accomplishing as position $\boldsymbol{P}$, posture $\boldsymbol{R}$ and velocity $\boldsymbol{V}$ of the swing foot's toe as shown in Figure 2. A humanoid robot can perform

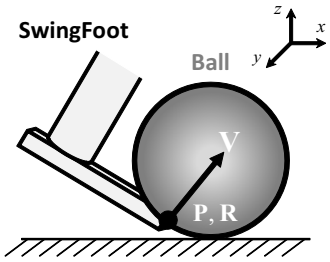


Figure 2. At the moment when a humanoid robot kicks a ball, position $\boldsymbol{P}$, posture $\boldsymbol{R}$ and velocity $\boldsymbol{V}$ are constrained for accomplishing a kicking task.

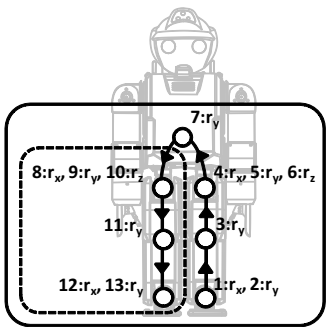


Figure 3. Degrees of freedom of a HOAP-3 humanoid robot (solid line: controlled joints in the base motion; dotted line: planned joints from the base motion; r_x, r_y and r_z denote rotatable axes in x, y and z directions, respectively).

a variety of kicking motion (e.g., the motions that kick the ball in various directions) according to the surrounding environment by considering the information at the time of kicking the ball. The instant that the robot kicks a ball is constrained as the via-point.

4.1. Planning of Kicking Motion for a HOAP-3 Humanoid Robot

We use a HOAP-3 humanoid robot that has 28 degrees of freedom, height of 60 cm, and weight of 8.8 kg. Controlled joints include 12 motors for the lower body and one motor for the upper body. Totally, 13 motors are controlled in generating the kicking motion (Figure 3). The motion of kicking the ball and supporting the body is performed by the right and left feet, respectively. The base motion obtained by our previous method [19] is to kick a ball straightforward along the ground. From the base motion, we consider achieving both the task and maintaining balance, and select the planning joints. In generating the kicking motion for the environments, the six trajectories of swing leg's joints 8 to 13 are modified from the base motion (Figure 3).

We set the experimental environment of the kicking motion in three-dimensional space. The location of the ball (d_x, d_y, d_z) and the kicking point (θ_b, ϕ_b) and the posture of the swing foot (θ_f, ϕ_f) are illustrated in Figure 4. At first, we set the movement and impact time (t_f and t_{via})of kicking the ball to 1.5 s and 0.8 s, respectively. In this application to

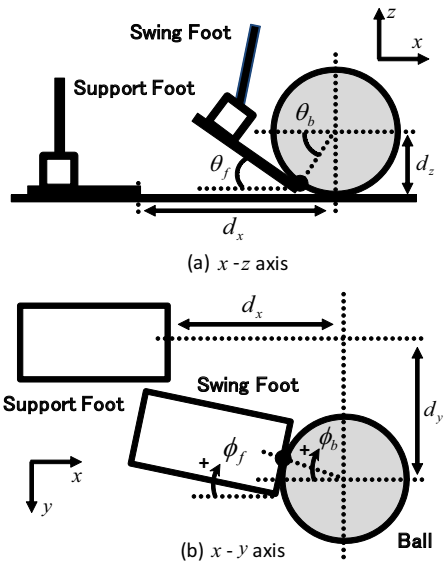


Figure 4. Experimental environment of the kicking motion (d: distance from the center of the ball to the toe of the support foot; θ_f, ϕ_f: angles of the swing foot relative to the horizontal and vertical axes; θ_b, ϕ_b: angles of the line between the center of the ball and the contact point).

kicking motion, we focus on the course of the ball. We performed two types of kicking motion for Environments 1 and 2; kicking a ball on the ground and in the air, respectively.

