Proceedings of the 2007 IEEE/RSJ International
Conference on Intelligent Robots and Systems
San Diego, CA, USA, Oct 29 - Nov 2, 2007

Motion Planning for Walking Pattern Generation of Humanoid Robots

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

Abstract— **In this paper, we plan the collision free motion for walking pattern generation of a humanoid robot. Our motion planner can take into account several features of the walking pattern generator. We first run the walking pattern generator by considering the contact wrench applied to the robot and monitor the collision among the links and the environments. Then, we plan the collision free motion for the period of time causing the collision. In our motion planner, we can consider the constraint condition which are the functions of time. Also, for keeping balance of the robot, we plan the motion with keeping the horizontal position of the COG as well as the position/orientation of the feet/hand. The effectiveness of the proposed method is confirmed by simulation and experiment.**

I. INTRODUCTION

In recent years, there are many humanoid robots that can realize various whole body motion. This is because much research has been done on the motion generation of humanoid robots such as [1], [2], [3], [8]. By observing an excellent motion of a humanoid robot, we tend to have an illusion that any motion can be generated fully automatically. However, it is often the case where a human carefully chooses the motion parameters so as to avoid unnecessary collision among the links and the environments. Although the research on the motion generation of humanoid robots has been mainly done on the balance compensation, the number of research on the motion planning to avoid unnecessary collision is much smaller than those of the balance compensation.

As for the motion planning of robotic systems, the random sampling based method[22] has invested. By using the roadmap of randomly-sampled milestones, the planner generates a collision free path in the configuration space connecting the start and the goal configurations. The random sampling is quite effective since it can be applied to the robotic system with large dimension of the configuration space. Recently, the random sampling based methods have begun to apply to humanoid robots [12], [13], [16], [4], [5]. However, there is still no motion planning framework for a humanoid robot walking on flat/irregular terrain with keeping the dynamical balance.

Based on this consideration, this paper proposes a method for generating a walking pattern of a humanoid robot without causing unnecessary collision among the links and the environments. Fig. 1 shows a typical case where a humanoid robot walks through a gate. In this case, the humanoid robot has to walk with avoiding the contact with the gate.

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

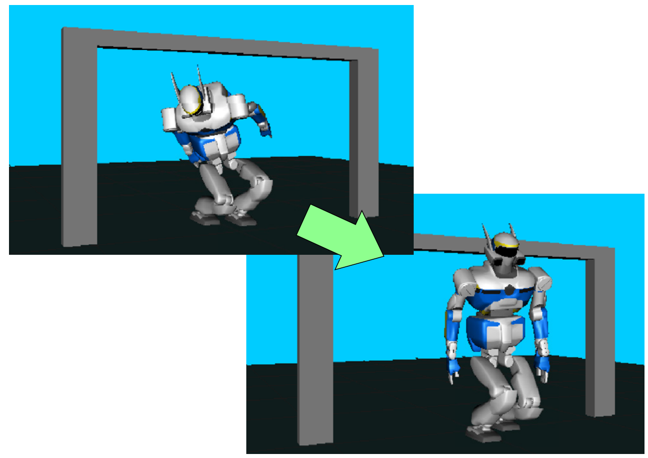


Fig. 1. A humanoid robot walking through the gate

When planning the walking pattern of a humanoid robot, there are several issues we have to consider. First, in most of the existing walking pattern generator, the walking motion is generated for a given trajectory of the ZMP (Zero Moment Point). Once the desired ZMP trajectory is given, the position/orientation of the feet are determined so as to keep the ZMP included in the support polygon. Then, by solving the ordinary differential equation, the trajectory of the CoG (Center of Gravity) is calculated. To keep balance of the robot, these trajectories become the constraint conditions. Hence, when planning the motion of a humanoid robot, we have to consider the constraint conditions which are the functions of time.

Secondly, there is a problem in the ZMP itself. When a humanoid robot walks, the robot receives the six dimensional force/moment from the environment. The contact state between the robot and the environment changes depending on the force/moment. However, the ZMP is only the two dimensional information expressing the position of the center of pressure on the ground. Especially when walking on uneven terrain, it becomes often difficult to judge whether or not the robot falls down [6]. For such kind of problem, the walking pattern generator considering the full six dimensional force/moment has been researched[7].

