2008 IEEE/RSJ International Conference on Intelligent Robots and Systems
Acropolis Convention Center
Nice, France, Sept, 22-26, 2008

Kinodynamic Gait Planning for Full-Body Humanoid Robots

Kensuke Harada, Mitsuharu Morisawa, Kanako Miura,
Shin-ichiro Nakaoka, Kiyoshi Fujiwara, Kenji Kaneko, and Shuuji Kajita

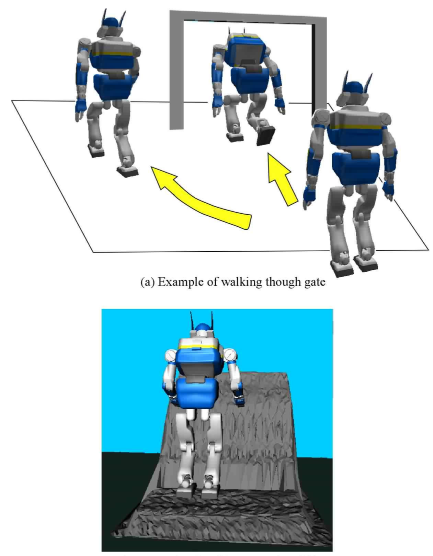
Abstract— **This paper proposes the kinodynamic gait planning for humanoid robots where both kinematics and dynamics of the system are considered. We can simultaneously plan both the foot-place and the whole-body motion taking the dynamical balance of the robot into consideration. As a dynamic constraint, we consider the differential equation of the robot's CoG. To solve this constraint, we assume two walking pattern generators; the offline and the online ones. We randomly sample the configuration space to search for the path connecting the start and the goal configurations. When sampling the configuration space, three milestones are sequentially connected to the parent milestone. To show the effectiveness of the proposed methods, we show simulation and experimental results where the humanoid robot HRP-2 walks on several environments.**

I. Introduction

The ultimate goal of the humanoid research is to realize a humanoid robot that can work instead of a human in our daily environment. To realize this goal, a humanoid robot has to recognize the working environment and plan the motion pattern depending on the environment. However, the motion planning of a humanoid robot is difficult due to several reasons; First, a humanoid robot has to keep balance with keeping contact between the foot and the environment. Second, a humanoid robot has to avoid unnecessary contact among the links and between a link and the environment. And thirdly, a humanoid robot has to select the foot placement so that the humanoid robot can finally reach the goal configuration. This research focuses on the humanoid motion planning that can take all these functions into account at the same time.

Let us focus on some typical examples where our method becomes effective. Fig.1 (a) shows an example where a humanoid robot walks from the start to the goal position. Here, a gate is placed between the start and the goal position. If the humanoid robot can walk through the gate, the humanoid robot can reach the goal with minimum walking distance. However, if the height of the beam is low and it is difficult for the humanoid robot to walk through, the humanoid robot has to change the walking direction so as to avoid the gate. Fig.1 (b) shows another example where a humanoid robot walks on the rocky cliff. In this case, the humanoid robot has to select the foot placement so that the robot can keep balance easily. Also, if it is difficult to keep balance by simply standing on two legs, the humanoid robot will touch the hand to the environment. To realize these functions, this paper proposes a method simultaneously planning the whole body motion and the foot placement of a humanoid robot with keeping its dynamical balance.

The authors are with Intelligent Systems Research Institute, National Institute of Advanced Industrial Science and Technology, 1-1-1 Umezono, Tsukuba 305-8568, Japan kensuke.harada@aist.go.jp



(a) Example of walking though gate

(b) Example of walking on rocky cliff

Fig. 1. Two examples of humanoid motion planning

For generating the walking pattern of a humanoid robot with keeping the dynamical balance, the ZMP based methods have been used. For given ZMP trajectory, the trajectory of the robot's CoG (center of gravity) is calculated by solving the ordinary differential equation. Recently, the research has been done on the online walking pattern generation [2], [3], [4], [8], [9]. In these researches, to adaptively change the foot placement according to the sensor information, the new trajectory of the CoG is calculated by solving the ordinary differential equation and is smoothly connected to the currently executing trajectory. We will show in this paper that this technique can be effectively installed in the planning method.

Now, let us consider simultaneously planning the whole body motion and the foot step placement with keeping the dynamical balance. As for the motion planning of humanoid robots, the random sampling based method such as PRM and RRT has been researched [10], [11], [14], [6], [7]. Recently, the randomized kinodynamic planning was proposed [17], [18] where it can be used for the motion planning taking the dynamics of the system into consideration. Here, the roadmap of randomly-sampled milestones are connected by using the solution of the differential equation. In this paper, we realize this simultaneous planning by assuming that each milestone of the planner is composed of the whole body joint angles and position of the foot step. Then, each milestone is connected by using the solution of the differential equation. Here, different from most of the kinodynamic planners, our proposed planner calculates the trajectory of the system composed of a few milestones when sampling the milestone. This is because, when generating the walking motion in realtime, we calculate the walking pattern composed of a few steps and use just the first one when executing the calculated trajectory [3], [4], [9].

