2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Madrid, Spain, October 1-5, 2018

Walking on a Steep Slope Using a Rope by a Life-Size Humanoid Robot

Masahiro Bando, Masaki Murooka, Shunichi Nozawa, Kei Okada, and Masayuki Inaba

***Abstract*— In this paper, we propose methods for walking on a steep slope using a rope by a humanoid robot. There are two difficulties for walking on a steep slope without a rope. First, range of motion of ankle joints get limited. Second, feet of a robot slip on a steep slope. For these problems, using a rope is effective solution because the robot can receive enough friction force from the slope and walk on a steep slope by pulling a rope with proper tension. In addition, the robot pulling a rope on a slope can relax limitations of ankle joints. Therefore, we propose methods to determine tension of a grasped rope by solving a linear least-square problem considering deformability of a rope. With these methods, a life-size humanoid robot HRP-2 could walk on a steep slope which angle is 40 degree.**

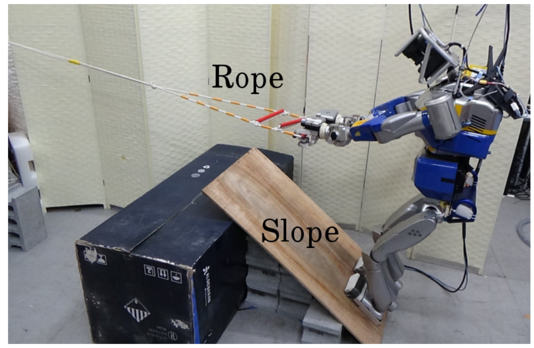

Fig. 1. HRP-2 using a rope on a 40 degrees slope: By pulling a rope, the robot can increase the friction force from the slope and ensure range of motion of its ankle joints.

I. INTRODUCTION

Generally, a biped robot is expected to show more universal ability for locomotion by using its legs than a wheel type robot. Especially, a humanoid robot is able to use not only its legs but also its arms for locomotion, so it can execute various type of locomotion. There are a lot of studies on full-body locomotion such as climbing of a ladder[1][2] and walking up a stair with a hand rail[3]. In this paper, we propose walking up on a steep slope by using a rope as another type of full-body locomotion to extend humanoid robot's ability for locomotion.

Walking on a slope itself does not always require a rope. When a slope is not so steep, many studies show that a humanoid robot can walk up on a slope without any assist[4]. However, it should use a rope when it walks on a steep slope as a human climber does while climbing a mountain surface.

There are two advantages for using a rope on a steep slope. The first one is ensuring the range of motion of pitch joint of ankle. When standing on a steep slope with only legs, a humanoid robot has to make its instep close to the front of its lower leg. If there is not enough range of motion at its ankles, its ankle will reach the limit of the joint during walking motion, and the robot will not be able to continue the motion. This is more critical problem for humanoid robots walking with their knees bended because the insteps and front of the legs get closer when their knees are bended. There are several solutions for this problem. One solution is taking a forward bending posture to relax ankle joints, and another one is using a robot which has very large joint flexibility at ankles. Our solution for this problem is pulling a rope by the robot's hands in order to keep balance in the state where the robot's hip is behind.

The second advantage is that a robot can reduce the risk of slipping on a slope. To walk on a slope without a rope, the static friction coefficient between a slope and robot's feet must be large enough. If the static friction coefficient μ is lower than $\tan\theta$ (θ is the angle of a slope), the robot's feet slip against the slope. This is an inevitable problem when the robot is standing on the slope without any support, and using of a rope is a simple solution for this problem. When the robot is pulling a rope as Fig.I, the normal component of reaction force from the slope increases. In addition, the tension of the rope counteracts the component of the gravity parallel to the slope. As a result, the robot can acquire enough friction force to walk on the slope by pulling a rope.

To use a rope as support, it is important to determine the tension of a grasped rope so that a robot can keep its balance. In this paper, we propose a method to check the stability of a robot pulling a rope and an algorithm to determine reference tension of a rope considering constraint on rope.

