

Two-Stage Time-Parametrized Gait Planning for Humanoid Robots

Kensuke Harada, *Member, IEEE*, Shizuko Hattori, Hirohisa Hirukawa, *Member, IEEE*, Mitsuharu Morisawa, Shuuji Kajita, *Member, IEEE*, and Eiichi Yoshida, *Member, IEEE*

Abstract—In this paper, we propose a framework on planning collision-free walking motion for biped humanoid robots. Our proposed planner is composed of two phases. In the first phase, the constraint condition is generated as a function of time by using the walking pattern generator. In the second phase, the collision-free walking motion is planned. To generate the collision-free motion, we add a time parameter to each milestone of the single-query, bidirectional, lazy collision checking planner in order to explicitly consider the time-parametrized constraint conditions. Further, we smoothen the generated path by using B-spline interpolation. Through experimental results, we show that our planner is effective in realizing collision-free walking motion by real humanoid robots.

Index Terms—Biped gait, humanoid robot, motion planning.

I. INTRODUCTION

IN RECENT years, many humanoid robots capable of performing complex and human-like whole-body motion have been developed [1]. By observing the motion of humanoid robots, it is difficult for us to realize that human beings typically select motion parameters carefully in order to avoid unnecessary collisions of links.

With regard to the collision-free motion planning research, random-sampling based methods [2] and [3] have been extensively researched and have been applied to several systems including the robotic manipulator [4], mobile robots [5], and biomechanics [6]. In recent years, random-sampling-based methods have been applied to the motion planning of humanoid robots [7]–[11]. The authors have first proposed a framework for the collision-free whole-body motion of a biped humanoid robot walking on flat/irregular terrain while maintaining dynamic balance [12].

Since biped gait for humanoid robots has been extensively researched [13]–[19], this paper aims to include these previous results into the random-sampling-based planner. Fig. 1 shows a typical case in which our research can be applied. From this example, we can see that the humanoid robot HRP-2 walks through the gate by using the joints of the whole body.

Let us consider planning the walking pattern of a humanoid robot. By using most of the existing walking pattern generator, a walking motion has been generated for a given trajectory of

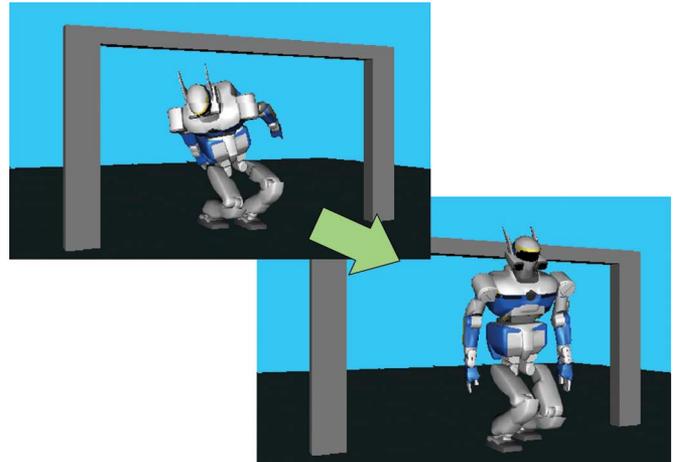


Fig. 1. Humanoid robot walking through the gate.

the zero moment point (ZMP). Once the desired ZMP trajectory is given, the position/orientation of the feet are determined so as to include the ZMP in the support polygon. Then, by solving an ordinary differential equation, the trajectory of the center of gravity (CoG) is calculated. To keep the robot balanced, the CoG trajectory works as time-parametrized constraint conditions imposed on the robot's configuration. Hence, when planning the collision-free motion of a humanoid robot, we have to consider these time-parametrized constraint conditions.

In order to include this constraint condition into the motion planner, we propose a two-stage time-parametrized motion planning method for a biped humanoid robot. In the first stage, we generate the time-parametrized constraint conditions by running the walking pattern generator. Simultaneously, we monitor whether or not unnecessary collision occurs in this stage. In the second stage, we then plan the collision-free motion considering these time-parametrized constraint conditions. Since our proposed planner includes the time parameter as additional information included in each milestone of the tree, we can generate a path connecting the start and the goal configurations while maintaining the time relation.

Our planner is sophisticated in that it has the following two additional features. First, when a humanoid robot walks, the robot receives a 6-D force/moment from the environment [20]. A walking pattern generator used, in this paper, can consider a 6-D force/moment [19]. Second, in most random-sampling-based planning methods, milestones have been connected by using line segments. Since the position of the ZMP depends on the acceleration of the robot, the robot may fall down, if its

Manuscript received October 2, 2008; revised January 26, 2009; accepted July 18, 2009. Date of publication November 20, 2009; date of current version August 4, 2010. Recommended by Technical Editor V. N. Krovi.

The authors are with the Humanoid Research Group, Intelligent Systems Research Institute, National Institute of Advanced Industrial Science and Technology, Tsukuba 305-8568, Japan (e-mail: kensuke.harada@aist.go.jp).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TMECH.2009.2032180

motion includes velocity discontinuity. Here, if each milestone is connected by using a line segment, velocity discontinuity occurs when connecting two line segments. Hence, when planning the motion of the humanoid robot, we have to generate a smooth path in which velocity discontinuity is avoided.

This paper is organized as follows. After describing the related works in Section II, we present an overview of the proposed method in Section III. In Section IV, we explain the method used for generating the walking motion while considering the 6-D force/moment. In Section V, we explain the planning method for collision-free motion considering the time-parameter. We further explain the method used for smoothen the generated path. The effectiveness of our proposed method is validated through the numerical examples and experimental results described in Section VI.

II. RELATED WORK

A. Walking Pattern Generation of Humanoid Robots

With regard to the walking pattern generation of humanoid robots, many researchers proposed the ZMP-based methods in which they solved the differential equation expressing the relationship between the ZMP and the robot's motion. Takamishi *et al.* [13] proposed a method to calculate the trunk motion by using fast Fourier transform (FFT). Kagami *et al.* [14] proposed an approach to numerically solve the differential equation of the trunk motion. Kajita *et al.* [15] proposed a method based on preview control. Nagasaka *et al.* [16], Harada *et al.* [17], and Sugihara *et al.* [18] proposed analytical-solution-based approaches.

Recently, Hirukawa *et al.* [20] proposed a method for judging the contact-state transition between the robot and the environment by using a contact wrench. They later applied it to the walking pattern generation of a biped humanoid robot [19].

However, the above researches do not consider the collision of the links except for the contact between the feet and the ground.

B. Motion Planning with Collision Avoidance

Research aiming to realize whole-body motion while avoiding collisions employs one of two approaches; the first is the instantaneous approach [21] using the potential field method [22] and the second is offline planning approaches [7]–[11], and [24] using probabilistic methods [2], [3].

With regard to the instantaneous approach, Sentis *et al.* [21] formulated a whole-body motion control framework. Within this framework, they considered the avoidance of the collision between an obstacle and a link of the robot. Stasse *et al.* [23] proposed another approach that can deal with unilateral constraints.

The advantage of the probabilistic methods is that they can plan the motion of a robot having many DOF within a reasonable time by considering a number of contact pairs simultaneously. Kavraki *et al.* [2] proposed the probabilistic roadmap (PRM) planner. Then, Kuffner and Lavelle proposed the single query method called rapidly exploring random trees (RRT) [3]. Sanchez *et al.* [25] also proposed the single query method called

the single-query, bidirectional, lazy in collision checking (SBL). The random sampling method has been applied to a mechanical system with a closed kinematic chain [26], [27], and with velocity constraints [4], [28].