Based on this consideration, this paper proposes a method for simultaneous planning the whole body motion and the foot step placement with keeping the dynamical balance of the robot. To realize this planner, we effectively combine the online walking pattern generator and the randomized kinodynamic planner. This paper is composed as follows; After showing the related researches in Section 2, we show the planning method in Section 3.

II. Related Works

Walking Pattern Generation of Humanoid Robots:

As for the walking pattern generation of humanoid robots, many researchers have used the ZMP. Takanishi et al.[1] proposed a method to calculate the trunk motion by using the FFT. Recently, the online (real-time) walking pattern generator has been researched. Kagami et al.[9] proposed an approach numerically solving the differential equation of the trunk motion. Kajita et al.[2] used the preview control. Harada et al. [3], Morisawa et al.[4] and Sugihara et al.[8] proposed the analytical solution based approaches.

Motion Planning with Collision Avoidance:

As for the motion planning based on the random sampling, Kavraki et al.[21] proposed the probabilistic roadmap (PRM) planner. Kuffner et al.[16] and Sanchez et al.[19] later proposed the single-query and bi-directional method. Lazy collision checking was proposed in [20], [19]. The random sampling method was extended to the mechanical system with closed kinematic chain[17] and kinodynamic planner taking the velocity of the system into consideration [17], [18].

Recently, some researchers research the motion planning of a humanoid robot. Kuffner et al.[10], [11] first generate the collision free motion satisfying the statical balance and then transformed it to the dynamically stable motion. Chestnutt et al.[12] proposed a heuristic method of footstep planning. Stilman et al.[13] proposed the manipulation planning of a movable object. Yoshida [14] approximated the shape of the robot by a rectangular and extended the approach for a omni-directional vehicle. Harada et al.[22] proposed a random sampling based planner combining with the walking pattern generator. Recently, there are a couple of researches simultaneously planned the foot step position and the whole body motion with keeping the statical balance. Hauser et al. [6] proposed the multi-step planning method applicable to the rock-climbing humanoid robot. Sanada et al. [5] proposed a RRT based method used to avoid an obstacle.

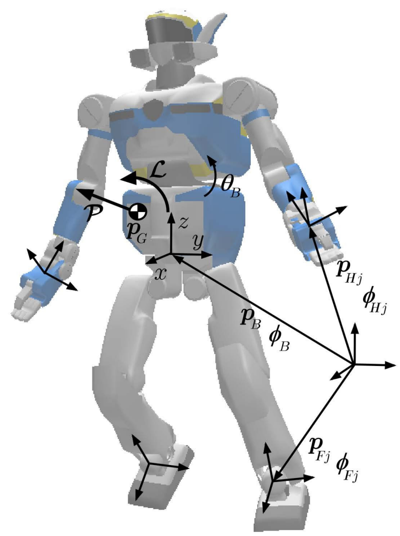


Fig. 2. Model of HRP2 Humanoid Robot

However, there has been no research on humanoid motion planning where the foot-step position and the whole-body motion are simultaneously planned with keeping the dynamical balance of the robot.

III. Definitions

A. Configuration Space

Fig. 2 shows the model of a humanoid robot. Let $\boldsymbol{\xi}_*$ be the six dimensional vector composed of the position $\boldsymbol{p}_*$ and the orientation $\boldsymbol{\phi}_*$ of the coordinate frame fixed to a link of the robot. The subscripts Fj, Hj $(j = r, l)$, B and G denote the feet, the hands, the waist and the CoG, respectively.

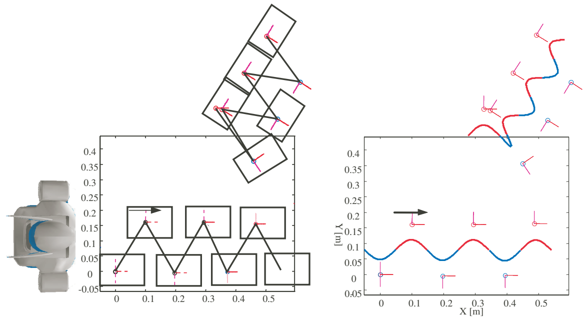
We assume that the 3D models of the robot and the environment are known. These models are used for collision checking. A configuration of a humanoid robot is composed of the position/orientation of the waist ($\boldsymbol{\xi}_B$) and all joint angles ($\boldsymbol{\theta}$). Let us consider the following coordinate transformation:

$$(\boldsymbol{p}_G, \boldsymbol{\phi}_B, \boldsymbol{\xi}_{Fr}, \boldsymbol{\xi}_{Fl}, \boldsymbol{\xi}_{Hr}, \boldsymbol{\xi}_{Hl}, \boldsymbol{\theta}_r) = \boldsymbol{f}(\boldsymbol{\xi}_B, \boldsymbol{\theta}), \qquad (1)$$