Thirdly, as of now, there is no research on planning the motion of a humanoid robot integrated with the walking pattern generator like the one mentioned above. It is an open problem which DOF of the robot should be used for planning the motion of the robot when generating the walking pattern of the robot.

In order to overcome these difficulties, we propose a new motion planning method. In this method, we first run the walking pattern generator taking the six dimensional force/moment into account. When generating the walking motion, we monitor whether unnecessary collision occurs or not. If the collision occurs, we run the motion planner for the period of time causing the unnecessary collision. Our random sampling planner is a newly proposed one considering the constraints which are the functions of time. To deal with the constraints, we consider adding an additional parameter to each milestone. We further integrate the motion planner with the walking pattern generator. The effectiveness of the proposed method is confirmed by simulation and experiment.

II. RELATED WORKS

Walking Pattern Generation of Humanoid Robots:

As for the walking pattern generation of humanoid robots, many researchers use the ZMP. Takanishi et al.[10] proposed a method to calculate the trunk motion by using the FFT. Kagami et al.[11] proposed an approach numerically solving the differential equation of the trunk motion. Kajita et al.[1] used the preview control. Nagasaka et al.[9], Harada et al.[3] and Sugihara et al.[8] proposed the analytical solution based approaches.

Recently, Hirukawa et al.[6] proposed a method to judge the transition of contact state between the robot and the environment by using the six dimensional force/moment so called the Contact Wrench Sum (CWS) applied to the robot. They[7] also proposed a method to generate the walking pattern by using the CWS.

Motion Planning with Collision Avoidance:

Kavraki et al.[22] proposed the probabilistic roadmap (PRM) planner. Sanchez et al.[21] later proposed the SBL (Single-query, Bi-directional, Lazy in Collision checking) of the single-query method. The random sampling method was extended to the mechanical system with closed kinematic chain[18], [19] and with velocity constraints[20], [23].

Recently, some researchers research the motion planning of a humanoid robot. Kuffner et al.[12], [13] first generate the collision free motion satisfying the statical balance and then transformed it to the dynamically stable motion. Chestnutt et al.[14] proposed a heuristic method of footstep planning. Stilman et al.[15] proposed the manipulation planning of a movable object. Yoshida [16] approximated the shape of the robot by a rectangular and extended the approach for a omni-directional vehicle. Hauser et al. [4] proposed the multi-step planning method applicable to the rock-climbing humanoid robot. They [5] also proposed a method to guide the search by using the motion primitives. In this research, by adapting the primitive to the environment model, the PRM is performed within the period of time when the collision occurs.

However, there is no research on motion planning of a humanoid robot walking on the flat/rough terrain with keeping the dynamical balance. Also, the feature of the proposed method is that we can plan all the DOF of the robot under constraint.