Recently, some researchers applied probabilistic methods to humanoid motion planning. Kuffner *et al.* [7], [8] assumed that the position of the feet does not change during the motion and proposed a two-stage method. In their method, they first generated collision-free motion while maintaining the static balance of the robot, and then, they transformed it to the a dynamically balanced one by using a balance compensator [29]. Here, a dynamically stable gait is generated by solving the differential equation for a given period of time. In particular, it is almost impossible to obtain quick biped-gait motion by simply transforming the statically balanced motion using the balance compensator [29]. Yoshida [9] approximated the shape of the robot to be a rectangular convex and extended the approach for an omni-directional vehicle. Hauser *et al.* [10], [11] proposed the multistep planning method applicable to a rock-climbing humanoid robot while maintaining the static balance. Sanada *et al.* [30] also proposed a planning method for a humanoid robot while maintaining static balance.

Recently, the number of humanoid motion planning has been increased. Chestnutt *et al.* [31] proposed a method for planning a foot-step without considering the upper-body motion. Stilman *et al.* [32] proposed the manipulation planning of a movable object. Harada *et al.* [33] proposed the pushing manipulation of a heavy object placed on the ground.

The contribution of this research is to combine the walking pattern generator and the collision-free motion planner. For this purpose, we newly supply a two-stage time-parametrized framework for the whole-body motion planning of a humanoid robot walking on flat/rough terrain while maintaining the dynamic balance.

III. OVERVIEW OF PROPOSED METHOD

Fig. 2 shows a model of the humanoid robot used in this research. Let \mathbf{p}_*/ϕ_* be the 3-D vectors of the position/orientation of the coordinate frame fixed to a part of the robot. The subscripts Fj , Hj , B , and G denote the j th foot, j th hand, waist, and CoG, respectively. In addition, let \mathcal{P} and \mathcal{L} be the linear and angular momentum about the CoG of the robot, respectively, and $\mathbf{g} = [0 \ 0 \ g]^T$ be the gravity force vector.

We assume that the 3-D models of the robot and the environment are known. These models are used for collision checking. A configuration $q \in \mathcal{C}$ of the humanoid robot is composed of the position/orientation of the waist (\mathbf{p}_B/ϕ_B) and all the joint angles (θ). For a biped humanoid robot, since some of the links such as the feet and the hands are constrained to the environment, we plan the robot's motion such that these links make contact with the environment at the desired position within the specified period of time. At the same time, the robot has to avoid all other collisions among its links and those between a link and the environment. $\mathcal{C}_{\text{free}} \subset \mathcal{C}$ denotes the set of configurations where such unnecessary collisions do not occur. Moreover, we impose the desired trajectories to some parts of the robot such as the

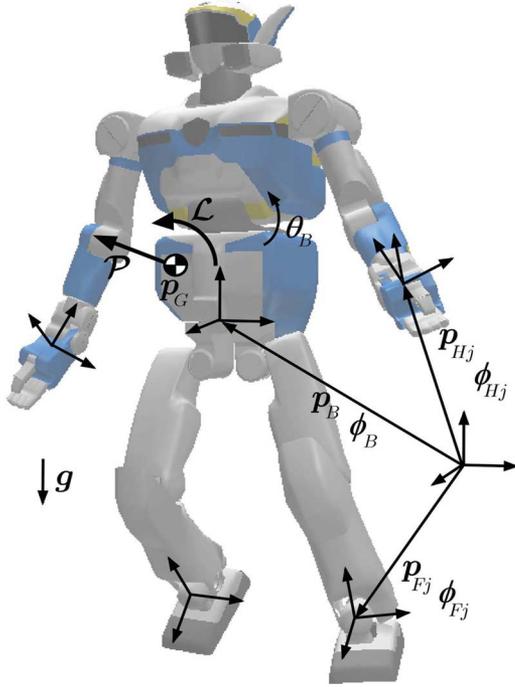


Fig. 2. Model of HRP2 humanoid robot.

feet, hands, and horizontal position of the CoG. Here, we regard the condition for the robot to follow the desired trajectory as the constraint conditions imposed on the robot's configuration. These constraint conditions are functions of both the robot's configuration q and the time t , and they have the form

$$f(q, t) = 0. \quad (1)$$

Let $\mathcal{C}_{\text{cons}}(t) \subset \mathcal{C}$ be a subset of the configuration such that the robot follows the desired trajectory at a specified time. In our motion planning problem, we search for the configuration $q(t)$ ($t_0 \leq t \leq t_n$) of the robot from the start q_{st} to the goal q_{ed} included in the set $\mathcal{C}_{\text{free}} \cap \mathcal{C}_{\text{cons}}(t)$.

Fig. 3 shows an overview of the motion planner proposed in this research. As shown in Fig. 3(a), we first run the walking pattern generator described in Section IV. While running the walking pattern generator, we monitor whether or not the unnecessary collisions occur. If collisions do not occur, the motion planner returns the joint trajectory of the robot. On the other hand, if unexpected collisions occur, we plan a collision-free path of the configuration within the period of time where collisions occur by using the method described in Section V. Then, the planner returns the collision-free joint trajectory after smoothing it by using spline interpolation.

IV. WALKING PATTERN GENERATOR

We first explain the walking pattern generator [19] used in this research. To overcome the inherent problems caused in the conventional ZMP, as described in the introduction, we consider the 6-D force/moment in order to generate the walking motion of a biped humanoid robot.

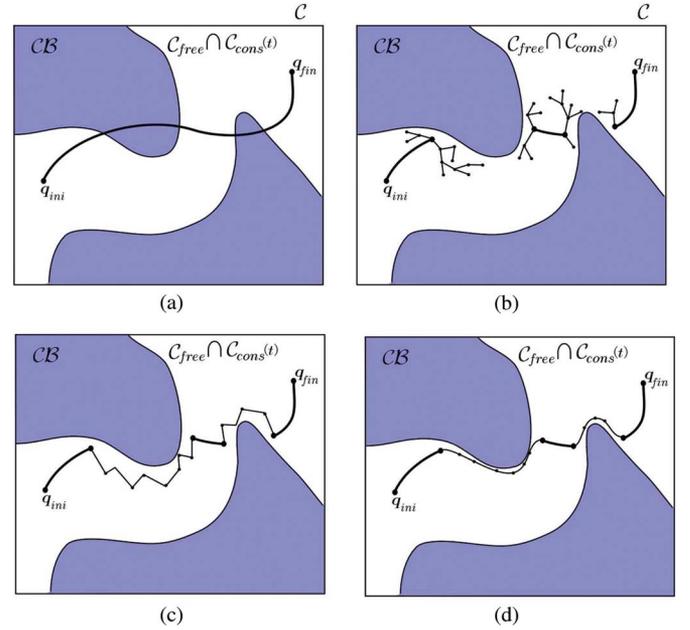


Fig. 3. Overview of the proposed method. (a) Walking pattern generation and collision detection. (b) Motion planning using PRM. (c) Path generation. (d) Shortcut and smoothing.

Let the sum of the gravity and the inertia force applied to the robot be f_G and the sum of the moments about the CoG of the robot be τ_G with respect to the reference coordinates. The contact wrench sum (CWS) applied to the robot can be represented as

$$f_G = M(g - \ddot{p}_G) \quad (2)$$

$$\tau_G = p_G M(g - \ddot{p}_G) - \dot{\mathcal{L}}. \quad (3)$$

In addition, let the contact wrench cone (CWC) be the set of CWS spanned by the sum of the friction cone at each contact point. The CWC can be represented as

$$f_C = \sum_{k=1}^K \epsilon_k \left(n_k + \sum_{l=1}^2 \delta_k^l t_k^l \right) \quad (4)$$

$$\tau_C = \sum_{k=1}^K p_k \epsilon_k \left(n_k + \sum_{l=1}^2 \delta_k^l t_k^l \right) \quad (5)$$

where n_k is the unit normal vector at the k th contact point p_k ; t_k^l ($l = 1, 2$), the unit tangent vectors at p_k ; ϵ_k is a non-negative scalar; δ_k^l is a scalar; and K is the number of contact points. A contact is defined to be strongly stable, if the CWS is an internal element of the CWC under sufficient friction. By using the CWS, the motion pattern of a humanoid robot walking on several different types of surfaces [19] has been generated.