M. Bando, M. Murooka, S. Nozawa, K. Okada, and M. Inaba are with Department of Mechano-Infomatics, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan bando at jsk.t.u-tokyo.ac.jp

II. APPROACH TO SLOPE WALKING USING ROPE

A. *Overview of the Proposed System*

The overview of our proposed system is Fig.2. The system is mainly composed of the following modules:

1) Stability Check by QP: This module checks the stability of a robot pulling a rope on a slope with certain posture by quadratic programming(QP). This module is used in the algorithm to search to proper reference tension of a rope.
2) Reference Rope Tension Search: This module searches proper reference tension of a rope for walking on a slope. It decides reference tension based on our proposed algorithm and sends reference forces of the robot's hands to control modules.
3) Walking Pattern Generator: We use preview control based on table-cart model for generation of walking

pattern[5]. This method is suitable for real-time walking pattern generation.

4) Stabilizer: We use stabilizer based on Kajita's stabilizer[6] for stabilization during walking. It receives reference positions of zero-moment point (ZMP) and center of gravity (CoG) of the robot from the Walking Pattern Generator.
5) Impedance controller: We apply impedance control[7] to the hands of the robot. It receives reference forces exerted to the robot's hands from Reference Rope Tension Search and controls the position of the hands.

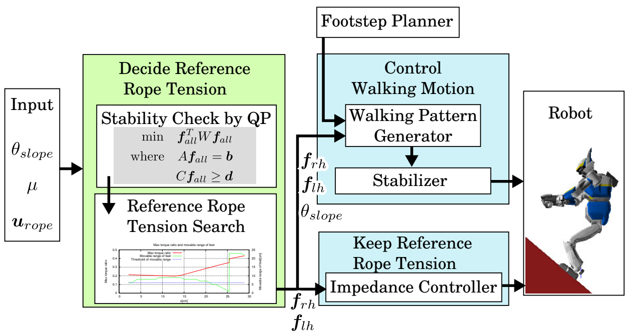

Fig. 2. The overview of the proposed system

We explain the detail of the stability check in Sec. III, the detail of reference rope tension search in Sec. IV. In Sec. V, we explain an experiment of walking on a steep slope with a rope by HRP-2, then we report our conclusion in Sec. VI.

B. Related Works

1) Climbing of a Ladder by a Robot: One of the examples that a humanoid robot uses its arms for locomotion is climbing a ladder. There are several studies on climbing a ladder. Vaillant et al. conducted vertical ladder climbing by HRP-2 using real-time closed-loop QP controller[1].

The main difference of using a rope from ladder climbing is that the force exerted to a robot's hands from a rope has very limited direction. It is because of the deformability of a rope. Therefore, we represent the deformability as inequality constraint and use the constraint to check the stability of a robot.

2) Climbing Stairs Using a Handrail: Kudruss et al. proposed optimal control for whole-body motion generation and conducted climbing stairs using a handrail by HRP-2[3]. In the study, HRP-2 could climb stairs more efficiently by using a handrail than not using.

Compared to using a handrail, using a rope has a merit that a robot can grasp the rope in front of its body, so it is easy to use both hands. In addition, a rope can be used at the situation in which there is no handrail and stairs, such as the natural environments. It is useful for a humanoid robot to acquire the ability for locomotion in the natural environment by using a rope for the support.

3) Robot Using a Rope for Walking on a Slope: Doi et al. developed TITAN-XI, the quadruped robot which can move on a slope[8]. The robot has a winch to use a rope for the assist of walking. By using a rope, the robot can prevent falling down backward on a steep slope.

In our study, the main purpose of using a rope is not to avoid falling down. A humanoid robot can move its CoG right above its feet by moving its upper body properly, so it is not so hard for a humanoid robot to avoid falling down. The purpose of using a rope in this paper is to fulfill the friction constraint on a steep slope, which is not satisfied by only legs.