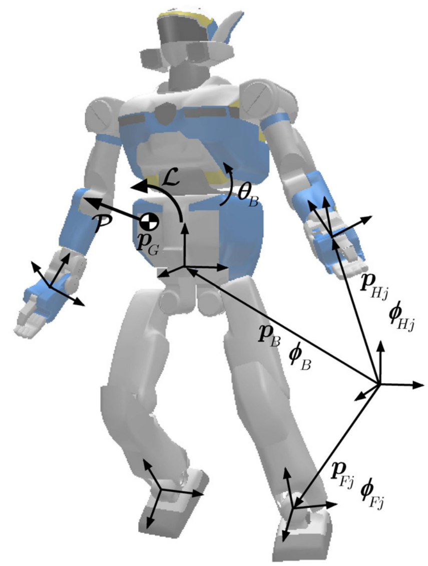


Fig. 2. Model of HRP2 Humanoid Robot

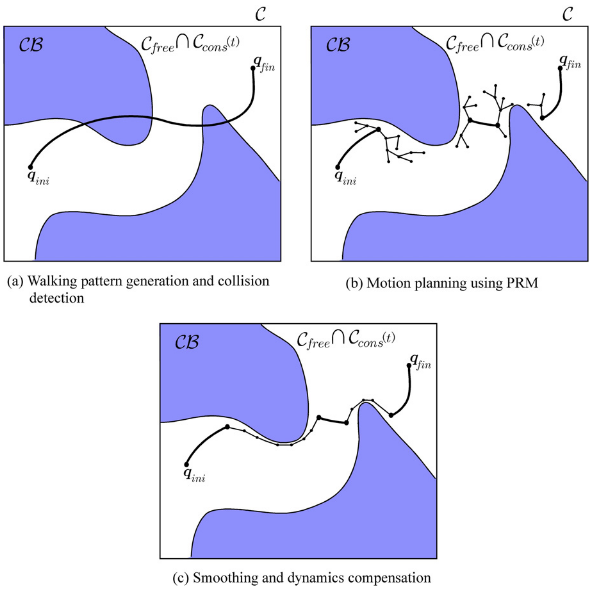
III. PROPOSED METHOD OVERVIEW

Fig. 2 shows the model of a humanoid robot. Let $\boldsymbol{p}_*/\boldsymbol{\phi}_*$ be the three dimensional vectors of the position/orientation of the coordinate frame fixed to a link. The subscripts Fj, Hj, B and G denote the j-th foot, the j-th hand, 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 $q \in \mathcal{C}$ of the humanoid robot is composed of the position/orientation of the waist ($\boldsymbol{p}_B/\boldsymbol{\phi}_B$) and all the joint angles ($\boldsymbol{\theta}$). 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 motion of it so that these links keep contact with the environment at the desired position within the specified period of time. At the same time, the robot has to avoid the unnecessary collision between the links of the robot and between a link and the environment. $\mathcal{C}_{free} \subset \mathcal{C}$ denotes the set of configurations such that unnecessary collision does not occur. Furthermore, we will impose the desired trajectories for the position/orientation of some part of the robot (such as the feet, the hands, and the vertical position of the CoG). We regard the condition for the robot to follow the desired trajectory as the constraint conditions for the robot's configuration. These constraint conditions are the functions of both the robot's configuration q and the time t having the form:

$$f(q, t) = 0. \tag{1}$$

Let $\mathcal{C}_{cons}(t) \subset \mathcal{C}$ be the subset of the configuration such that the robot follows the desired trajectory at the specified time. In our motion planning problem, we search for the



(a) Walking pattern generation and collision detection

(b) Motion planning using PRM

(c) Smoothing and dynamics compensation

Fig. 3. Overview of the proposed method

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 $\mathcal{C}_{free} \cap \mathcal{C}_{cons}(t)$.

Fig. 3 shows the overview of the motion planner proposed in this paper. As shown in Fig. 3 (a), we first run the walking pattern generator explained in Section V. While executing the walking pattern generator, we monitor whether the unnecessary collision occurs or not. If the collision does not occur, the motion planner returns the joint trajectory of the robot. On the other hand, if unnecessary collision occurs, we plan the collision free path of the configuration within the period of time causing the collision by using the method explained in Section IV. Then, the planner returns the collision free joint trajectory. If the robot cannot keep balance due to the angular momentum of the robot, then the walking pattern generator generates the walking motion again.

IV. MOTION PLANNER

Our proposed method can deal with the constraint conditions which are the functions of time. Without the constraint condition, our planner is same as the SBL (Single-query, Bi-directional, Lazy in collision checking) planner[21]. The SBL planner incrementally constructs a network of milestones made of two trees T_{st} or T_{ed} rooted at q_{st} and q_{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 the collision of the path between two collision free milestones included in the path.

In the following, we explain how to include the constraint condition into the planner. Fig. 4 shows the overview of the proposed planner. As shown in Fig. 4(a), while executing the walking pattern generator, we record the period of time when the collision occurs. Then, before starting the motion planner, we determine the time of the start/goal of the planner, t_0

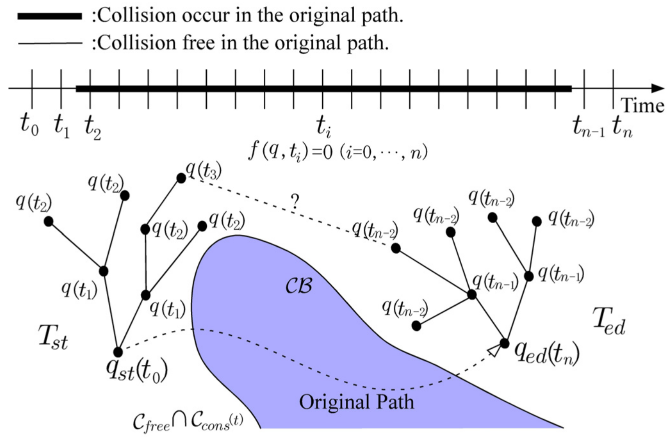


Fig. 4. The method to include the constraint condition

and t_n, so that the period of collision is included between t_0 and t_n. We further consider descretizing the time span as $t = t_0, t_1, \cdots, t_{n-1}, t_n$. These are used for an additional parameter of each milestone. If the time t_i is associated with the milestone m, then we have

$$t_i = \text{time}(m). \tag{2}$$

Also, this milestone m has to satisfy the constraint condition $f(q, t_i) = 0$.