When generating the walking pattern, we first specify the desired trajectory of the CWS. Currently, with our implementation, we can just consider 2-D moment about the horizontal axes. Then, by solving an ordinary differential equation, we obtain the horizontal trajectory of the robot's CoG. Once the CoG trajectory is obtained, we can calculate the linear/angular momentum of the robot. By using resolved momentum control [1], we can

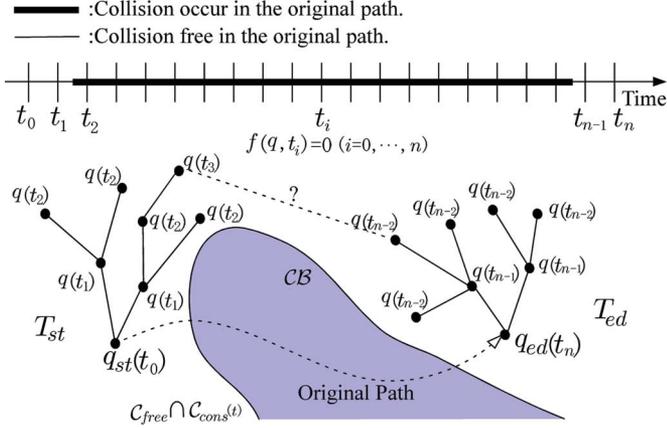


Fig. 4. Method for including the constraint condition.

transform the linear/angular momentum to the joint angles of the whole body.

As explained in the previous section, when generating the walking pattern, we record the time trajectories of the horizontal CoG, $x_G(t)$ and $y_G(t)$, the foot trajectory, \mathbf{p}_{F_i} , ϕ_{F_i} ($i = 1, 2$), the hand trajectory with respect to the chest coordinate system, ${}^C\mathbf{p}_{F_i}$, ${}^C\phi_{F_i}$ ($i = 1, 2$), and the neck joint trajectory. These trajectories are used when generating collision-free motion, as described in the next section.

V. MOTION PLANNER

Our planner is based on the SBL planner [25]. The SBL planner incrementally constructs a network of milestones m composed of two trees T_{st} and T_{ed} rooted at m_{st} and m_{ed} , respectively. The planner grows trees of collision-free milestones until a connection is found between two trees. Once a connection is found, the planner checks for collisions of the path between the two collision-free milestones included in the path.

In this section, we mainly explain how to include the walking pattern generator in the planner and how to smoothen a solution path. Fig. 4 shows an overview of the proposed planner. As shown in Fig. 4(a), while running the walking pattern generator, we record the period of time in which unnecessary collisions occur. Then, before starting the motion planner, we determine the time of the start/goal of the planner, t_0 and t_n , such that the period of collision is included between t_0 and t_n . We further consider discretizing the time span as $t = t_0, t_1, \dots, t_{n-1}, t_n$. In our proposed planner, each milestone m is composed of the configuration of the robot q and the time parameter t . If time t_i is associated with milestone m_i , we have

$$t_i = \text{time}(m_i). \quad (6)$$

In this case, the milestone m_i has to satisfy the constraint condition $f(q, t_i) = 0$.

A. Tree Expansion

We assume that the root configurations satisfy $q_{st} \in \mathcal{C}_{free} \cap \mathcal{C}_{cons}(t_0)$ and $q_{ed} \in \mathcal{C}_{free} \cap \mathcal{C}_{cons}(t_n)$. When expanding trees, we include the constraint condition using the following rule.

Algorithm 1: Tree Expansion Rule for T_{st}

- 1) A (parent) milestone m_i is picked from T_{st} where m_i satisfies $\text{time}(m_i) = t_i$ and $q(t_i) \in \mathcal{C}_{free} \cap \mathcal{C}_{cons}(t_i)$.
- 2) The child milestone m_j of the father m_i is set to satisfy $\text{time}(m_j) = t_{i+1}$ and $q(t_{i+1}) \in \mathcal{C}_{free} \cap \mathcal{C}_{cons}(t_{i+1})$.

As for T_{ed} , if the father milestone m_i satisfies $\text{time}(m_i) = t_i$ and $q(t_i) \in \mathcal{C}_{free} \cap \mathcal{C}_{cons}(t_i)$, its son m_j is set to be satisfying $\text{time}(m_j) = t_{i-1}$ and $q(t_{i-1}) \in \mathcal{C}_{free} \cap \mathcal{C}_{cons}(t_{i-1})$.

The son milestone satisfying $q(t_j) \in \mathcal{C}_{free} \cap \mathcal{C}_{cons}(t_j)$ is generated as follows.

Algorithm 2: Implementation Rule for Tree Expansion

- 1) At a distance less than ρ^1 from the father milestone, a configuration is randomly selected.
- 2) The forward kinematics is solved for the selected configuration.
- 3) We solve the inverse kinematics so that the configuration satisfies the constraint condition $f(q, t) = 0$.
- 4) Check if the new configuration is collision free satisfying $q \in \mathcal{C}_{free}$.
- 5) If Step 4 is satisfied, we select the new milestone as a son of its father. Otherwise, go to Step 1.

The forward kinematics in Step 2 calculates the position of the CoG and the position/orientation of the waist, feet, and hands for a given configuration q . On the other hand, the inverse kinematics in Step 3 calculates the configuration q for the given CoG, waist, feet, and hands. Steps 2 and 3 are explained in greater detail in Section VII-A.

B. Tree Connection

Let us consider the milestone m_i added during the previous expansion phase. We try to connect this milestone to one belonging to the other tree. Let us consider the case where the milestone m_i belongs to T_{st} and satisfies $\text{time}(m_i) = t_i$ and $q(t_i) \in \mathcal{C}_{free} \cap \mathcal{C}_{cons}(t_i)$. We use the following algorithm to connect T_{st} with T_{ed} .

Algorithm 3: Tree Connection for T_{st}

- 1) Select the milestone m_j belonging to T_{ed} closest to m_i .
- 2) We assume that $\text{time}(m_j) = t_j$. If the distance between these two milestones is less than $\tilde{\rho}$ and $t_i < t_j$, the trees are considered to be connected.

If the trees are connected, the candidate path connecting q_{st} and q_{ed} can be generated. The condition $t_i < t_j$ is required because for the path connecting milestones m_{st} and m_{ed} , the time parameter should be of an increasing order.

In the case where the new milestone m_i belongs to T_{ed} , the child milestone m_j of m_i has to satisfy $t_i > t_j$ where $\text{time}(m_i) = t_i$ and $\text{time}(m_j) = t_j$.

C. Path Generation

Once a path candidate is determined, we check for collisions of each segment connecting two milestones belonging to the path. Let us consider the segment connecting two milestones m_i and m_j . We consider the case where m_i and m_j satisfy

¹We defined the metric in the configuration space by D cube.

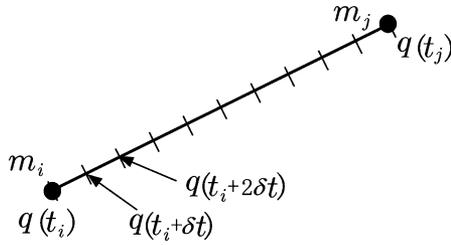


Fig. 5. Method for checking for collisions of the segment.

time(m_i) = t_i and time(m_j) = t_j , respectively. When checking for collisions of this segment, a point on the segment should satisfy the constraint condition. For the collision checking of a straight line segment, we can consider the constraint condition by using the following algorithm.