C. Contribution of This Paper

As described in Sec. I, using of a rope has merits for walking on a slope. Especially, it is an effective method for the locomotion in the natural environment like a mountain surface. However, use of a rope for the support of locomotion by a humanoid robot is not researched well. It is probably because using a deformable object like a rope for the support has higher risk of losing balance than using a rigid object. To use a rope without losing balance, it is important to determine the tension of a grasped rope properly considering the deformability of the rope.

Therefore, we propose two methods to determine the reference tension of the rope. The first one is a method to check the stability of a robot pulling a rope and calculate contact forces and tension of a rope. We express the characteristic of a rope as inequality constraints, and the stability can be checked by solving a least-square problem including the inequality constraints of a rope with quadruped programming. The second one is an algorithm to search proper tension by using the stability check method. This algorithm searches the proper reference tension considering torque of robot's legs and range of motion of robot's foot, so the robot can avoid too high torque to leg joints or failure of walking motion. With these methods, a robot can calculate proper tension of the grasped rope based on the angle and the static friction coefficient of a slope. We confirmed that a life-size humanoid robot can walk on a 40 degrees slope on which the robot slips without any support.

III. Stability Check by Quadratic Programming

Whether a robot can stand on a slope statically stable can be judged by considering contact forces of a robot's limbs. We assume that the robot stands on a slope as Fig.3, and the angle of the slope, static friction coefficient and the direction of the rope is known. In this situation, the following equations are satisfied when the robot keeps its stability.

$$\begin{cases} \sum_i \boldsymbol{f}_i = -m\boldsymbol{g} \\ \sum_i (\boldsymbol{p}_i \times \boldsymbol{f}_i + \boldsymbol{n}_i) = -\boldsymbol{p}_{cog} \times m\boldsymbol{g} \end{cases} \tag{1}$$

$$\Leftrightarrow A_{limbs}\boldsymbol{f}_{all} = \boldsymbol{b}_{limbs} \tag{2}$$

$$i = \{rh, lh, rf, lf\}, \ \boldsymbol{g} = (0, 0, -g)$$
$$\boldsymbol{f}_{all} = (\boldsymbol{f}_{rf}^T \ \boldsymbol{n}_{rf}^T \ \boldsymbol{f}_{lf}^T \ \boldsymbol{n}_{lf}^T \ \boldsymbol{f}_{rh}^T \ \boldsymbol{n}_{rh}^T \ \boldsymbol{f}_{lh}^T \ \boldsymbol{n}_{lh}^T)^T$$

$\boldsymbol{p}_i, \boldsymbol{f}_i, \boldsymbol{n}_i$ is respectively the position, the contact force and the contact moment of each limb. $g = 9.8[m/s^2]$ is gravity

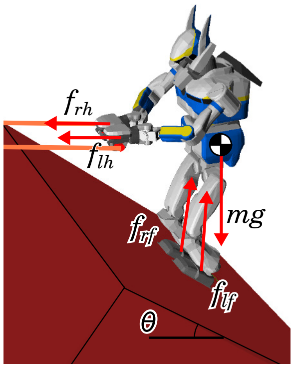

Fig. 3. State of a robot standing on a slope

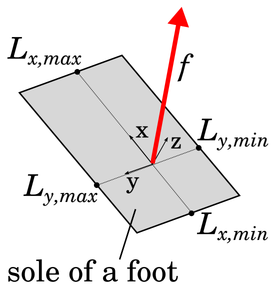

Fig. 4. Local coordinates of a sole

acceleration, and $\boldsymbol{p}_{cog}$ is the position of the robot's center of gravity. Eq.(1) can be represented as the constraint on $\boldsymbol{f}_{all}$ using a matrix $\boldsymbol{A}_{limbs}$ and a vector $\boldsymbol{b}_{limbs}$. $\boldsymbol{f}_{all}$ is a vector including all the forces and moments exerted to the robot's limbs. In addition to this equation of balance, we have to consider constraint on a rope tension and friction of feet.