4.1.1. Kicking a ball on the ground

In Environment 1, the ball is kicked along the ground in a direction to the left or right of exactly straight ahead (i.e., the course of the ball is changed in x-y space). Table 1 gives the specific conditions of the experimental environment. The target positions $\boldsymbol{P}'$ and postures $\boldsymbol{R}'$ can be calculated according to these conditions. Furthermore, the velocity conditions $\boldsymbol{V}'$ are determined by the cost function and equality constraint function in the optimization process **Step B2**. The criterion for maximizing the horizontal velocity V_x of the swing foot is used for kicking the ball on the ground.

$$\begin{aligned} \max \quad & V_x \\ \text{s.t.} \quad & V_y = \alpha, V_z = 0 \end{aligned} \tag{22}$$

where V_x, V_y and V_z are the velocity of swing foot in x, y and z directions, respectively. Optimal value V_x and specified value $V_y = \alpha$ can be confirmed from Table 1.

Figure 5(a) and (b) shows the results of planned toe-position trajectories for the swing foot corresponding to the conditions of Environment 1(a) and (b). The reciprocatory motions are generated by setting only one via-point for the same initial and terminal postures ($1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ in Figure 5(a) and (b)). However, the trajectories depend on the conditions of the via-point. Figure 5(a) shows that the robot can kick a ball to the right, with the swing foot moving from the left side to the right side. Figure 5(b) shows a kicking motion which is opposite to the motion in Figure 5(a) and the robot therefore kicks the ball to the left.

Table 1. Conditions of experimental environments. The values with superscript $*$ in **Step B** indicate optimal velocities. In Environment 1 of Step B, the velocity V_y with superscript α is specified value α in Equation (22).

	Step A			Step B
	Location of Ball $[cm]$ (d_x, d_y, d_z)	Posture of Swing Foot $[deg]$ (θ_f, ϕ_f)	Kicking Point $[deg]$ (θ_b, ϕ_b)	Optimized velocity* $[m/s]$ (V_x, V_y, V_z)
Environment 1	(6.5, 7.9, 3.5)	a: (0.0, 10.0) b: (0.0, −10.0)	a: (0.0, 10.0) b: (0.0, −10.0)	a: $(1.3^*, -0.13^\alpha, 0.0)$ b: $(1.3^*, 0.13^\alpha, 0.0)$
Environment 2	6.5, 7.9, 3.5)	(45.0, 0.0)	(45.0, 0.0)	(1.89*, 0.0, 0.4*)

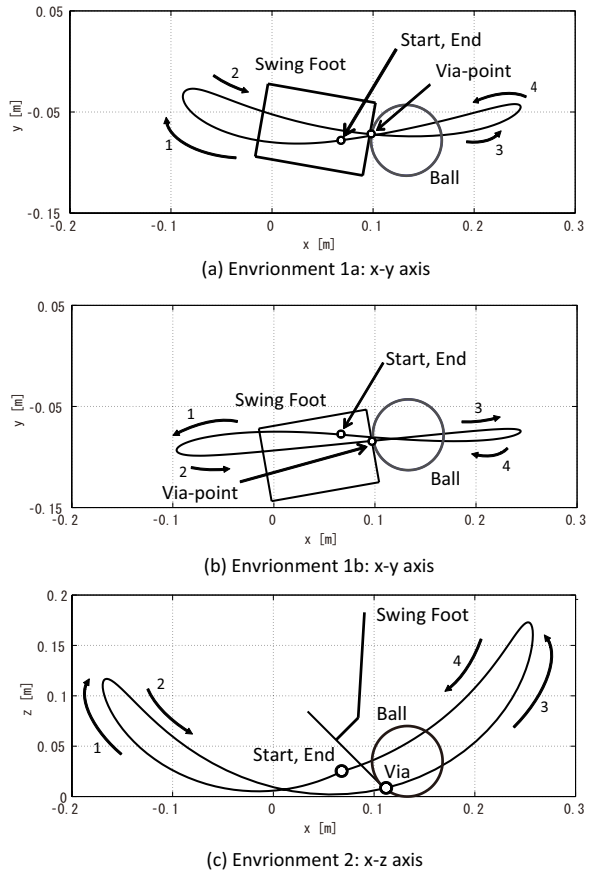


Figure 5. Planned position trajectory of the swing foot's toe.