Algorithm 4: Collision-Check of a Path

- 1) Connect m_i and m_j by using a straight-line segment.
- 2) Select a point on the line segment dividing the segment in the ratio $n\delta\bar{t}/(t_j - t_i) \mid (t_j - t_i - n\delta\bar{t})/(t_j - t_i)$, $n = 1, 2, \dots$. We set the configuration of the robot without considering the constraint condition as

$$q(t_i + n\delta\bar{t}) = \frac{n\delta\bar{t}q(t_j) + (t_j - t_i - n\delta\bar{t})q(t_i)}{t_j - t_i}, \quad n = 1, 2, \dots \quad (7)$$

- 3) By using Steps 2 and 3 of Algorithm 2 with the constraint condition $f(q, t_i + n\delta\bar{t}) = 0$, a new configuration satisfying the constraint condition $q'(t_i + n\delta\bar{t}) \in \mathcal{C}_{\text{cons}}(t_i + n\delta\bar{t})$ is obtained.
- 4) Check if this configuration satisfies $q'(t_i + n\delta\bar{t}) \in \mathcal{C}_{\text{free}}$. If this condition is not satisfied, return *false*.
- 5) If the segment is collision free within the predefined resolution, return *true*. Otherwise, go to Step 2 with different $n\delta\bar{t}$.

The selection of $n\delta\bar{t}$ follows [25]. After the collision-free path is generated, we consider smoothing it by using the method described in the next section.

D. Path Smoothing

The path obtained in the previous section may not be a smooth one; it may include detours and unexpected discontinuity of velocity. Our smoothing algorithm comprises two steps: the shortcut path is found in the first step and spline interpolation is applied in the second step. In both steps, it is essential to avoid unnecessary collisions of links.

The shortcut algorithm is a simple extension of the one proposed in [25], where the time parameter is newly considered. Since the method of collision checking of a segment is exactly the same as Algorithm 4, we mainly explain spline interpolation in this section.

The path of the robot's configuration obtained by the shortcut operation may still include the discontinuity of velocity since two milestones are connected by using a straight-line segment. In the case of a humanoid robot, the discontinuity of velocity must be avoided since the robot may fall down due to the effect of a large acceleration.

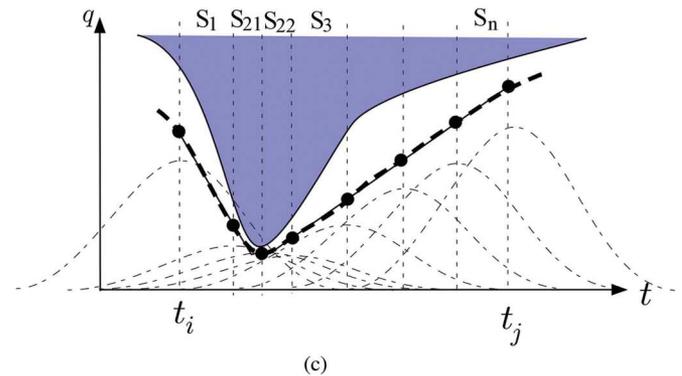
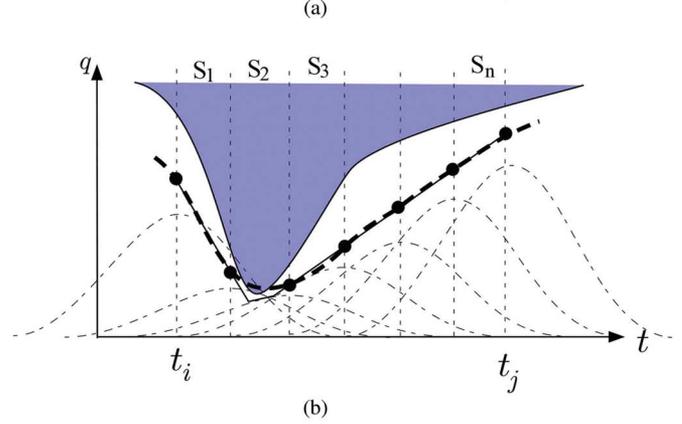
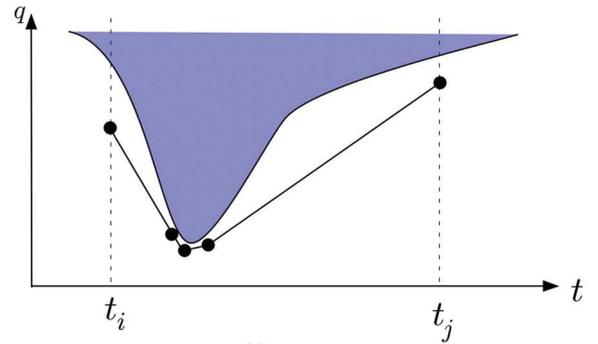


Fig. 6. Spline interpolation. (a) Input path generated by shortcut operation. (b) Initial Spline interpolation. (c) Spline interpolation inserting an additional node.

To overcome this problem, we apply b-spline interpolation. An overview of the spline interpolation algorithm is shown in Fig. 6. Fig. 6(a) shows the path obtained by the shortcut operation. Then, as shown in Fig. 6(b), we split the trajectory between $q(t_i)$ and $q(t_j)$ into n segments S_1, S_2, \dots, S_n . By applying b-spline interpolation, we can obtain a set of curved trajectories. Fig. 6(b) shows the case in which the trajectory is split into n segments.

We check for collisions of the curved segment from S_1 to S_n . A collision-checking algorithm for the curved segments can be obtained by simply modifying Step 1 of Algorithm 4 where m_i and m_j are connected by using a curved trajectory instead of a straight line. If the collision occurs in S_i , we consider adding an additional node to S_i and splitting S_i into S_{i1} and S_{i2} . Fig. 6(b) shows the case where collision occurs at S_2 . Then, Fig. 6(c)

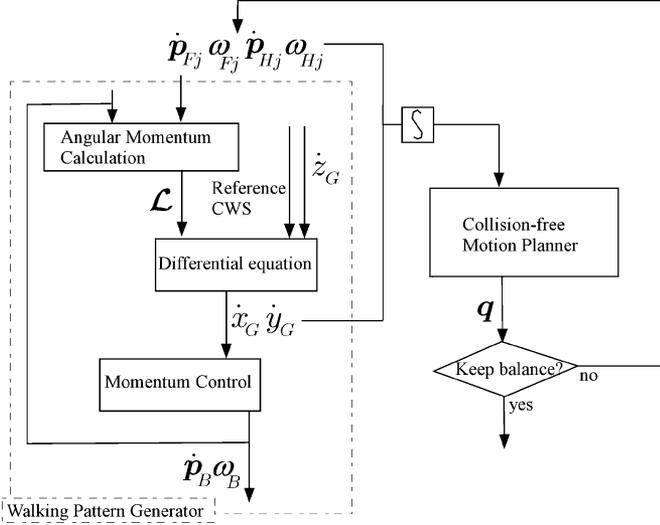


Fig. 7. Method to combine the walking pattern generator with the motion planner.

shows that an additional node is inserted to S_2 and S_2 is split into S_{21} and S_{22} . After checking for collision of S_i , we check for collision of S_{i+1} . If we finished checking for collisions of S_n , we return to S_1 . We iterate this operation until we obtain a collision-free trajectory connecting $q(t_i)$ to $q(t_j)$.