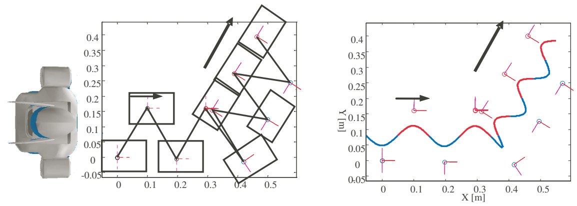
where $\boldsymbol{\theta}_r$ denotes the joint angles of the chest, the neck and the finger. Among the coordinate variables defined in eq.(1), we specify the desired trajectory for the position/orientation of the hands. The hand trajectory is expressed w.r.t. either the inertial coordinate or the chest coordinate systems. Hence, we consider the following configuration space when planning the motion of a humanoid robot:

$$\boldsymbol{q} = (\boldsymbol{p}_G, \boldsymbol{\phi}_B, \boldsymbol{\xi}_{Fr}, \boldsymbol{\xi}_{Fl}, \boldsymbol{\theta}_r) \in \mathcal{C}, \qquad (2)$$

where $\mathcal{C}$ denotes the configuration space of the robot and $\mathcal{C}_{free} \subset \mathcal{C}$ its free space. We plan the motion of the robot based on eq.(2). Our proposed planner can be applied to general humanoid robots where its configuration has the form of eq.(2) and where each arm/leg has at least 6DOF.



(a) Calculation of New Trajectory (Left: ZMP, Right: CoG)



(b) Connection of New Trajectory to Current One (Left: ZMP, Right: CoG)

Fig. 3. Overview of online walking pattern generation

B. Gait Pattern Generation

Among the variables defined in eq.(2), the horizontal position of the CoG has to be determined so as to satisfy the dynamical balance of the robot. Let $\boldsymbol{p}_G = (x_G, y_G, z_G)$ and $\boldsymbol{p}_Z = (x_Z, y_Z, z_Z)$ be the position of the CoG and ZMP, respectively. Also, let $x-y$ plane coincide with the horizontal plane. The following differential constraints are imposed on the system:

$$\ddot{x}_G = \frac{g}{z_G - z_Z}(x_G - x_Z), \tag{3}$$

$$\ddot{y}_G = \frac{g}{z_G - z_Z}(y_G - y_Z). \tag{4}$$

In eqs.(3) and (4), the dynamical balance of the robot can be kept if the ZMP is strictly inside the supporting polygon.

By using Fig. 3, we show the overview of the online walking pattern generation [9], [3], [4]. As shown in Fig. 3(a), given the ZMP trajectory for a humanoid robot walking straightforward, we first calculate the the trajectory of the horizontal CoG by using eqs.(3) and (4). Then, if we want the robot to change the walking direction to the left, we calculate the trajectory of the horizontal CoG by using the ZMP trajectory turning left. As shown in Fig. 3(b), the change of the walking direction can be realized by smoothly connecting the new CoG trajectory to the current executing one. Let $\boldsymbol{q} = (x_G, y_G, \hat{\boldsymbol{q}})$, we can classify the walking pattern generation methods into three classes:

1) Offline walking pattern generator: Assume that the start and the goal configurations are given. Also, assume that the trajectories of both $\hat{\boldsymbol{q}}$ and $\boldsymbol{p}_Z$ from the start to the goal are given. Calculate the trajectory of x_G and y_G from the start to the goal by using eqs.(3) and (4).

2) Online walking pattern generator I [3]: Assume that the configuration at the beginning of the next double-support phase is given. Also, assume that the trajectories of both $\hat{\boldsymbol{q}}$ and $\boldsymbol{p}_Z$ for a few steps from the beginning of the next double-support phase are given. Calculate the trajectory of x_G and y_G for a few steps from the beginning of the next double-support phase by using eqs.(3) and (4).

3) Online walking pattern generator II [4]: Assume that the configuration at the beginning of the next single-support phase is given. Also, assume that the trajectories of both $\hat{\boldsymbol{q}}$ and $\boldsymbol{p}_Z$ for a few steps from the beginning of the next single-support phase are given. Calculate the trajectory of x_G and y_G for a few steps from the beginning of the next double-support phase by using eqs.(3) and (4).

In the above three methods, the path trajectory composed of a few steps is calculated. This is because it is difficult to determine the position/velocity of the CoG if it is split into each step. In case of the motion planning based on the online walking pattern generator proposed in this research, the configuration of just the first step is used as a milestone.