Due to the characteristic of a rope, the force that can be exerted by a rope is restricted to the pulling direction. Therefore, the direction of the force exerted to the robot's hands matches the direction of the rope. This characteristic can be represented by the following equations.

$$\begin{cases} \boldsymbol{f}_i \cdot \boldsymbol{u}_1 = \boldsymbol{f}_i \cdot \boldsymbol{u}_2 = 0 \\ \boldsymbol{f}_i \cdot \boldsymbol{u}_{rope} \geq 0 \\ \boldsymbol{n}_i = 0 \end{cases} \Leftrightarrow \ C_{rope}\boldsymbol{f}_{all} \geq \boldsymbol{d}_{rope} \tag{3}$$

$$i = \{rh, lh\}, \ \boldsymbol{u}_1 \perp \boldsymbol{u}_{rope}, \ \boldsymbol{u}_2 \perp \boldsymbol{u}_{rope}, \ \boldsymbol{u}_1 \perp \boldsymbol{u}_2$$

$\boldsymbol{u}_{rope}$ is the normalized vector representing the direction of the rope which defined by the direction from the point at which the robot grasps the rope to the point at which the rope is fixed to the environment. In addition, the rope cannot exert moments to the hands because of the characteristic of the rope, so the moment exerted to the hands, $\boldsymbol{n}_i$ is 0 N/m. These inequality constraints can be converted to the inequality constraint on $\boldsymbol{f}_{all}$ using a matrix $\boldsymbol{C}_{rope}$ and a vector $\boldsymbol{d}_{rope}$.

The constraint on static friction between the slope and the robot's foot cannot be represented as a linear constraint when friction cone is strictly considered. Therefore, we approximate the friction cone as four-sided inverted pyramid. In this case, the constraint on static friction is expressed as the following equations. These equations are expressed in the local coordinate system of a foot described in Fig.4.

$$\begin{cases} |f_x| \leq \mu f_z \\ |f_y| \leq \mu f_z \end{cases} \Leftrightarrow \ C_{fri}\boldsymbol{f}_{all} \geq \boldsymbol{d}_{fri} \tag{4}$$

$\boldsymbol{f}$ is the contact force of each foot represented in the foot coordinate system, and μ is the static friction coefficient between the slope and the foot sole. We don't consider the friction constraint on the rotation around the z-axis of the foot sole. It is because the friction constraint around the z-axis doesn't become a critical problem in usual motions. In addition, accurate static friction coefficient of rotation is hard to estimate because it depends on the shape and the pressure distribution of a sole. By transforming the local coordinate system to the world coordinate system, this constraint on the friction force can be converted to the linear constraint on $\boldsymbol{f}_{all}$ using a matrix $\boldsymbol{C}_{fri}$ and a vector $\boldsymbol{d}_{fri}$.

The possibility that the robot falls down while climbing becomes high if the center of the pressure (CoP) of the foot gets too close to the edge of the foot. To avoid falling down, the robot has to satisfy the constraint on the position of the CoP. This constraint is represented as the following equations using contact force $\boldsymbol{f}$ and moment $\boldsymbol{n}$ in the foot coordinate system of Fig.4.

$$\begin{cases} -l_{x,min}f_z \geq n_y \geq -l_{x,max}f_z \\ l_{y,max}f_z \geq n_x \geq l_{y,min}f_z \end{cases} \Leftrightarrow \ C_{cop}\boldsymbol{f}_{all} \geq \boldsymbol{d}_{cop} \tag{5}$$

$$L_{x,min} \leq l_{x,min} \leq l_{x,max} \leq L_{x,max}$$
$$L_{y,min} \leq l_{y,min} \leq l_{y,max} \leq L_{y,max}$$

$l_{x,min}$, $l_{x,max}$, $l_{y,min}$, and $l_{y,max}$ are constants to decide the rectangle region in which CoP must exist. This constraint also can be represented as the constraint on $\boldsymbol{f}_{all}$ in the world coordinate system using a matrix $\boldsymbol{C}_{cop}$ and a vector $\boldsymbol{d}_{cop}$.