4.1.2. Kicking a ball in the air

In Environment 2, the motion to kick the ball in the air is performed (i.e., the course of the ball is changed in x-z space). We can also confirm the specific experimental conditions from Table 1. Specifically, velocity conditions $\boldsymbol{V}'$ are optimized by using the the criterion for maximizing the flying distance. The vertical velocity of the swing foot has to be greater than zero to kick a ball in the air.

$$\begin{aligned} \max \quad & \frac{2V_xV_z}{g} \\ \text{s.t.} \quad & V_y = 0 \end{aligned} \tag{23}$$

where g is gravitational acceleration.
In Table 1, we can confirm the optimized conditions $\boldsymbol{V}_x$ and $\boldsymbol{V}_z$. Furthermore, the motion to kick the ball in the air is illustrated in Figure 5(c). We can confirm that the robot takes a reciprocatory action ($1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ in Figure 5(c)), and the posture of swing foot is angled to kick the ball in the air. As a result, the peak of the vertical position of the swing foot is almost 20 cm by setting vertical velocity greater than zero.

4.2. Experiments involving Kicking Motion for the HOAP-3 Humanoid Robot

In this section, we demonstrate the experimental results for the kicking motions generated in Section 4.1. Figure 6, 7 and 8 show scenes of kicking motions of the HOAP-3 humanoid robot corresponding to the base motion and the conditions of Environment 1 and 2. Figure 6 shows the base motion, which is the motion to kick a ball along the ground. The robot was able to score a goal if there was no obstacle. However, with an obstacle in front or to the left (or right) side of the goal, the kicking motion with base motion would result in failure. When the obstacle is on the left side of the goal, the way to score a goal is that robot kicks the ball on the left side so that its trajectory is to the right of the obstacle. Under the conditions of Environment 1 in Table 1, we generated the motion to kick a ball on the left (or right) side. Figure 7a shows the kicking of the ball to the right of the obstacle; the robot successfully completes the kicking task. In the opposite direction of kicking the ball to the left of an obstacle, the robot could again score a goal (Figure 7b). Figure 8 shows the experimental result for the conditions of Environment 2 in Table 1. In this case, there is an obstacle directly in front of the goal, and the robot has to kick the ball in the air over the obstacle. We confirmed that the ball was in the air and reached the goal over the obstacle as shown Figure 8. Therefore, the robot accomplished the kicking task under a variety of surrounding conditions. Video of the kicking motions are shown in Multimedia 1;

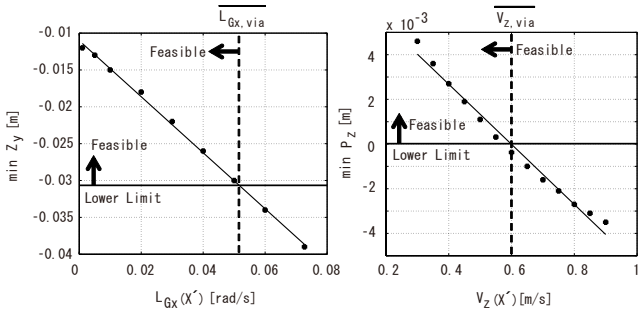


Figure 6. Limitations of velocity at the via-point and bound values of nonlinear constraints. (a) relation between the upper limit of centrical angular momentum at the via-point in x axis and the lower bound of planned ZMP trajectory in y axis. (b) relation between the upper limit of swing foot's velocity at the via-point in z axis and the lower bound of planned swing foot's position trajectory in z axis.

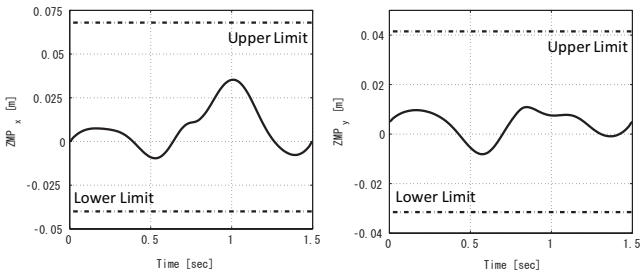


Figure 7. Profiles of ZMP of planned trajectories on x and y axes in Environment 2 (solid line: trajectories using proposed method; dash-dotted line: upper and lower limits of the support foot).

in particular, we confirmed that the movements of the robot were very smooth.

5. Discussion

In our method, the feasible motions could be planned by modifying via-point constraints. Together with these processes, the motion planning problem was formulated as a simple iterative method with a LP problem. We expect that the computation time would be greatly decreased as compared to previous method using SIP. We also present a comparison of computation time with our previous method [19].
The computational environment was an Intel(R) Core(TM) i7 860 central processing unit operating at 2.80 and 2.79 GHz with 4.00 GB memory and running Windows 7. In this work, the linprog() function of MATLAB MathWorks Inc. was used to solve the LP problem.