As described in [3], [4], the difficulty of the online methods is the smooth connection of two CoG trajectories. When solving eqs.(3) and (4), we assume the two-point boundary value problem so as to ensure that the robot does not fall down at both the initial and the final configurations. This means that we cannot specify the initial velocity of the CoG and that the discontinuity of the velocity may occur when connecting two CoG trajectories. We overcame this problem by simultaneously planning the CoG and the ZMP trajectories including the configuration of the connecting point[3], [4]. We also note that, the Online method II is more difficult than the Online method I [4]. This is because the ZMP trajectory in the single support phase is planned where the support polygon of the single support phase is smaller than that of the double support phase. However, by using the Online method II, the quick change of gait can be realized.

IV. Single-query, Single-directional Planning

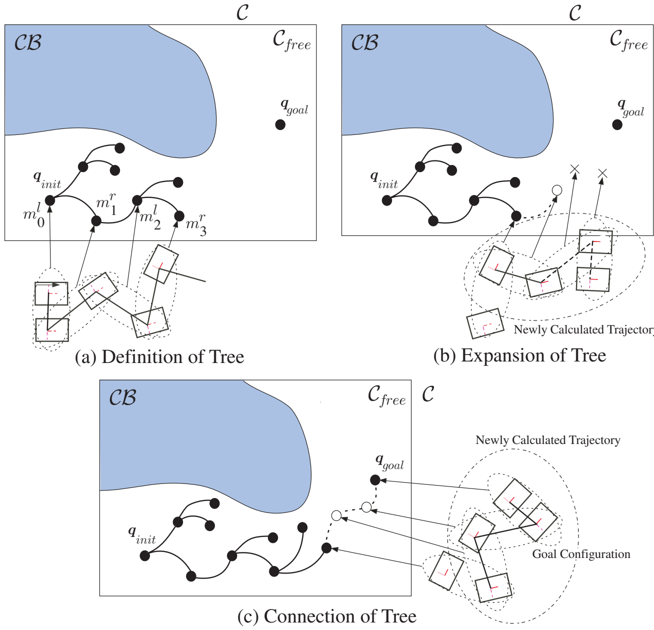
This section explains how to plan the humanoid robot's motion with keeping the dynamical balance.

A. Overall Algorithm

Fig.4 shows the overview of our planning algorithm. Given the start ($\boldsymbol{q}_{init}$) and the goal ($\boldsymbol{q}_{goal}$) configurations, the planner builds a tree T rooted at the start configuration and tries to connect the tree to the goal configuration. If the tree can be connected to the goal, the planner returns the path τ from the start to the goal. This algorithm can be summarized by Algorithm 1.

Algorithm 1 PLANNER($\boldsymbol{q}_{init}, \boldsymbol{q}_{goal}$)

```
Install q_init as the root of T
Repeat s times
    EXPAND-TREE
    τ ← CONNECT-GOAL
    If τ ≠ nil, then return tau
Return failure
```



(a) Definition of Tree

(b) Expansion of Tree

(c) Connection of Tree

Fig. 4. Overview of motion planning method

B. Tree Expansion

Fig.4 (a) shows the definition of the tree of milestones used in our planner. We define each milestone by using the configuration of the robot just before the single-support phase begins. Here, we can define two kinds of milestones depending on the forthcoming single-support phase. For a milestone m, m^r and m^l denote that the forthcoming single-support phase is the right-support and the left-support phases, respectively. Let m_i^r be the i-th milestone with the forthcoming right-support phase. The parent of this milestone can be defined by

$$m_j^l = \mathrm{parent}(m_i^r), \tag{5}$$

where the function $\mathrm{parent}(*)$ is used to define the parent of the milestone $*$. Eq.(5) means that the left-support phase comes after the right-support phase and vise versa.

Overview of the tree expansion algorithm is shown in Fig.4 (b). First, we randomly pick a (parent) milestone from T. Then, by randomly sampling the configuration space, we generate a new (child) milestone. Here, if we picked the parent milestone m_j^r, we make $\boldsymbol{\xi}_{Fr}$ of the child milestone be same as that of the parent. On the other hand, if picking m_j^l, $\boldsymbol{\xi}_{Fl}$ of the child be same as the parent.

When expanding tree, for the child milestone to satisfy eqs. (3) and (4), we consider generating the walking pattern by sequentially connecting three milestones to the parent. Here, the firstly added milestone corresponds to the child milestone and another two milestones will be deleted if the child milestone is collision free. We also define that the robot stops at the lastly added milestone.

Here, depending on the walking-pattern generator explained in the subsection III-B, we explain two methods to generate the child milestone:

1) Tree Expansion using Offline Walking Pattern Generator: Assume that $\boldsymbol{q}_{init}$ and the lastly added configuration are given. Also, assume that the trajectories of both $\hat{\boldsymbol{q}}$ and $\boldsymbol{p}_Z$ between $\boldsymbol{q}_{init}$ and the lastly added configuration are given. By using the offline walking pattern generator, calculate the trajectory x_G and y_G. If the firstly added configuration is collision free, we define this milestone as a child of the parent.