Using the constraints from Eq.(1) $\sim$ to Eq.(5), the stability of the robot with a certain posture can be checked by quadratic programming.

$$\min_{\boldsymbol{f}_{all}} \quad \boldsymbol{f}_{all}^T W \boldsymbol{f}_{all} \tag{6}$$
$$\text{where} \quad A_{limbs}\boldsymbol{f}_{all} = \boldsymbol{b}_{limbs} \tag{7}$$
$$C_{rope}\boldsymbol{f}_{all} \geq \boldsymbol{d}_{rope} \tag{8}$$
$$C_{fri}\boldsymbol{f}_{all} \geq \boldsymbol{d}_{fri} \tag{9}$$
$$C_{cop}\boldsymbol{f}_{all} \geq \boldsymbol{d}_{cop} \tag{10}$$

W is a weight matrix, and we use a unit matrix for a weight matrix to reduce the norm of $\boldsymbol{f}_{all}$. When this least-squares problems has no answer, the robot with the posture is not stable. If it has any answer, the target posture of the robot is feasible, and calculated contact forces are used for the reference rope tension search explained in the next section. We use QPOASES[9] to solve this least-square problem.

IV. Reference Rope Tension Search

In the previous section, we explained the method to check the stability of the robot grasping a rope with a certain posture. However, not all the posture judged as feasible is suitable for climbing motion. Some postures require too high torque of the robot's legs, and some postures don't have enough joint range of motion for walk. For stable walking on a slope, we propose an algorithm to select a posture and reference tension of a rope corresponding to the posture which satisfy several requirements. We show the overview of the algorithm in Fig.5

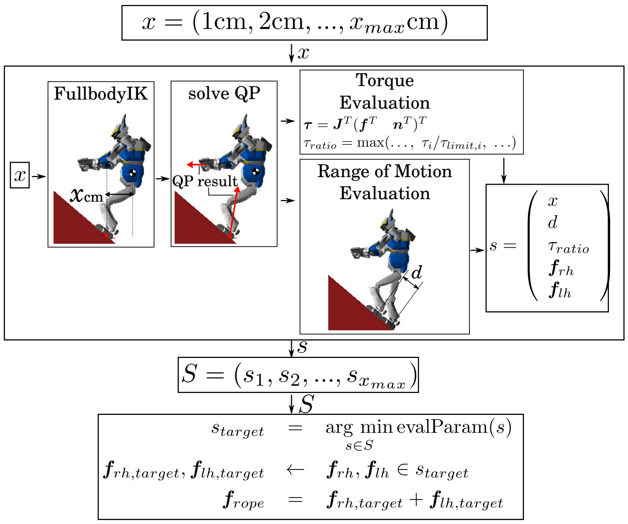

Fig. 5. Overview of reference rope tension search

We evaluate following two parameters for the search of proper reference tension. The first one is max torque ratio of legs. The torque which each joint of the legs exerts has to be less than the maximum torque of each joint. The torque required to the robot's legs can be calculated from the contact forces of the feet. The contact force $\boldsymbol{f}$ and moment $\boldsymbol{n}$ of each foot is calculated by QP of the previous section, so the joint torque $\boldsymbol{\tau}$ is represented as the following equation.

$$\boldsymbol{\tau} = \boldsymbol{J}\ (\boldsymbol{f}^T \boldsymbol{n}^T)^T \tag{11}$$

$\boldsymbol{J}$ is jacobian of the joints from the robot's root link to each foot. From the calculated $\boldsymbol{\tau}$ and the limit of joint torque $\boldsymbol{\tau}_{limit}$, the max torque ratio is determined by the following expression.

$$\tau_{max} = \max(\ldots,\ \tau_i/\tau_{limit,i}\ ,\ldots) \tag{12}$$

The second parameter to evaluate is range of motion of the robot's foot. In this algorithm, we regard the distance that the robot's foot can move backward along a slope from a certain posture as the range of motion. The distance is calculated by solving inverse kinematics repeatedly which moves the robot's foot 1cm backward along a slope, so the parameter is discrete. If the calculated distance exceed a certain threshold value, we regard that the robot's foot has enough range of motion.