A. Tree Expansion

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

Algorithm 1: Tree Expansion Rule for T_{st}

1) A (father) milestone m_f is picked from T_{st}.
2) If m_f satisfies $\text{time}(m_f) = t_i$ and $q(t_i) \in \mathcal{C}_{free} \cap \mathcal{C}_{cons}(t_i)$, then the son milestone m_s is set to be satisfying $\text{time}(m_s) = t_{i+1}$ 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_f satisfies $\text{time}(m_f) = t_i$ and $q(t_i) \in \mathcal{C}_{free} \cap \mathcal{C}_{cons}(t_i)$, its son m_s is set to be satisfying $\text{time}(m_s) = t_{i-1}$ 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: Implimentation Rule for Tree Expansion

1) At the 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) For the links imposing the constraint conditions $f(q, t_j) = 0$, their position/orientation are replaced by the desired ones. Then, the new configuration is obtained by solving the inverse kinematics.
4) Check whether or not the new configuration is collision free and its configuration is within the movable range.

[1] We defined the metrix in the configuration space by D cube

5) If all the conditions are satisfied, we select the new milestone as a son of its father. Otherwise, go to Step 1.

We will explain more concretely about this algorithm in subsection IV-D.

B. Path Generation

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

Algorithm 3: Tree Connection for T_{st}

1) Select the milestone m' of T_{ed} closest to m_s where satisfying $\mathrm{time}(m') = t_j$ and $q(t_j) \in \mathcal{C}_{free} \cap \mathcal{C}_{cons}(t_j)$.
2) If the distance between these two milestones is less than ρ and $t_i < t_j$ is satisfied, determine that the trees are connected

If the trees are connected, the candidate of path connecting q_{st} and q_{ed} can be generated. The condition $t_i < t_j$ is needed because, when a path is generated connecting q_{st} and q_{ed}, the time associated with each milestone of the path becomes an increasing order from the start to the goal.

Same discussion can be hold for the case where the new milestone m_s belongs to T_{ed} except for the condition $t_i > t_j$.

C. Lazy Collision Checking

Once a path candidate is determined, we check the collision of each segment connecting two milestones belonging to the path. Fig. 5 shows the overview of the segment checking method. Let us consider the segment connecting two milestones m_i and m_j. We consider the case where m_i and m_j satisfy $\mathrm{time}(m_i) = t_i$ and $\mathrm{time}(m_j) = t_j$, respectively. By simply using the straight line, there is no guarantee that the constraint condition is satisfied at the point on the line segment. Hence, we use the following algorithm:

Algorithm 4: Collision Check of a Path

1) Connect m_i and m_j by using the straight line segment.
2) At the point on the line segment dividing the segment by $\delta t/(t_j - t_i) : (t_j - t_i - \delta t)/(t_j - t_i)$, obtain the configuration of this point as

$$q(t_i + \delta t) = \frac{\delta t q(t_j) + (t_j - t_i - \delta t) q_(t_i)}{t_j - t_i} \qquad (3)$$

3) By using the Step 2 - 5 of Algorithm 2 with the constraint condition $f(q, t_i + \delta t) = 0$, a new configuration satisfying $q'(t_i + \delta t) \in \mathcal{C}_{cons}(t_i + \delta t)$ can be obtained.
4) Check if this configuration satisfies $q'(t_i + \delta t) \in \mathcal{C}_{free}$ and if this configuration is within the movable range.

How to select a point on the line segment follows [21]. If the segment is find out to be collision free within the predefined resolution of the segment, the segment is marked $safe$. Also, if all the segment belonging to the path is marked $safe$, the planner returns this path.

Finally, we used a simple path smoother to minimize the path length. In the path smoother, we checked the collision of the path segment by using the same method of this section.