2) Tree Expansion using Online Walking Pattern Generator II: Assume that the parent configuration is given. Also, assume that the trajectories of both $\hat{\boldsymbol{q}}$ and $\boldsymbol{p}_Z$ between the parent and the lastly added configuration are given. By using the online walking pattern generator II, calculate the trajectory x_G and y_G. If the firstly added configuration is collision free, we define this milestone as a child of the parent.

Let us summarize the tree expansion algorithm by Algorithm 2:

Algorithm 2 EXPAND-TREE

Repeat until a new milestone has been generated
    Pick a milestone m from T at random with probability $\pi(m)$
    If $m = m^r$ then, $j = r$
    else then, $j = l$
    Exclude $\boldsymbol{\xi}_{Fj}$ from $\hat{\boldsymbol{q}}$
    For $i = 1, 2, \cdots$ until a new milestone has been generated
        Sample $\hat{\boldsymbol{q}}$ unformly at random from $B(m, \rho/i)$
        CONNECT-THREE-MILESTONES(m)
        (x_G, y_G) trajectory $\leftarrow$ ONLINE-WPG or OFFLINE-WPG
        If $\boldsymbol{q} = (x_G, y_G, \hat{\boldsymbol{q}}, \boldsymbol{\xi}_{Fj})$ of firstly added milestone is collision free then
            Install $\boldsymbol{q}$ in T as a child of m

where $\pi(m)$ and $B(m, \rho/i)$ denote the probability distribution and the subset of radius ρ/i, respectively [19]. The function CONNECT-THREE-MILESTONES(m) will be discussed in the discussion subsection.

C. Connection with Goal

Once a milestone is added to the tree, we check whether or not the tree can be connected with the goal. This method is the straightforward extension of the tree expansion method explained in the previous subsection. Fig.4 (c) shows the overview of the method of connecting tree with the goal.

If the distance between the most recently added milestone and the goal is less than ρ, the connection check can be performed by inserting two milestones between the added milestone and the goal. We define that the robot stops at the goal. This algorithm can be summarized by Algorithm 3:

Algorithm 3 CONNECT-GOAL

$m \leftarrow$ most recently added milestone
If $d(m, m_{goal}) < \rho$ then
    INSERT-TWO-MILESTONES(m, m_{goal})
    (x_G, y_G) trajectory $\leftarrow$ ONLINE-WPG or OFFLINE-WPG
    If two inserted and goal milestones are collision free then
        $\tau \leftarrow$ path connecting $\boldsymbol{q}_{init}$ and $\boldsymbol{q}_{goal}$
        Return TEST-PATH(τ)

where $d(m, m_{goal})$ denotes the distance between m and m_{goal}[19]. The function INSERT-TWO-MILESTONES(m, m_{goal}) will be discussed in the discussion subsection.

D. Lazy Collision Checking

After a candidate path τ connecting the start and goal configurations are generated, we will check the collision of all the segments included in the path by using the function TEST-PATH(τ) appeared in Algorithm 3. This subsection mainly explains how to generate the segment of path connecting the start and the goal configurations. And, how to select a point on the segment follows[19].

Depending on the walking pattern generator explained in the previous section we have to methods to generate the segment of path:

1) Segment Generation using Offline Walking Pattern Generator: Assume that $\boldsymbol{q}_{init}$ and $\boldsymbol{q}_{goal}$ are given. Also, assume that the trajectories of both $\hat{q}$ and $\boldsymbol{p}_Z$ between $\boldsymbol{q}_{init}$ and $\boldsymbol{q}_{goal}$ are given. Calculate the trajectory x_G and y_G by using the offline walking pattern generator.

2) Segment Generation using Online Walking Pattern Generator II: Assume that $\boldsymbol{q}_{init}$ and $\boldsymbol{q}_{goal}$ are given. Also, assume that the trajectories of both $\hat{\boldsymbol{q}}$ and $\boldsymbol{p}_Z$ between $\boldsymbol{q}_{init}$ and q_{goal} are given. Sequentially generate the trajectory of x_G and y_G from $\boldsymbol{q}_{init}$ to $\boldsymbol{q}_{goal}$ by using the online walking pattern generator II.

E. Discussion

1) Step Connection: We first explain the function CONNECT-THREE-STEPS and INSERT-TWO-STEPS appeared in Algorithm 2 and 3, respectively. We have used simular functions for the online walking pattern generator [3]. We can select these functions relatively freely. For example, we can connect more than four milestones and can insert more than two milestones although the calculation time will increase.