5.1. Modification of Constraint Conditions

In the application of generating kicking motion, the desired upper and lower limits $\overline{L_{ieq}}$ and $\underline{L_{ieq}}$ in Equation (18) could be modified by the

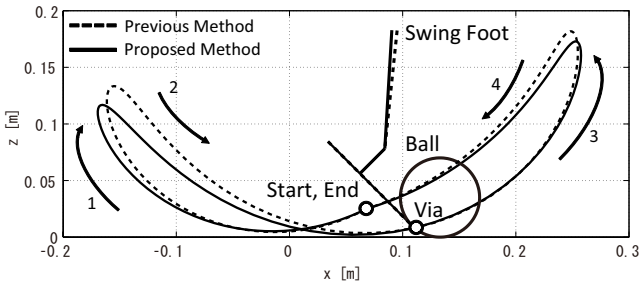


Figure 8. Comparison of the planned position trajectory of the swing foot's toe in Environment 2 (solid line: trajectory using the proposed method; dotted line: trajectory using the previous method).

three iterations. Figure 9(a) shows the relationship between the upper limit of centroidal angular momentum in the x-direction and the calculated results of the minimum value of ZMP in the y-direction. The value of centroidal angular momentum in the x (and y)-direction is related to the value of ZMP in the y (and x)-direction. We can confirm that the function is monotonically increasing. With respect to the constraints of collision avoidance with the ground, we could confirm approximately linear relations between the limit value and the calculated values in Figure 9(b). The other constraints have a similar relationship with Figure 9. By calculating the size of correction from the first and second iterations, we could obtain appropriate gradient values to satisfy the constraints in the third iteration.
Figure 10 shows the planned ZMP profiles of planned trajectories in Environment 2. There are sufficient margins between the ZMP and the boundary of the base of support. For the conditions of Environment 1, we confirmed that the planned ZMP trajectories also have sufficient margins. Therefore, the robot is able to maintain balance in each kicking motion. Furthermore, we can confirm that the condition for collision avoidance with the ground is satisfied from the planned toe-position trajectories for the swing foot in Figure 5(c).

5.2. Comparison with the Previous Method

Table 2 shows the averages and standard deviations of computation time of the proposed and previous methods. Fifty trajectories were

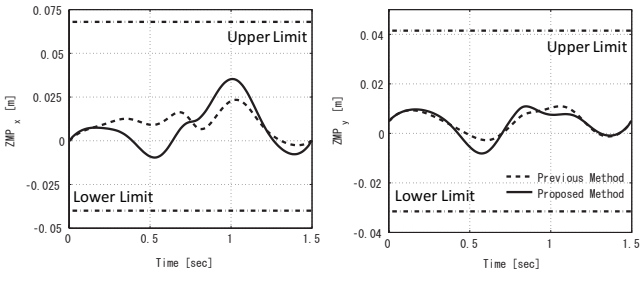


Figure 9. Comparison of the planned ZMP trajectories in Environment 2 (solid line: trajectories using the proposed method; dotted line: trajectories using the previous method; dash-dotted line: upper and lower limits of the support basis on x and y axes.

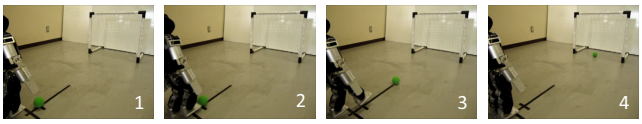
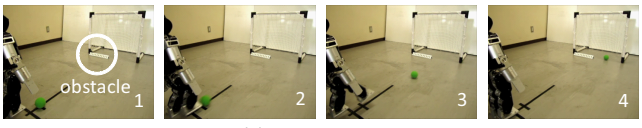
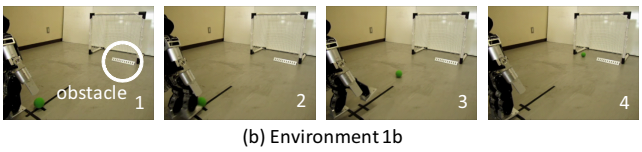


Figure 10. Experimental scenes of base motion by the HOAP-3



(a) Environment 1a



(b) Environment 1b

Figure 11. Experimental scenes of kicking motion in Environment 1 by the HOAP-3

calculated under fifty points of ball location from 2.5 cm to 5 cm. We can confirm that the computation time for the motions to kick a ball on the ground and in the air were less than two seconds when using the proposed method. The computation time was about 0.15 seconds for **Step A**, and about 0.5 seconds per one iteration for **Step B**. In the case of three iterations, the computation time of **Step B** becomes almost 1.7 seconds. Therefore, the robot can plan a feasible motion less than 2 seconds.