The detailed algorithm of our reference rope tension search is described in Algorithm 1.

In this algorithm, a robot searches the best offset of CoG within 0cm to x_{max}cm. $\boldsymbol{q}_{init}$ is the initial posture with which the robot's CoG is right above its feet and the robot's hands grasp a rope. θ is the angle of a slope, and μ is static friction coefficient of the slope. The details of each function are as the following.

Algorithm 1 Search algorithm of reference rope tension

Require: $\theta, \mu, \boldsymbol{q}_{init}$
Ensure: $\boldsymbol{f}_{rope}, \boldsymbol{f}_{rh}, \boldsymbol{f}_{lh}$

```
x ← 0
S ← [ ]
while x ≤ x_max do
    q, x_offset ← calcTargetPosture(q_init, θ, x)
    f_rf, f_lf, f_rh, f_lh, ret ← calcContactForces(μ)
    if ret = feasible then
        τ_ratio ← maxTorqueRatio(q, f_rf, f_lf)
        d ← checkRangeOfMotionOfFeet(q, θ)
        s ← (x_offset, d, τ_ratio, f_rh, f_lh)
        push(s, S)
    end if
    x ← x + 1
end while
s_target ← argmin_{s∈S} evalParam(s)
f_rh,target ← f_rh ∈ s_target
f_lh,target ← f_lh ∈ s_target
f_rope ← f_rh,target + f_lh,target
```

1) calcTargetPosture: First, this function calculates the posture with which the robot's feet touches the slope. Then, it moves the robot's waist backward by x cm and calculates accurate offset to initial CoG, x_{offset}. If θ or x is so large that IK cannot be solved, the robot lowers its waist by 5 cm and calculates again.
2) calcContactForces: This function calculates the forces exerted to the robot's feet and hands by solving Eq.(6) with QP. It returns whether the QP problem is feasible, and it also returns the calculated contact forces when the problem is feasible.
3) maxTorqueRatio: The function to calculate the maximum value of the leg's torque ratio based on Eq.(11) and Eq.(12).
4) checkRangeOfMotionOfFeet: The function to calculate the range of motion of feet by moving foot 1cm backward along a slope repeatedly until IK cannot be solved.
5) evalParam: The return value of this function is determined by evaluating the parameters as following.

$$\text{evalParam}(s) = \begin{cases} \tau_{ratio} & d > d_{limit} \\ 1.0 & \text{otherwise} \end{cases} \tag{13}$$

We show graphs which show the results of the reference tension search in Fig.6 $\sim$ Fig.9. Fig.6 and Fig.7 are the graphs when the angle of a slope is 20 degrees. Fig.8 and Fig.9 are the ones when the angle is 40 degrees. We used a robot model of HRP-2 whose weight is 60kg for calculation, and we define that the direction of the rope is the front direction of the robot. We assumed that $l_{x,min}$, $l_{x,max}$, $l_{y,min}$ and $l_{y,max}$ in Eq.(5) are equal to zero. It means that CoP of each foot is limited to the origin of the foot coordinate system. The static friction coefficient between the slope and the foot is set to 0.7. With this friction coefficient, it is impossible to avoid slipping on a 40 degrees slope without any support. x_{max} is 40 cm, and the lower limit of the range of motion of the foot is 6cm which was determined experimentally.

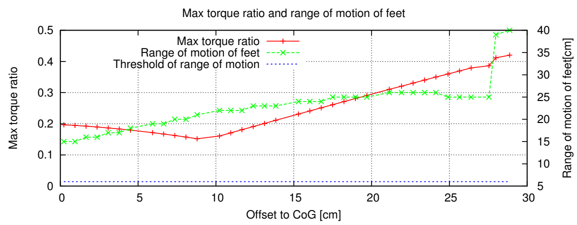

Fig. 6. Calculated torque ratio and range of motion of foot (20 degrees): The range of motion exceeds the threshold for all x_{offset}. The max torque ratio is least when $x_{offset} = 8.8$cm.