Let F_{ij}, $i = l, r, j = 0, \cdots, 3$ be the three dimensional vector composed of the x, y position and yaw orientation of the foot. Fig.5 (a) shows the case where we sample a milestone and connect it to m_j^l. Let $\bar{F}_{r1}$ be the original position of F_{r1}. We selected F_{l2} so that the relative position of F_{l2} w.r.t. $\bar{F}_{l2}$ be same as the relative position of F_{r1} w.r.t. $\bar{F}_{r1}$. Also, Fig.5 (b) shows the case where m_j^r is connected to the goal. In this case, we set $F_{l1} = (F_{l0} + F_{l2})/2$.

Fig.5 also shows how to determine $\hat{\boldsymbol{q}}$. $\hat{\boldsymbol{q}}$ is determined by using the spline interpolation of the values at each milestone. Finally, we determine the z position and $roll$ and $pitch$ orientation of the foot so that the foot can adapt to the terrain.

2) Two Walking Pattern Generators: When expanding the tree by using the offline walking pattern generator, we calculate the motion of the robot from the start to the added milestones. It means that the x_G and y_G of each milestone included in the path between the start and the added milestones change depending on the configuration of the added milestones. This is one of the problems of the offline walking pattern generator used for the kinodynamic gait planning. On the other hand, although such problem will not occur for the online gait pattern generator, the planned ZMP trajectory will shift from the line and sometimes lies out of the support polygon. Hence, when checking the collision of the added milestone, we also have to check whether the ZMP trajectory is included in the support polygon or not.

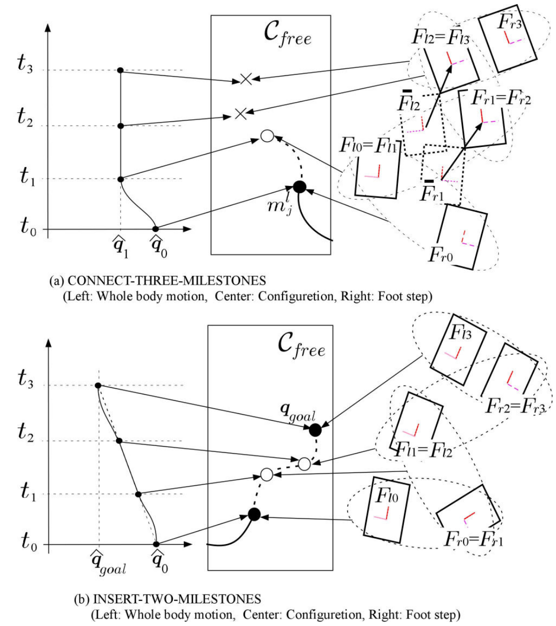


Fig. 5. Functions CONNECT-THREE-STEPS and INSERT-TWO-STEPS

V. Results

We calculated the motion of the humanoid robot HRP-2. HRP-2 has totally 30 joints and its height and weight are $h = 1.54$[m] and 56[kg], respectively. As for the simulation software, we used OpenHRP to generate the walking pattern where it is combined with the motion planner MPK (Motion Planning Kit) developed in Stanford university[23].

First, we show the motion of the robot walking on the rocky cliff. Here, we assume that the hands contact the environment to enhance the stability of the robot. Here, when planning the motion, we did not consider the reaction force of the hands. As for the hand trajectory, we specified its desired trajectory for a single step and connect then when planning the motion. As a walking pattern generator, we used the offline one. As $\hat{\boldsymbol{q}}$, we consider two joint angles (pitch and yaw axes) of the waist. Also, when searching for the foot place, we fixed the yaw angle of the foot. By using the laser range sensor, we captured the shape of the rocky cliff and transformed it to the VRML file by hand. The result of calculation is shown in Fig.6. Here, HRP-2 walks to the right for 0.18[m] with the hands contacts the environment. Although we just specified the start and the goal configurations, the walking motion can automatically be generated. It took 38[s] to calculate the whole motion of this example. After planning the motion, we sometimes encountered the case where the path includes some roundabout ways. Also, sometimes it took relatively long time to obtain the solution