Compared with the proposed method, the computation time of the previous method was over 2000 seconds. In particular, in the planning of the motion to kick the ball in the air, over 4000 seconds was required. There are two reasons for the much shorter computation time of the proposed method. First, the calculation process was divided into two steps. The joint angles of the via-point were first calculated, and the joint angular velocity of the via-point was then optimized as LP problem, thus further reducing the computation cost. Next, a number of nonlinear inequality constraint conditions were represented as only a few linear constraints of the via-point velocity. Consequently, the proposed method is much faster than the previous method in calculating whole-body motion of a humanoid robot corresponding to the various constraints.

In both methods, the precisions with respect to the equality constraints are comparable, because the precision depends on the tolerance on the violation of the equality constraints. The simplification of optimization affects the optimality of the cost function in Equations (8) and (16). In the proposed method, the cost function must be a linear function of the via-point velocity, and the search space is reduced in comparison with the SIP problem. Figure 11 shows the trajectories of the swing foot that were planned using the previous and proposed methods. We confirm that the patterns of both trajectories are similar. Both trajectories take reciprocatory actions ($1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ in Figure 11). Although the trajectories are not fully consistent, both trajectories were constrained at the same via-point and satisfied the given conditions for achieving the task. Therefore, the humanoid robot could successfully carry out the task. The ZMP trajectories planned using each method are shown in Figure 12. For both trajectories, the ZMP is located within the base of the support with sufficient margins. Consequently, the humanoid robot could maintain balance during the motion.

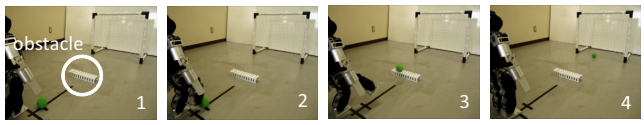


Figure 12. Experimental scenes of kicking motion in Environment 2 by the HOAP-3

6. Conclusion

We proposed a fast planning algorithm for the whole-body motions of humanoid robots. The motions satisfying various constraints were planned by specifying via-point constraints. The via-point constraints that consist of each joint angle and angular velocity of the robot were determined by an optimization process for achieving the task. The feasible motions could be planned rapidly by modifying the via-points from the base motion. In our method, the appropriate via-point parameters were calculated by dividing the process into two sequential steps. Specifically, various nonlinear inequality constraints over whole motion duration were transformed into linear inequality constraints at the via-points. Together with these transformations, the number of constraint conditions was greatly reduced. Furthermore, we could formulate the motion planning problem as a simple iterative method with a LP problem. Finally, the computation time was greatly reduced when we used the proposed method, and the robot could plan the various motions in a short timeframe.

Our method is applied to generate the kicking motion for a HOAP-3 humanoid robot. In generating kicking motion, the moment of kicking a ball was constrained as a via-point. Implementing our proposed algorithm, the HOAP-3 humanoid robot kicked a ball in various directions in three-dimensional space. The HOAP-3 humanoid robot could complete the kicking task even if the location of an obstacle was changed. Furthermore, the computation time was about 1.5 seconds. Our method requires much less computation time than our previous method based on a SIP problem. Furthermore, we could confirm that the proposed method kept up the motion performance compared with the previous method. These results indicate that the proposed algorithm can achieve efficient motion planning for humanoid robots.

However, we assumed in this research that the appropriate timing of the via-point was given. It is desired that the humanoid robot autonomously determines the timing to be useful in the human environment. We intend to investigate how to acquire and modify the timing of via-points to accomplish a task. On the other hand, we used only a single via-point in generating kicking motions. We expect that the motion performance would be improved by specifying several via-points. However, as more via-points are specified, the computation time might increase. Furthermore, the question of how many via-points should be used in motion planning has not been examined - this is problem awaiting solution, and work is in progress. Finally, together with these works, we will expand the motion planning algorithm to more complex whole-body tasks for humanoid robots.