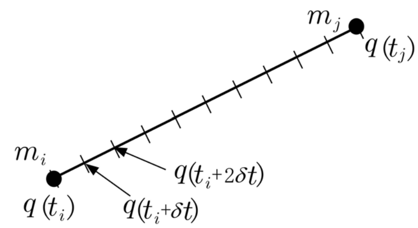


Fig. 5. The method to check the collision of the segment.

D. Implimentation Details

In our implementation, $f(q, t)$ includes the difference between the current and the desired position/orientation of both feet, those of both hands and the horizontal position of the CoG. Here, since it is difficult to precisely follow the horizontal position of the CoG to the desired trajectories, we use the following approximation when expanding trees and checking collision of the path. First, a configuration is (randomly) selected. Second, by solving the forward kinematics, we obtain the position/orientation of both feet, those of both hands, and the horizontal position of the waist. After replacing them by the desired values, we calculate the horizontal position of the CoG. To compensate the difference between the current and the desired position of the CoG, we modify the position of the body. Finally, the configuration is obtained by solving the inverse kinematics again. This configuration precisely satisfies the constraint condition for the feet and the hands and approximately satisfies the constraint condition for the horizontal position of the CoG.

V. WALKING PATTERN GENERATOR

By using the motion planner explained in the previous section, we now state how to generate the collision free walking motion of the robot with keeping the dynamical balance.

A. Contact Wrench Sum[7]

Let *Contact Wrench Sum*(CWS) be the sum of the force/moment applied to the robot. Also, let *Contact Wrench Cone* (CWC) be the set of the CWS spanned by the sum of the friction cone at each contact point. The contact should be strongly stable if the CWS is an internal element of the CWC under sufficient friction. By using the CWS, we can generate the walking pattern of a humanoid robot under arbitrary terrain. On the other hand, the CWS and the CWC are equivalent to the ZMP and the support polygon, respectively, when the robot walks on a flat plane.

Once the desired trajectory of the CWS is given, we can obtain the following differential equation with respect to the balance of the moment about the horizontal axes:

$$
\begin{aligned}
&M(\ddot{z}_G+g)(y_G-y_C)-M\ddot{y}_G(z_G-z_C)+\dot{\mathcal{L}}_x \\
&=\sum_{k=1}^{K}\epsilon_k^0(y_k n_{kz}-z_k n_{ky}) \\
&-M(\ddot{z}_G+g)(x_G-x_C)+M\ddot{x}_G(z_G-z_C)+\dot{\mathcal{L}}_y \\
&=-\sum_{k=1}^{K}\epsilon_k^0(x_k n_{kz}-z_k n_{kx})
\end{aligned} \tag{4}
$$

where $\boldsymbol{p}_G = (x_G \ y_G \ z_G)$ and M is the total mass of the robot. $\boldsymbol{p}_C = (x_C \ y_C \ z_C)$ and the right-hand side of (4) are the functions of the CWS[7]. (4) is a straight-forward extension of the equation of the ZMP to that of the CWS. As is the case of the ZMP equation, we can solve this equation for x_G and y_G by using the finite differentiation[11] of (4).

Once x_G and y_G are obtained, we can calculate the linear/angular momentum of the robot. By using the reference linear/angular momentum $\mathcal{P}^{ref}$ and $\mathcal{L}^{ref}$, the desired trajectory of the waist link can be calculated by using the following equation[2]:

$$
\begin{bmatrix} \dot{\boldsymbol{p}}_B \\ \boldsymbol{\omega}_B \end{bmatrix} = \boldsymbol{A}^{\dagger}\boldsymbol{y}, \tag{5}
$$

where $*^{\dagger}$ denotes the pseudo inverse of $*$ and

$$
\begin{aligned}
\boldsymbol{y} \equiv \begin{bmatrix} \mathcal{P}^{ref} \\ \mathcal{L}^{ref} \end{bmatrix} - \sum_{i=1}^{2} \begin{bmatrix} \boldsymbol{M}_{F_i}^* \\ \boldsymbol{H}_{F_i}^* \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{p}}_{Fi}^{ref} \\ \boldsymbol{\omega}_{Fi}^{ref} \end{bmatrix} \\
-\sum_{i=1}^{2} \begin{bmatrix} \boldsymbol{M}_{H_i}^* \\ \boldsymbol{H}_{H_i}^* \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{p}}_{Hi}^{ref} \\ \boldsymbol{\omega}_{Hi}^{ref} \end{bmatrix}.
\end{aligned} \tag{6}
$$

For more precise definition of (5), please refer [2], [7]. Once the reference trajectory of the waist link is given, the joint trajectory of the robot can be obtained by solving the inverse kinematics.