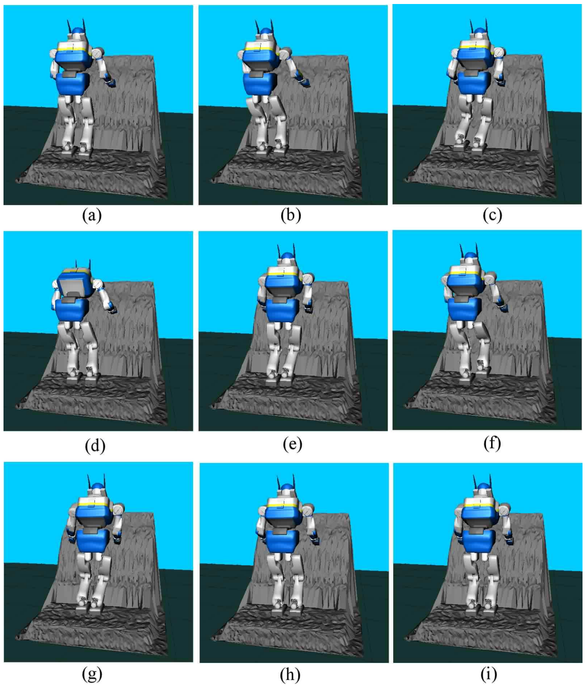


Fig. 6. Walking motion of HRP-2 on rocky cliff

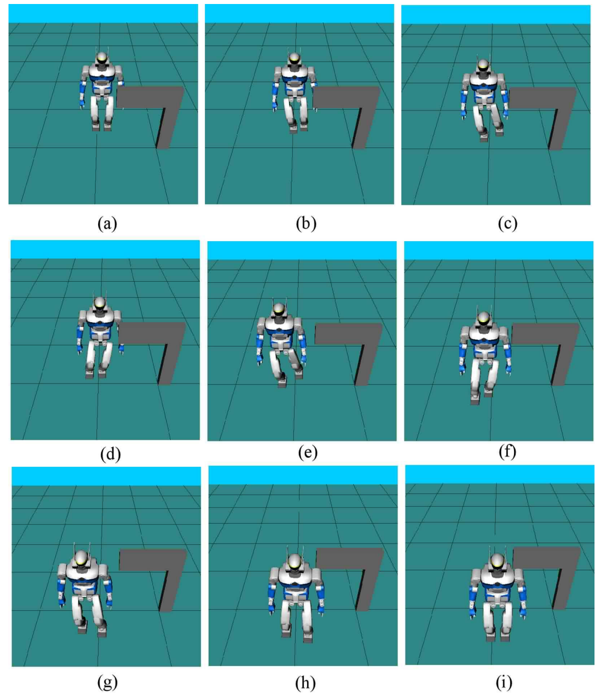


Fig. 7. Walking motion of HRP-2 avoiding obstacle

path. The motion planner including the shortcut operation is our current work.

As a second example, the robot walks with avoiding the obstacle. The result is shown in Fig. 7. As we can see from the figure, the robot avoids the obstacle walking to the right. It took 147[s] to calculate this motion.

Fig.8 shows the experimental result on rocky cliff. Here, we did not succeeded in making the robot walk on the rocky cliff if we plan the motion based on the VRML model captured by the raser range sensor due to the error. Better result was obtained if we used the VRML model made by hand. However, this is a challenging experiment and currently it is difficult for the robot to keep balance after three steps.

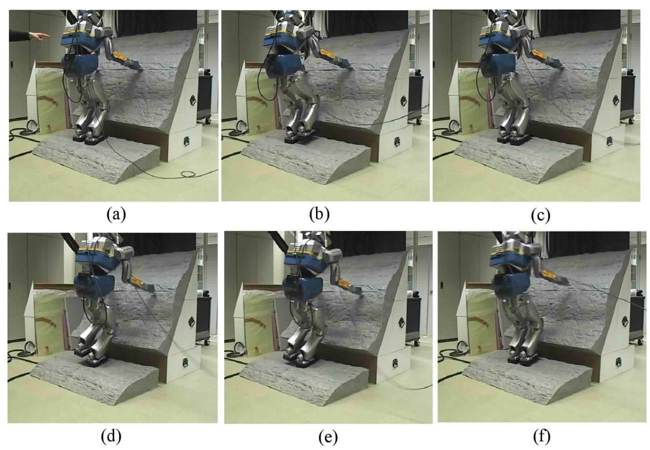


Fig. 8. Experiment of HRP-2 on rocky cliff

VI. Conclusions

This paper proposed the kinodynamic gait planning for humanoid robots where both kinematic and dynamic constraints are imposed. By using our proposed method, the simultaneous planning of foot-place and the whole-body motion can be realized with keeping the dynamical balance of the robot. To search for the path connecting the start and goal configurations, we randomly search for the configuration space by using the walking pattern generator. To show the effectiveness of our proposed method, we first calculated the motion of the humanoid robot HRP-2 walking on the rocky cliff with the hands contact the environment. Then we calculate the motion of HRP-2 walking with avoiding the obstacle.

In this research, the foot trajectory between two milestones was determined based on the walking pattern generator. The extension of our research to the problem of the stepping over obstacles is our future research topic. The numerical examples in more complex environment is also our future research topic.

Finally, the first author would like to express his sincere gratitude for Dr. Kazuhito Yokoi and Dr. Eiichi Yoshida with AIST for their helpful discussions.