Acknowledgements

This study was supported by a Grant-in-Aid for Scientific Research (B) No.21300092 and (C) No.23560526.

Table 2. Comparison of the computation time. In proposed method, the total computation time is calculated by the time for **Step A** + the time for **Step B** × iteration number. SD means standard variation.

	Kicking a ball on the ground $[sec]$	Kicking a ball in the air $[sec]$
Proposed method	Step A: 0.156 (SD: 0.023) Step B: 0.522 (SD: 0.011) × 3 Total Time: 1.723 (SD: 0.045)	Step A: 0.166 (SD: 0.018) Step B: 0.525 (SD: 0.012) × 3 Total Time: 1.741 (SD: 0.045)
Previous method	2490.38 (SD: 747.89)	4407.60 (SD: 5725.01)

Appendix 1: Maintain Balance with Centroidal Angular Momentum

In this research, we consider the conditions for maintaining the balance on the basis of the centroidal angular momentum (CAM). The center of mass (COM) is an important factor related to the zero moment point (ZMP). In addition to the COM, CAM has been mentioned as an essential factor of the ZMP [28–30]. The reaction mass pendulum (RMP) model [31] is proposed for the balance control model of humanoid robots based on the COM and CAM. The ZMP based on the RMP model is given by

$$\begin{aligned} Z_x &= c_x - \frac{c_z \ddot{c}_x + \dot{L}_{Gy}/m}{\ddot{c}_z + g} \\ Z_y &= c_y - \frac{c_z \ddot{c}_y - \dot{L}_{Gx}/m}{\ddot{c}_z + g}, \end{aligned} \tag{24}$$

where $\boldsymbol{c} = (c_x, c_y, c_z)$ is the position of the COM and $\boldsymbol{L}_G = (L_{Gx}, L_{Gy}, L_{Gz})$ is the CAM in x, y and z directions. Additionally, $\boldsymbol{Z} = (Z_x, Z_y)$ describes the position of the ZMP in the x-y plane. m is the total mass of the robot and g is gravitational acceleration. We ignore the term of $\ddot{c}_z$, because g is usually much greater than $\ddot{c}_z$ in kicking motion. The terms for the COM are then combined as

$$\begin{aligned} Z_x &= h_x(c_x, \ddot{c}_x, c_z) - \frac{\dot{L}_{Gy}}{mg}, \quad \left(h_x(c_x, \ddot{c}_x, c_z) = c_x - \frac{c_z \ddot{c}_x}{g} \right) \\ Z_y &= h_y(c_y, \ddot{c}_y, c_z) + \frac{\dot{L}_{Gx}}{mg}, \quad \left(h_y(c_y, \ddot{c}_y, c_z) = c_y - \frac{c_z \ddot{c}_y}{g} \right). \end{aligned} \tag{25}$$

Since the planned trajectories of the joints of the support leg are the same as the base motion, the COM of the planned trajectory can be regarded as that of the base motion. Therefore, the function $h_x(c_x, \ddot{c}_x, c_z)$ and $h_y(c_y, \ddot{c}_y c_z)$ can be constant and are given by the values of the base motion. The conditions for maintaining balance in the proposed method are taken into account by inequality conditions of centroidal angular momentum at the via-point.