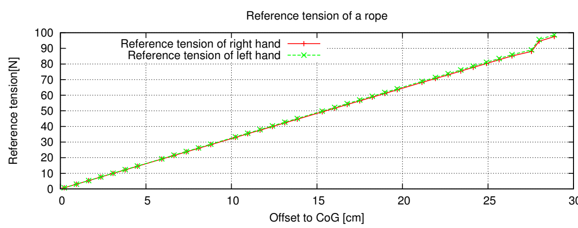

Fig. 7. Calculated reference tension of a rope exerted to each hand (20 degrees)

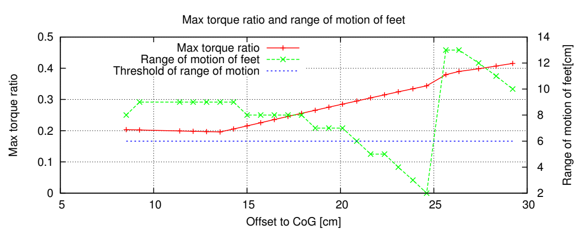

Fig. 8. Calculated torque ratio and range of motion of foot (40 degrees): The range of motion is lower than the threshold in 20cm $< x_{offset} <$ 25cm. After $x_{offset} = 25$cm, the range of motion gets high because the height of the waist gets 5cm low according to the algorithm. The max torque ratio is least when $x_{offset} = 13.6$cm.

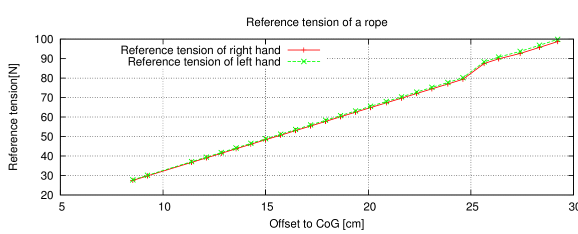

Fig. 9. Calculated reference tension of a rope exerted to each hand (40 degrees)

According to the results of the search, the range of motion of the foot is enough for walking when the angle of the slope is 20 degrees. On the other hand, the range of motion of the foot gets lower than the limit in the case of 40 degrees. In addition, the graph indicates that the tension at least about 30N is required to avoid slipping on a slope. With this robot model, the max torque ratio gets minimum at $x_{offset} = 8.8$cm when the angle is 20 degrees, so the reference tension of a rope for each hand is 28N which is the value corresponding to $x_{offset} = 8.8$cm in Fig.7. When the angle is 40 degrees, the max torque ratio gets minimum at $x_{offset} = 13.6$cm, so the reference tension is 44N according to Fig.9.

V. EXPERIMENT OF SLOPE CLIMBING WITH ROPE

A. Impedance Control Applied to Hands

To realize calculated reference forces of the robot's hands, we apply impedance control[7] to the hands. This control law is expressed as following.

$$M_p(\ddot{\boldsymbol{p}} - \ddot{\boldsymbol{p}}_d) + D_p(\dot{p} - \dot{p}_d) + K_p(\boldsymbol{p} - \boldsymbol{p}_d) = \boldsymbol{f} - \boldsymbol{f}_d \quad (14)$$
$$M_r(\ddot{\boldsymbol{r}} - \ddot{\boldsymbol{r}}_d) + D_r(\dot{\boldsymbol{r}} - \dot{\boldsymbol{r}}_d) + K_r(\boldsymbol{r} - \boldsymbol{r}_d) = \boldsymbol{\tau} - \boldsymbol{\tau}_d \quad (15)$$