If the difference between the shortcut trajectory and the splined trajectory reduces as the number of node increases and if the minimum distance between the shortcut trajectory and the obstacle is greater than 0, we can obtain a smooth and collision-free path by using the spline interpolation algorithm. As far as we tried, as the number of nodes increased, the difference between the shortcut trajectory and the splined trajectory decreased. However, the same may not necessarily be true if we use spline interpolation.

VI. COMBINATION OF MOTION PLANNER AND WALKING PATTERN GENERATOR

Fig. 7 shows the method used to combine the walking pattern generator with the motion planner. First, by using the method described in Section III, we generate the walking motion of a humanoid robot. While generating the walking pattern, we check whether or not unnecessary collisions occur. If a collision occurs, the motion planner plans a collision-free path for the robot's configuration. The output of this planner is the trajectories of the all joint angles. After planning the collision-free motion, there may be a difference between the actual and reference CWS. If this difference is large, we generate the walking motion again by using the output of the motion planner.

We summarize the proposed motion planner for walking pattern generation as follows.

Algorithm 5: Combination of Motion Planner and Walking Pattern Generator

- 1) Generate the walking motion by using the method described in Section III.

- 2) If undesired collisions occur while generating the walking motion, plan the trajectory of robot's configuration q .
- 3) If the difference between the actual and the reference CWSs is large due to the angular momentum of the robot, record the planned collision-avoiding motion of the robot and go to Step 1. Otherwise, terminate the algorithm.

In Step 3, the collision-free motion is recorded and is used for the walking pattern generator. Here, our walking pattern generator generates the walking motion iteratively [19]. The recorded collision-free motion is used as the initial trajectory used for the walking pattern generator.

With regard to the motion of the humanoid robot walking through the gate, as shown in Section VIII, the balance of the robot can be maintained after the first execution of Step 2 of Algorithm 5, and there is no need to execute Step 1 again. However, we cannot always guarantee that the collision-free motion is found by using this algorithm. The limitations of this algorithm are discussed in Section VII.

VII. DISCUSSION

A. Implementation

We first discuss the implementation of the constraint condition mentioned in Steps 2 and 3 of Algorithm 2 and Step 3 of Algorithm 4. In our implementation, $f(q, t)$ includes the difference between the current and the desired values of 1) the horizontal position of the COG (x_G, y_G), 2) the position/orientation of the feet ($p_{Fi}/\phi_{Fi}, i = 1, 2$) with respect to the global coordinate system, 3) the position/orientation of the hands (${}^C p_{Hi}/{}^C \phi_{Hi}, i = 1, 2$) with respect to the chest coordinate system, and 4) the neck joint angles (θ_N). We set

$$\xi = [\xi_G^T \hat{\xi}^T]^T \quad (8)$$

$$\xi_G = [x_G \ y_G]^T \quad (9)$$

$$\hat{\xi} = [p_{F1}^T \ \phi_{F1}^T \ p_{F2}^T \ \phi_{F2}^T \ p_{H1}^T \ \phi_{H1}^T \ p_{H2}^T \ \phi_{H2}^T \ \theta_N^T]^T. \quad (10)$$

Here, the desired values of ξ_G and $\hat{\xi}$ are denoted by ξ_{Gd} and $\hat{\xi}_d$, respectively, and they are recorded when executing the walking pattern generator.

The forward kinematics in Step 2 of Algorithm 2 calculates ξ for the randomly sampled configuration q . On the other hand, the inverse kinematics calculates the configuration q for given ξ . By using momentum control [1], we can calculate this inverse kinematics. However, this research uses an easy method. In Step 3 of Algorithm 2, we first replace $\hat{\xi}$ by $\hat{\xi}_d$. Then, we iterate the following two steps for a few times: 1) by solving the inverse kinematics of the individual arms and legs, we can obtain the horizontal position of the CoG, ξ_G ; and 2) we compensate the difference between ξ_G and ξ_{Gd} by using the horizontal position of the waist, x_B and y_B . In our simulation and experiment, we iterated this operation two times and we confirmed that the difference between the desired and the actual CoG position is less than 1 cm for the humanoid robot HRP-2.

The humanoid robot HRP-2 used in this research has two 6-DOF arms and two 6-DOF legs. Hence, when solving the inverse kinematics, we do not need to consider the redundancy.

Since this redundancy is effective in avoiding the collision of the links, consideration of the redundancy is considered to be our future research topic.

B. Limitations of Proposed Method

1) *Type of Obstacle*: Our proposed algorithm does not guarantee that the robot can avoid collisions with any type of obstacles. For example, the trajectory of the feet is fixed in our proposed algorithm. This implies that it would be almost impossible to avoid collisions between the lower body and an obstacle. In addition, we assume that obstacle does not move during the motion of the robot.

2) *Determination of Time Parameter*: As shown in Fig. 4, we discretize the time between t_0 and t_1 . Let

$$\delta t = t_2 - t_1 = \dots = t_n - t_{n-1}.$$

Let δt_{st} and δt_{ed} denote the time between t_0 and the beginning of a collision and between t_n and the end of a collision, respectively. In our proposed algorithm, δt , δt_{st} , and δt_{ed} are determined manually.

If δt is small, it may take time to plan the collision-free motion. On the other hand, if δt is large, the planner may not find a collision-free path. In addition, if δt_{st} and δt_{ed} are small, the collision-avoidance motion of the robot may become very rapid. This can be avoided by setting larger values.

In addition, if the robot cannot maintain its balance due to a large angular momentum during a collision-avoidance motion, we can make the motion time longer so as to reduce the effect of the angular momentum [7]. We intend to plan the motion time automatically in future research.

3) *Approximation of Inverted Pendulum*: In our algorithm, we consider making the horizontal position of the CoG track the desired ones. If the dynamics of the humanoid robot can be well approximated by using an inverted pendulum [1], [15], [16], and [18], the robot can maintain its balance since the contact wrench between the feet and the ground tracks the desired one. In many cases, the dynamics of a biped-walking robot can be well approximated by using an inverted pendulum [36]. On the other hand, the effect of the angular momentum may become large in some cases. We have numerically confirmed that we can reduce the effect of angular momentum by performing iterative calculations using our proposed algorithm as described in the next section.

VIII. SIMULATION AND EXPERIMENT

We first show some numerical examples by using the humanoid robot HRP-2. HRP-2 has 30 joints, and its height and weight are $h = 1.54$ m and 56 kg, respectively. We used OpenHRP to generate the biped-walking pattern combined with the motion planner motion planning kit (MPK) [37].

Table I shows the calculation time of the proposed algorithm using a 2.4 GHz PC. While our walking pattern generator [19] iteratively calculates the biped gait, this calculation was performed iteratively five times when measuring the calculation time. In addition, we performed the shortcut operation 150 times. After shortcutting the path, we smoothen the path by

TABLE I
CALCULATION TIME

Trial #	1	2	3	4	5
Walking pattern generate [s]	45	45	45	45	45
Motion plan [s]	14	48	19	38	15
Shortcut [s]	25	41	29	39	41
Spline Interpolation [s]	54	53	52	53	54

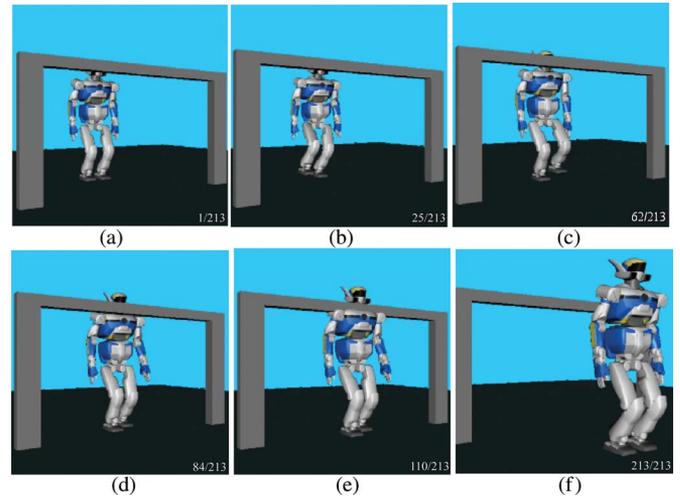


Fig. 8. Original motion of the robot generated by the walking pattern generator.