References

[1] S. Kajita, F. Kanehiro, K. Yokoi and H. Hirukawa, IEEE International Conference on the Intelligent Robots and Systems, 239-246, 2001, Maui, USA.
[2] T. Sugihara, Y. Nakamura and H. Inoue, IEEE International Conference on Robotics and Automation, 11-15, 2002, Washington, DC, USA
[3] S. Kajita, F. Kanehiro, K. Fujikawa, K. Harada, K. Yokoi, H. Hirukawa, IEEE International Conference on Robotics and Automation, 14-19, 2003, Taipei, Taiwan.
[4] D.P. Bertsekas, Constrained optimization and Lagrange Multiplier methods (Academix Press, Boston, 1982).
[5] R. Hettich and K.O. Kortanek, SIAM Review, 3, 53 (1993).
[6] A. Ude and CG. Atkeson and M. Riley, IEEE International Conference on Robotics and Automation, 2223-2228, 2000, San Francisco, CA, USA.
[7] S. Miossec, K. Yokoi and A. Kheddar, IEEE International Conference on Robotics and Biomimetics, 299-304, 2006, Beijing, China.
[8] S. Carpin and E. Pagello, IEEE-RAS International Conference on Humanoid Robots, 71-77, 2006, Genova, Italy.
[9] S. Lengagne, N. Ramdani and P. Fraisse, IEEE Transactions on Robotics, 6, 27 (2011).
[10] J.H. Kim, Mechanism and Machine Theory, 4, 46 (2011).
[11] S. Lengagne, A. Kheddar, E. Yoshida, Workshop on Humanoid Service Robot Navigation in Crowded and Dynamic Environments at the IEEE Humanoids Conference, 26-28, 2011, Bled, Slovenia.
[12] S. Suzuki, Y. Tazaki, T. Suzuki, IEEE-RAS International Conference on Humanoid Robots, 596-601, 2011, Bled, Slovenia.
[13] S. Kudoh, T. Komura and K. Ikeuchi, IEEE International Conference on the Intelligent Robots and Systems, 2563-2568, 2002, EPFL Lausanne, Switzerland.
[14] A. Konno, T. Myojin, T. Matsumoto, T. Tsujita and M. Uchiyama, The international Journal of Robotics Research, 13, 30 (2011).
[15] R. Bulirsch, A. Stoer, K.-H. Well (eds), Optimal Control Theory and Numerical Method (International Series of Numerical Mathematics, Basel, Switzerland, 1993).
[16] T. Flash and H. Horgan, The Journal of Neuroscience, 7, 5 (1985).
[17] Y. Uno, M. Kawato and R. Suzuki, Biological Cybernetics, 2, 6 (1989).
[18] H. Miyamoto, S. Schaal, F. Gandolfo, H. Gomi, Y. Koike, R. Osu, E. Nakano, Y. Wada and M. Kawato, Neural Networks, 8, 9 (1996).
[19] CH. Sung, T. Kagawa and Y. Uno, International Conference on Ubiquitous Robots and Ambient Intelligence, 362-367, 2011, Incheon, Korea.

[20] S. I. Gass, Linear Programming: Methods and Applications, 5th edition (McGraw-Hill, New York, 1995).
[21] A. Piazzi and A. Visoli, IEEE Transactions on Industrial Electronics, 1, 47 (2000).
[22] CGL. Bianco and A. Piazzi, International Journal of Control, 13, 75 (2002).
[23] M. Vukobratovia and B. Borovac, International Journal of Humanoid Robotics, 1, 1 (2004).
[24] C. T. Kelly, Solving Nonlinear Equations with Newton's Method (The Society for Industrial and Applied Mathematics, Philadelpha, 1987).
[25] RM. Murray, Z. Li and SS. Sastry, A Mathematical Introduction to Robotic Manipulation (CRC Press, Florida, 1994).
[26] M. Stilman, IEEE Transactions on Robotics, 3, 26 (2010).
[27] M. Gienger, M. Toussaint and C. Goerick, Motion Planning for Humanoid Robots (Springer, London, 2010).
[28] M. Abdallah, A. Goswami, IEEE International Conference on Robotics and Automation, 1996-2001, 2005, Barcelona, Spain.
[29] A. Goswami and V. Kallem, IEEE International Conference on Robotics and Automation, 3785-3790, 2004, LA, USA.
[30] J. Pratt, J. Carff, S. Drakunov and A. Goswami, IEEE-RAS International Conference on Humanoid Robots, 200-207, 2006, Genova, Italy.
[31] SH. Lee and A. Goswami, IEEE International Conference on Robotics and Automation, 4667-4672, 2007, California, USA.
[32] N. Perrin, O. Stasse, L. Baudouin, F. Lamiraux and E. Yoshida, IEEE Transactions on Robotics, 28, 2 (2012).
[33] A. Hornung and M. Bennewitz, IEEE International Conference on Robotics and Automation, 997-1002, 2012, Minnesota, USA.
[34] H. Noborio, T. Naniwa and S. Ariomoto, IEEE International Conference on Robotics and Automation, 327-332, 1989, Arizona, USA.
[35] A. Dasgupta, Y. Nakamura, IEEE International Conference on Robotics and Automation, 1044-1049, 1999, Detroit, MI, USA.
[36] T. Bretl, S. Rock and J.-C. Latombe, IEEE International Conference on Robotics and Automation, 2946-2953, 2003, Taipei, Taiwan.