References

[1] A. Takanishi, H. Lim, M. Tsuda, and I. Kato, Realization of Dynamic Biped Walking Stabilized by Trunk Motion on a Sagittally Uneven Surface, *Proc. of IEEE International Workshop on Intelligent Robots and Systems (IROS '90)*, pp.323-330, 1990.

[2] 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, *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 1620-1626, 2003.

[3] K. Harada, S. Kajita, K. Kaneko, and H. Hirukawa, An Analytical Method on Real-Time Gait Planning fpr a Humanoid Robot, *Proc. of IEEE-RAS/RSJ Int. Conf. on Humanoid Robotics*, 2004.

[4] M. Morisawa, K. Harada, S. Kajita, S. Nakaoka, K. Fujiwara, F. Kanehiro, K. Kaneko, and H.Hirukawa, Experimentation of Humanoid Walking Allowing Immediate Modification of Foot Place Based on Analytical Solution, *Proc. of 2007 IEEE Int. Conf. on Robotics and Automation*, 2007.

[5] M. Sanada, E. Yoshida, and K. Yokoi, Passing under Obstacles with Humanoid Robots –Step Motion Planning for Sideway Passing under Motion–, 8th Domestic Conference of SI Dept. of SICE, 2007 (In Japanese).

[6] K. Hauser, T. Bretl, J.-C. Latombe, Non-gaited Humanoid Locomotion Planning, *Proc. of IEEE Int. Conf. of Humanoid Robots*, 2005

[7] K. Hauser, T. Bretl, K. Harada, and J.-C. Latombe, Using motion primitives in probabilistic sample-based planning for humanoid robots, *Workshop on Algorithmic Foundations of Robotics (WAFR)*, 2006.

[8] T. Sugihara and Y. Nakamura, A Fast Online Gait Planning with Boundary Condition Relaxation for Humanoid Robots, *Proc. of IEEE Int. Conf. on Robotics and Automation(ICRA'05)*, 2005.

[9] S. Kagami, K. Nishiwaki, T. Kitagawa, T. Sugihiara, M. Inaba, and H. Inoue, A Fast Generation Method of a Dynamically Stable Humanoid Robot Trajectory with Enhanced ZMP Constraint, *Proc. of IEEE International Conference on Humanoid Robotics*, 2000.

[10] J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, Motion Planning for Humanoid Robots Under Obstacle and Dynamic Balance Constraints, *Proc. IEEE Int. Conf. on Robotics and Automation*, 2001.

[11] J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, Motion Planning for Humanoid Robots, *Int. Symposium of Robotics Research (ISRR'03)*, 2003.

[12] J. Chestnutt, M. Lau, J.J. Kuffner, G. Cheung, J. Hodgins, and T. Kanade, Footstep Planning for the ASIMO Humanoid Robot, *Proc. IEEE Int. Conf. on Robotics and Automation*, 2005

[13] M. Stilman and J.J. Kuffner, Navigation among movable obstacles : Real-time reasoning in complex environments, *Int. J. of Humanoid Robotics*, 2(4), 2005

[14] Eiichi Yoshida, Humanoid Motion Planning using Multi-Level DOF Exploitation based on Randomized Method, *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* , 2005.

[15] Eiichi Yoshida, I. Belousov, C. Esteves, and J.-P. Laumond, Humanoid Motion Planning for Dynamic Tasks, *Proc. of IEEE-RAS Int. Conf. on Humanoid Robot*, 2005.

[16] J.J. Kuffner and S.M. Lavalle, RRT-Connect: An Efficient Approach to Single-Query Path Planning, *Proc. IEEE Int. Conf. on Robotics and Automation*, 2000.

[17] S.M. Lavalle and J.J. Kuffner, Randomized Kinodynamic Planning, *Proc. IEEE Int. Conf. on Robotics and Automation*, 1999.

[18] R. Kindel, D. Hsu, J.C. Latombe, S. Rock, Kinodynamic Motion Planning Amidst Moving Obstacles, *Proc. IEEE Int. Conf. on Robotics and Automation*, 2000.

[19] G. Sanchez and J.C. Latombe, A Single-Query Bi-Directional Probabilistic Roadmap Planner with Lazy Collision Checking, *Int. Symposium of Robotics Research (ISRR'01)*, 2001.

[20] R. Bohlin and L.E. Kavraki, Path planning using lazy PRM, *Proc. IEEE Int. Conf. on Robotics and Automation*, 2000.

[21] L.E. Kavraki, P. Svestka, J.C. Latombe, and M. Overmars, Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces, *IEEE Transactions on Robotics and Automation*, 12(4), 1996.

[22] K. Harada, S. Hattori, H. Hirukawa, M. Morisawa, S. Kajita, and E. Yoshida, Motion Planning for Walking Pattern Generation of Humanoid Robots, *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2007.

[23] http://robotics.stanford.edu/ mitul/mpk/