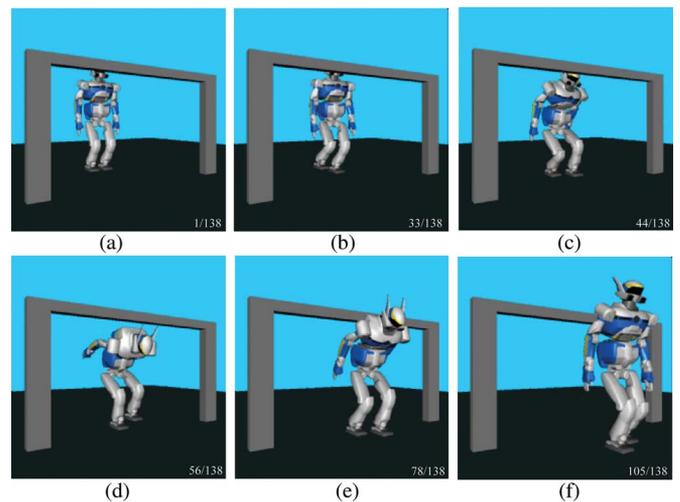


Fig. 9. Collision-free motion of the robot.

using the spline interpolation and then terminated the algorithm. We used fourth order b-spline function to smoothen the path. As shown in this table, the robot's motion of approximately 15 s is calculated within 3 min.

We planned the motion of a humanoid robot walking through a gate. Fig. 8 shows the output of the walking pattern generator. As we can see from Fig. 8(d) and (e), a collision occurs between the gate and the robot. Then, between 5 s before and 3.5 s after the collision, we plan the collision-free motion of the robot. The result of motion planning is shown in Fig. 9; the robot avoids the collision between itself and the gate.

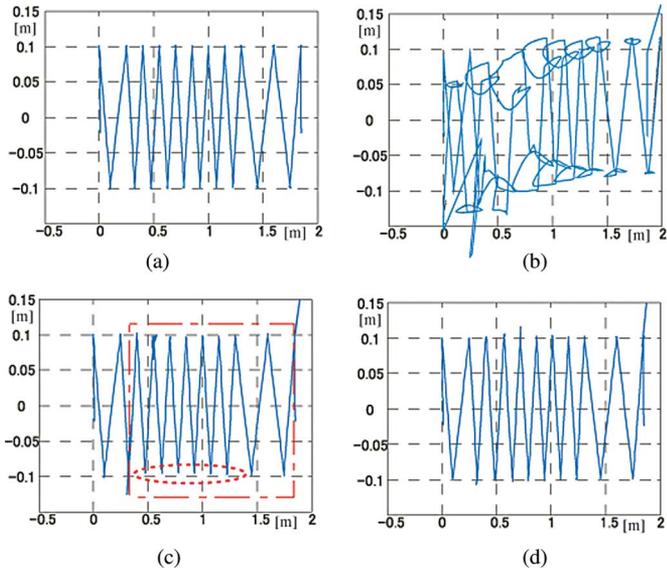


Fig. 10. Simulation result of the ZMP trajectory. (a) Output of the walking pattern generator. (b) Output of the shortcut operation. (c) Output of the spline interpolator. (d) Compensation of the ZMP error caused in the motion planner.

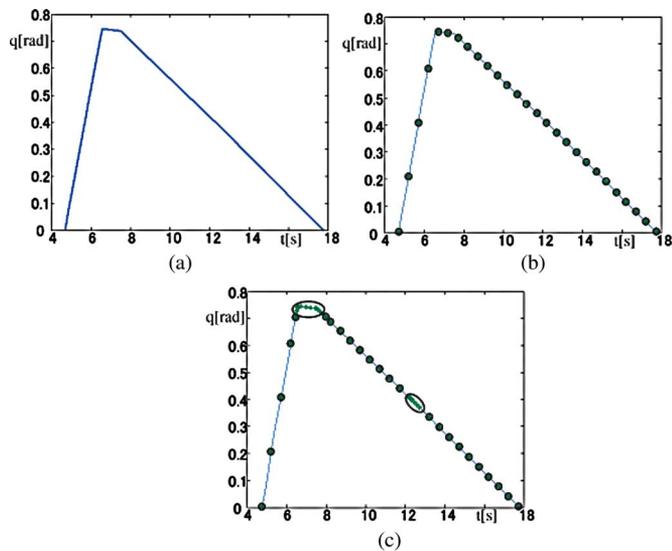


Fig. 11. Time trajectory of the waist joint. (a) Planned trajectory (joint angle of waist). (b) Initial node assignment. (c) Interpolation using b-spline function.

Fig. 10 shows the ZMP trajectories during the simulation. We set the reference CWS such that the horizontal ZMP position coincides with the rotation center of the ankle joint during the single-support phase. Fig. 10(a) shows the output of the walking pattern generator. As shown in this figure, the desired CWS is realized. On the other hand, Fig. 10(b) shows the ZMP trajectory after performing the shortcut operation. The discontinuous velocity led to a large error in the ZMP. Fig. 10(c) shows the ZMP trajectory after the spline interpolation. A slight deviation in the ZMP trajectory remains, as indicated by the dotted circle, although it is not significant. In addition, the dashed rectangle shows the region in which the collision-free motion is planned. Furthermore, as shown in Fig. 10(d), we generated the walk-

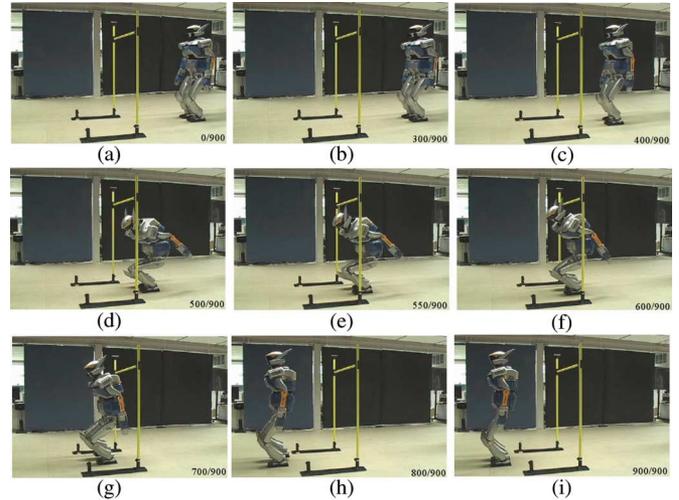


Fig. 12. Experimental result.