However, to calculate the reference $\mathcal{P}^{ref}$ and $\mathcal{L}^{ref}$, we need the information of the linear/angular velocity of the waist. Thus, we use the following iteration rule[7] to generate the walking pattern with keeping the dynamical balance of the robot.

Algorithm 5: Walking Pattern Generation

1) Give the reference trajectories of $(\boldsymbol{p}_{Fi}, \boldsymbol{\phi}_{Fi})$, $(\boldsymbol{p}_{Hi}, \boldsymbol{\phi}_{Hi})$, the CWS and the initial reference trajectory of $(\boldsymbol{p}_B, \boldsymbol{\phi}_B)$.
2) Calculate x_G and y_G by solving (4).
3) Calculate $(\boldsymbol{p}_B, \boldsymbol{\phi}_B)$ by using (6). If this $(\boldsymbol{p}_B, \boldsymbol{\phi}_B)$ is close enough to the ones given in Step 1, terminate the algorithm. Otherwise, go to Step 2 by using this $(\boldsymbol{p}_B, \boldsymbol{\phi}_B)$ as the initial reference.

B. Combination of Motion Planner and Pattern Generator

Fig.6 shows the method to combine the walking pattern generator with the motion planner.

First, by using Algorithm 5, we obtain the walking motion of a humanoid robot. While generating the walking pattern, we check if unnecessary collision occurs or not. If the collision occurs, by using the trajectories of $(\boldsymbol{p}_{Fi}, \boldsymbol{\phi}_{Fi})$, $(\boldsymbol{p}_{Hi}, \boldsymbol{\phi}_{Hi})$ and $(x_G, \ y_G)$, the motion planner plans the collision free path of the robot's configuration. The output of this planner is the trajectories of the vertical CoG (z_G), the body's orientation ($\boldsymbol{\phi}_B$) and the additional joint angles ($\boldsymbol{\theta}_B$) such as the waist joint. After planning the motion, the difference between the actual and the reference CWS might occur. If this difference is large, we generate the walking motion again by using the output of the motion planner.

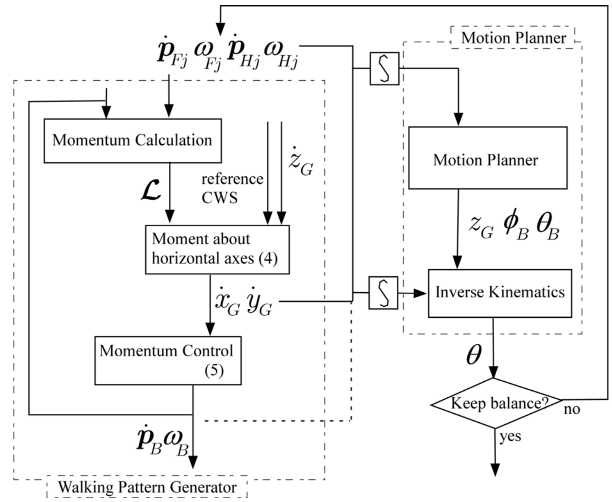


Fig. 6. The method to combine the walking pattern generator with the motion planner

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

Algorithm 6: Combination of Motion Planner and Walking Pattern Generator

1) Generate the walking motion by using Algorithm 5.
2) If the undesired collision occurs while generating the walking motion, plan z_G, $\boldsymbol{\phi}_B$ and $\boldsymbol{\theta}_B$.
3) If the difference between the actual CWS and the reference CWS is large, goto Step 1. Otherwise, terminate the algorithm.

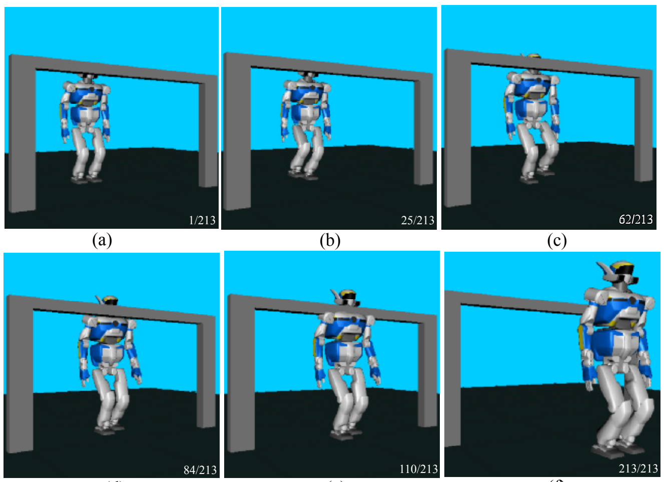
VI. SIMULATION AND EXPERIMENT

We performed the simulation of walking through the gate by using 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 combined with the motion planner MPK (Motion Planning Kit)[24].