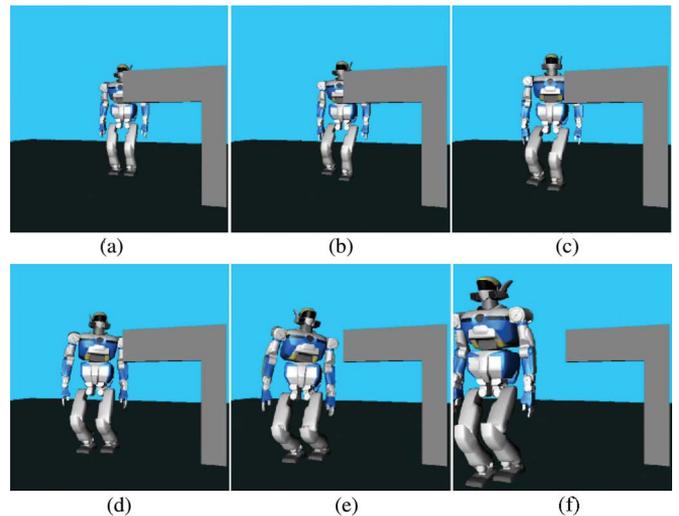


Fig. 13. Motion of robot avoiding the obstacle to its side.

ing pattern again by using the output of the collision-avoiding motion planner. As shown in this figure, the ZMP deviation becomes smaller than that in (c).

Fig. 11 shows how the spline interpolation works. Fig. 11(a) shows the time trajectory of the joint at the waist after the shortcut operation. Fig. 11(b) shows the initial node assignment of the spline interpolation. After performing the algorithm described in Section IV-D, the nodes of the spline interpolation are shown in Fig. 11(c).

Fig. 12 shows the experimental result. This is the experiment corresponding to the case shown in Fig. 10(c). As shown in this result, the humanoid robot walks through the gate while maintaining its balance.

Figs. 13 and 14 show the result of another numerical example. In Fig. 13, the humanoid robot HRP-2 avoids an obstacle placed to its side. In Fig. 14, HRP-2 passes through a narrow space.

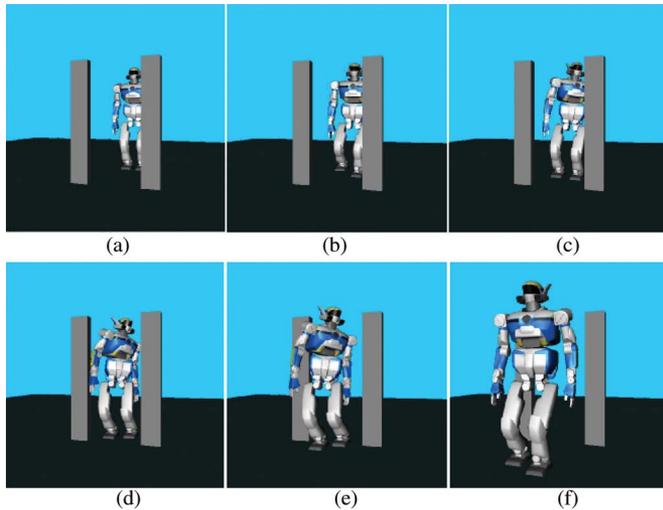


Fig. 14. Motion of robot passing through a narrow space.

IX. CONCLUSION AND FUTURE WORK

In this paper, we discussed collision avoidance while generating the biped gait pattern of a humanoid robot. In our motion planner, we first monitor the collision of the links while running the walking pattern generator by using the CWS. Then, we performed collision-avoiding motion planning. Our planner can consider the walking pattern generator as a time-varying constraint condition. In addition, we plan the motion such that the robot can maintain its balance. The simulation and experimental results show that we can avoid collisions during the walking motion. The results also show that we can plan the collision-free motion within a reasonable time.

We have planned many future works. First, since we explicitly specify both the start and end times of the motion, there is a case in which the obstacle avoidance motion of the robot becomes very quick. For such cases, we currently have to manually adjust both the start and end times of the motion. We consider planning the start and end times of the planner.

In addition, we did not consider the limit of joint torque when planning the collision-free motion. We intend to consider the joint torque limit when planning collision-free motion is also considered in future research.

With regard to the planner, it is necessary for $t_i < t_j$ to be satisfied when connecting two trees as described in Algorithm 3. We intend to consider the case in which this condition is not satisfied in future research.

REFERENCES

- [1] 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 *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2003, pp. 1644–1650.
- [2] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566–580, Aug. 1996.
- [3] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2000, pp. 995–1001.
- [4] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2006.
- [5] A. Franchi, L. Freda, G. Oriolo, and M. Vendittelli, "The sensor-based random graph method for cooperative robot exploration," *IEEE/ASME Trans. Mechatronics*, vol. 14, no. 2, pp. 163–175, Apr. 2009.
- [6] G. Liu, J. Milgram, A. Dhanik, and J. C. Latombe, "On the inverse kinematics of a fragment of protein backbone," in *Proc. 10th Int. Symp. Adv. Robot Kinematics*, 2006, pp. 201–208.
- [7] J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, "Motion planning for humanoid robots under obstacle and dynamic balance constraints," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2001, pp. 692–698.
- [8] J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, "Motion planning for humanoid robots," in *Proc. Int. Symp. Robot. Res.*, 2003, pp. 4420–4426.
- [9] E. Yoshida, "Humanoid motion planning using multi-level DOF exploitation based on randomized method," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2005, pp. 25–30.
- [10] K. Hauser, T. Bretl, and J.-C. Latombe, "Non-gaited humanoid locomotion planning," in *Proc. IEEE Int. Conf. Humanoid Robots*, 2005, pp. 7–12.
- [11] K. Hauser, T. Bretl, K. Harada, and J.-C. Latombe, "Using motion primitives in probabilistic sample-based planning for humanoid robots," in *Proc. Workshop Algorithmic Found. Robot. (WAFR)*, 2006, pp. 507–522.
- [12] K. Harada, S. Hattori, H. Hirukawa, M. Morisawa, S. Kajita, and E. Yoshida, "Motion planning for walking pattern generation of humanoid robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2007, pp. 4227–4233.
- [13] A. Takamishi, H. Lim, M. Tsuda, and I. Kato, "Realization of dynamic biped walking stabilized by trunk motion on a sagittally uneven surface," in *Proc. IEEE Int. Workshop Intell. Robots Syst. (IROS 1990)*, pp. 323–330.
- [14] S. Kagami, K. Nishiwaki, T. Kitagawa, T. Sugihara, M. Inaba, and H. Inoue, "A fast generation method of a dynamically stable humanoid robot trajectory with enhanced ZMP constraint," in *Proc. IEEE Int. Conf. Humanoid Robot.*, 2000, pp. 190–195.
- [15] 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 *Proc. IEEE Int. Conf. Robot. Autom.*, 2003, pp. 1620–1626.
- [16] K. Nagasaka, Y. Kuroki, S. Suzuki, Y. Itoh, and J. Yamaguchi, "Integrated motion control for walking, jumping and running on a small bipedal entertainment robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2004, pp. 3189–3194.
- [17] K. Harada, S. Kajita, K. Kaneko, and H. Hirukawa, "An analytical method on real-time gait planning for a humanoid robot," in *Proc. IEEE-RAS/RSJ Int. Conf. Humanoid Robot.*, 2004, pp. 640–655.
- [18] T. Sugihara and Y. Nakamura, "A fast online gait planning with boundary condition relaxation for humanoid robots," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA 2005)*, pp. 306–311.
- [19] H. Hirukawa, S. Hattori, S. Kajita, K. Harada, K. Kaneko, F. Kanehiro, M. Morisawa, and S. Nakaoka, "A pattern generator of humanoid robots walking on a rough terrain," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 2181–2187.
- [20] H. Hirukawa, S. Hattori, K. Harada, S. Kajita, K. Kaneko, F. Kanehiro, K. Fujiwara, and M. Morisawa, "A universal stability criterion of the foot contact of legged robots -adios ZMP," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2006, pp. 1976–1983.
- [21] L. Sentis and O. Khatib, "A whole-body control framework for humanoids operating in human environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2006, pp. 2641–2648.
- [22] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.*, vol. 5, no. 1, pp. 90–98, 1986.
- [23] O. Stasse, A. Escande, N. Mansard, S. Miossec, P. Evrard, and A. Kheddar, "Real-time (self)-collision avoidance task on a HRP-2 humanoid robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2008, pp. 3200–3205.
- [24] E. Yoshida, I. Belousov, C. Esteves, and J.-P. Laumond, "Humanoid motion planning for dynamic tasks," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2005, pp. 1–6.
- [25] G. Sanchez and J. C. Latombe, "A single-query Bidirectional probabilistic roadmap planner with lazy collision checking," in *Proc. Int. Symp. Robot. Res. (ISRR 2001)*, pp. 403–417.
- [26] S. M. LaValle, J. H. Yakey, and L. E. Kavraki, "A probabilistic roadmap approach for systems with closed kinematic chains," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1999, pp. 1671–1676.
- [27] J. Cortes, T. Simeon, and J. P. Laumond, "A random loop generator for planning the motions of closed kinematic chains using PRM methods," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2002, pp. 2141–2146.