Table I shows the calculation time of the proposed algorithm using a Xeon 2.4[GHz] PC. In Algorithm 5 of the walking pattern generator, we iterate Steps 2 and 3 for five times. Also, as the path smoother, we iterate the smoothing algorithm briefly explained in subsection *IV C* for 150 times. After smoothing the path, we further smoothe the path using the Spline function taking about a minute and then terminate the algorithm. Among these steps, we show the time for the walking pattern generator, the path planner and the former

TABLE I
CALCULATION TIME

Trial #	1	2	3	4	5
Walking pattern generate [s]	45.0	44.9	45.2	45.1	44.9
Motion plan [s]	13.6	48.4	19.3	37.8	14.9
Path smoothe [s]	24.8	41.4	28.9	39.1	41.0



(a) (b) (c) (d) (e) (f)

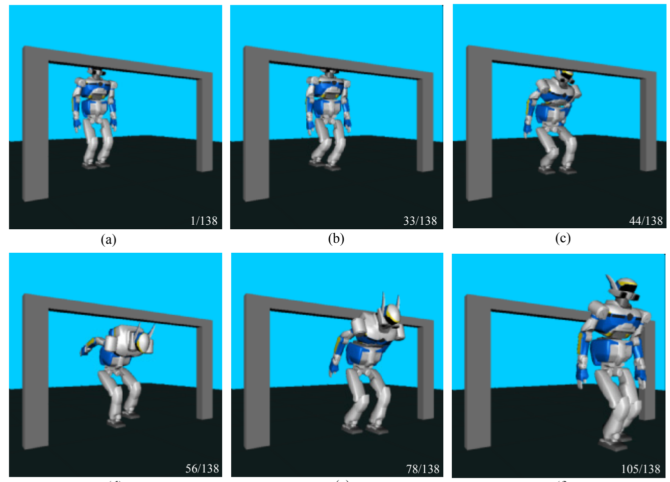
Fig. 7. Original motion of the robot generated by the walking pattern generator

path smoother in this table. As shown in this table, these steps for the robot's motion of 15[s] is calculated within two minutes.

Fig.7 shows the output of the walking pattern generator. As we can see from Fig.7 (d) and (e), the 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.8, the robot avoids the collision between the robot and the gate.

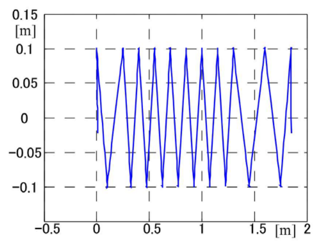
Fig. 9 shows the ZMP trajectories during the simulation. We set the reference CWS so that the horizontal ZMP position coincides with the rotation center of the ankle joint during the single support phase. Fig. 9(a) shows the ouput of the walking pattern generator. As shown in this figure, the desired CWS is realized. On the other hand, Fig. 9(b) shows the ZMP trajectory after planning the collision free motion. A slight deviation from the ZMP trajectory of (a) occured (although it is not significant). Furthermore, in Fig. 9(c), we generated the walking pattern again by using the output of the motion planner. As shown in this figure, the ZMP deviation becomes smaller than that of (b).

Fig. 10 shows the experimental result. This is the experiment corresponding to the simulation result of Fig.9(b). As shown in this result, the humanoid walks through the gate with keeping balance of the robot.

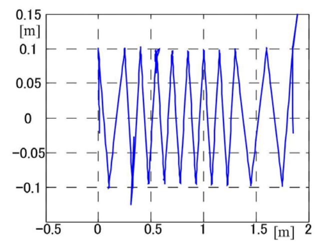


(a) (b) (c) (d) (e) (f)

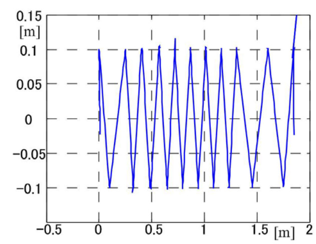
Fig. 8. The collision free motion of the robot



(a) Output of the walking pattern generator



(b) Output of the motion planner



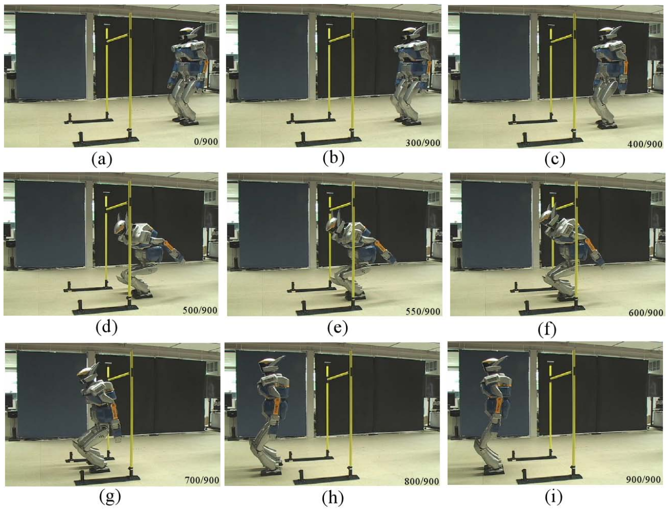
(c) Compensation of the ZMP error caused in the motion planner

Fig. 9. Simulation result of the ZMP trajectory

VII. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

In this paper, we proposed a motion planner for walking pattern generator of a humanoid robot. In our motion planner, we first monitor the collision among the links and between the robot and the environment while running the walking pattern generator by using the Contact Wrench Sum (CWS). Then we performed the motion planner. Since the time-variant constraint condition exists in walking pattern generator, out planner can consider this constraint condition. Also, we plan the motion so as to keep balance of the robot. The simulation and experimental results show that we can avoid the collision with keeping the walking motion of the robot. The results also show that we can plan the collision free motion within the reasonable time.



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Fig. 10. Experimental result

B. Future Works

We have many future works. First, since we explicitly specified both the start and the end time of the motion, there is a case where the motion of the robot avoiding the obstacle becomes very quick. For such cases, currently we have to manually adjust both the start and the end time of the motion. We will consider the time effect when planning the collision free motion of the robot. Second, in our proposed algorithm, we plan the collision free motion after generating the walking pattern. Simultaneous generation of the walking pattern and the collision free motion is our future research topic. Thirdly, we will plan the foot step on uneven surface of the environment within our framework.

REFERENCES

[1] 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.

[2] 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, *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1644-1650, 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] K. Hauser, T. Bretl, J.-C. Latombe, Non-gaited Humanoid Locomotion Planning, *Proc. of IEEE Int. Conf. of Humanoid Robots*, 2005

[5] 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.

[6] 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, *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 1976-1983, 2006

[7] 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, *Proc. IEEE Int. Conf. Robotics and Automation*, 2007

[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)*, pp.306-311, 2005.

[9] K. Nagasaka, Y. Kuroki, S. Suzuki, Y. Itoh, J. Yamaguchi, Integrated Motion Control for Walking, Jumping and Running on a Small Bipedal Entertainment Robot, *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 3189-3194, 2004.

[10] 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.

[11] 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.

[12] 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*, pp. 692-698 (ICRA 2001)

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

[14] 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

[15] M. Stilman and J.J. Kuffner, Navigation among movable obstacles : Real-time reasoning in complex environments, *Int. J. of Humanoid Robotics*, 2(4), pp.1-24, 2005

[16] 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* , pp. 25-30, 2005.

[17] 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*, pp. 1-6, 2005.

[18] S.M. Lavalle, J.H. Yakey, and L.E. Kavraki, A Probabilistic Roadmap Approach for Systems with Closed Kinematic Chains, *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 1671-1676, 1999.

[19] J. Cortes, T. Simeon, J.P. Laumond, A Random Loop Generator for Planning the Motions of Closed Kinematic Chains using PRM Methods, *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 2141-2146, 2002.

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

[21] 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.

[22] 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), pp.566-580, 1996.

[23] S.M. LaValle, Planning Algorithms, *Cambridge University Press*, 2006.

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