- [28] R. Kindel, D. Hsu, J. C. Latombe, and S. Rock, "Kinodynamic motion planning amidst moving obstacles," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2000, pp. 537–543.
- [29] S. Kagami, F. Kanehiro, Y. Tamiya, M. Inaba, and H. Inoue, "Autobalancer: An online dynamic balance compensation scheme for humanoid robots," presented at the Int. Workshop Alg. Found Robot. (WAFR), Tokyo, Japan, 2000, 2010.
- [30] H. Sanada, E. Yoshida, and K. Yokoi, "Passing Under Obstacles with Humanoid Robots," in *Proc. of IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2007, pp. 4028–4034.
- [31] J. Chestnutt, M. Lau, J. J. Kuffner, G. Cheung, J. Hodgins, and T. Kanade, "Footstep planning for the ASIMO humanoid robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2005, pp. 3618–3623.
- [32] M. Stilman, K. Nishiwaki, S. Kagami, and J. J. Kuffner, "Planning and executing navigation among movable obstacles," *Adv. Robot.*, vol. 21, pp. 1617–1634, 2007.
- [33] K. Harada, S. Kajita, F. Kanehiro, K. Fujiwara, K. Kaneko, K. Yokoi, and H. Hirukawa, "Real-time planning of humanoid robot's gait for force controlled manipulation," *IEEE/ASME Trans. Mechatronics*, vol. 12, no. 1, pp. 53–62, Feb. 2007.
- [34] F. Kanehiro, W. Suleiman, F. Lamiroux, E. Yoshida, and J.-P. Laumond, "Integrating dynamics into motion planning for humanoid robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2008, pp. 660–667.
- [35] K. Harada, M. Morisawa, K. Miura, S. Nakaoka, K. Fujiwara, K. Kaneko, and S. Kajita, "Kinodynamic gait planning for full-body humanoid robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2008, pp. 1544–1550.
- [36] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3D linear inverted pendulum mode: A simple modeling for a biped walking pattern generation," in *Proc. Int. Conf. Intell. Robots Syst.*, 2001, pp. 239–246.
- [37] Motion Planning Kit. [Online]. Available: <http://robotics.stanford.edu/mitul/mpk/>



Kensuke Harada (M'98) received the M.E. and Dr. E. degrees from the Graduate School of Mechanical Engineering, Kyoto University, Kyoto, Japan, in 1994 and 1997, respectively.

From 1997 to 2002, he was a Research Associate in Graduate Industrial and Systems Engineering, Hiroshima University, Hiroshima, Japan. From 2005 to 2006, he was a Visiting Scholar in the Computer Science Department, Stanford University, Palo Alto, CA. He is currently a Researcher at the Humanoid Research Group, Intelligent Systems Research Institute, National Institute of Advanced Industrial Science and Technology, Tsukuba.

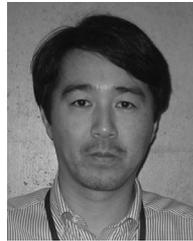
His research interests include the mechanics and control of humanoid robots and robotic hands.

Dr. Harada is a member of the IEEE Robotics and Automation Society.



Shizuko Hattori received the Graduate degree from the University of Electro Communications, Tokyo, Japan, in 1987.

She joined the Mechanical Engineering Laboratory, Hitachi Ltd., Hitachi, Japan, in 1987. Since 2004, she has been a Member of the Technical Staff at the Humanoid Research Group, Intelligent Systems Research Institute, National Institute of Advanced Industrial Science and Technology, Tsukuba, Japan.



Hirohisa Hirukawa (M'91) received the Ph.D. degree from Kobe University, Kobe, Japan, in 1987.

He joined the Electrotechnical Laboratory, National Institute of Advanced Industrial Science and Technology, Ministry of International Trade and Industry, Tsukuba, Japan, in 1987. From 1994 to 1995, he was a Visiting Scholar at Stanford University, Palo Alto, CA. Since 2001, he has been serving as the Leader of the Humanoid Robotics Group, Intelligent Systems Research Institute, National Institute of Advanced Industrial Science and Technology, which developed the humanoid robot HRP-2 and the software platform for humanoid robotics OpenHRP. HRP-2 was the first human-size humanoid robot that can lie down and get up, and was used to examine several applications of humanoid robots. His current research interests include humanoid robotics and motion planning.

Dr. Hirukawa is a member of the IEEE Robotics and Automation Society.



Mitsuharu Morisawa received the Ph.D. degree in integrated design engineering from Keio University, Tokyo, Japan, in 2004.

He was a Visiting Scholar at the Laboratoire d'Analyse et d'Architecture des Systèmes, Centre National de la Recherche Scientifique, Toulouse, France, in 2009. He has been with the Humanoid Research Group, Intelligent Systems Research Institute, National Institute of Advanced Industrial Science and Technology, Tsukuba, Japan, since 2004. His research interests include humanoid robotics, parallel mechanisms, and motion control.

Dr. Morisawa is a member of the Robotics Society of Japan and the IEEE Robotics and Automation Society.



Shuuji Kajita (M'91) received the M.E. and Dr.E. degrees in control engineering from Tokyo Institute of Technology, Tokyo, Japan, in 1985 and 1996, respectively.

In 1985, he joined the Mechanical Engineering Laboratory, Ministry of International Trade and Industry (MITI), Tsukuba, Japan. From 1996 to 1997, he was a Visiting Researcher at the California Institute of Technology, Pasadena. He is currently a Senior Researcher with the Humanoid Research Group, Intelligent Systems Research Institute, National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, which was reorganized from AIST-MITI in April 2001. His research interests include robotics and control theory.

Dr. Kajita is a member of the Society of Instrument and Control Engineers, Robotics Society of Japan, and IEEE Robotics and Automation Society.



Eiichi Yoshida (S'94–M'96) received the M.E. and Ph.D. degrees in precision machinery engineering from the Graduate School of Engineering, The University of Tokyo, Tokyo, Japan, in 1993 and 1996, respectively.

From 1990 to 1991, he was a Visiting Research Associate at the Ecole Polytechnique Fédérale de Lausanne, Swiss Federal Institute of Technology, Lausanne, Switzerland. From 2004 to 2008, he was a Codirector of the AIST/IS-CNRS/ST21 Joint French-Japanese Robotics Laboratory, Laboratoire d'Analyse et d'Architecture des Systèmes, Centre National de la Recherche Scientifique, Toulouse, France. He is currently the Codirector of the Humanoid Research Group, Intelligent Systems Research Institute, National Institute of Advanced Industrial Science and Technology, Tsukuba, Japan. His research interests include task and motion planning, modular robotic systems, and humanoid robots.

Dr. Yoshida is a member of the IEEE Robotics and Automation Society.