$\boldsymbol{f}$ and $\boldsymbol{\tau}$ are actual force and moment exerted to the hand, and $\boldsymbol{f}_d$ and $\boldsymbol{\tau}_d$ are reference ones. $\boldsymbol{p}$ and $\boldsymbol{r}$ are actual position and attitude of the hand, and $\boldsymbol{p}_d$ and $\boldsymbol{r}_d$ are reference ones. $\boldsymbol{f}_d$ is reference force of each hand determined by our algorithm to search proper tension of a rope. M, D, and K are gain constants, and we use the following parameters.

$$M_p = 10, D_p = 200, K_p = 100$$
$$M_r = 5, D_r = 100, K_r = 20, \boldsymbol{\tau}_d = \boldsymbol{0}$$

B. Walking Pattern Generation by Preview Control and Stabilizer

Walk motion generation on a slope is composed of feed-forward walking pattern generator and feedback stabilizer during walk.

The feed-forward walking pattern generator receives reference footsteps as input and generates trajectory of CoG by preview control[5] of zero-moment point (ZMP). The generator has to generate the CoG trajectory considering tension of a rope in order to use a rope on a slope. Therefore, it adds offset to the original trajectory which doesn't consider the tension in order to counteract the tension. The offset is determined so that ZMP of a robot pulling a rope matches the original reference ZMP, so the offset can be calculated by following equations.

$$\delta x = \frac{1}{mg} \sum_{i \in \{rh,lh\}} (-f_{i,x}(z_i - z_{zmp}) + f_{i,z}(x_i - x_{zmp})) \quad (16)$$
$$\delta y = \frac{1}{mg} \sum_{i \in \{rh,lh\}} (-f_{i,y}(z_i - z_{zmp}) + f_{i,z}(y_i - y_{zmp})) \quad (17)$$

$\boldsymbol{f}_i$ is reference tension determined by our search algorithm. x_i, y_i, and z_i are positions of a hand grasping a rope. x_{zmp}, y_{zmp}, and z_{zmp} are reference values of ZMP. Based on the

CoG trajectory applied the offset, the generator solves inverse kinematics of legs and generates walking motion.

For the feedback stabilizer during walking, we use Kajita's stabilizer[6]. It receives reference ZMP and CoG from the walking pattern generator and controls so that actual values match the reference values.

C. Configuration of Rope and Robot Hand in Experiment

To make it easy to grasp a rope, we use a rope ladder shown in Fig.10 for the experiment because it doesn't lose the basic characteristic of a rope, deformability. The rope ladder is composed of plastic pipes and a rope, and the distance between the adjacent pipes is about 12cm. The bars of the rope ladder are colored red to detect positions of the bars easily. We added orange stripe pattern to the bars to get clear point cloud by a stereo depth camera. Fig.11 shows how the robot grasps the rope. In the state of Fig.11, moment from the rope exerted to the robot's hands can be ignored. This condition matches the constraint of a rope described in Eq.(3).

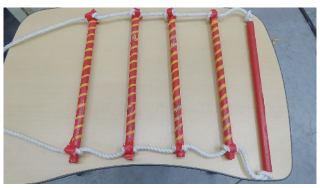

Fig. 10. Rope ladder used in the experiment.

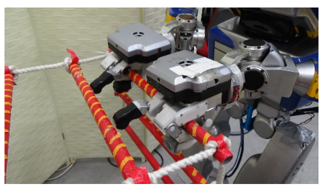

Fig. 11. Pose when the robot grasps the rope.

D. Recognition of Rope Ladder for Grasping

When a robot uses a rope ladder, the robot has to grasp the next bar of the rope ladder after it has walked a certain distance on a slope. In this paper, the robot simply detects a red bar of a rope ladder using point cloud acquired by a depth sensor attached to the robot's head. We show the detail of the detection algorithm below.

1) Extract red points from point cloud near the robot's hands.
2) Apply euclidean clustering of PCL[10] to the extracted point cloud.
3) For each cluster, connect the point which has the greastest y-component and one which has the smallest y-component in the global coordinate system, and regard the line segment formed by the two points as a bar of a rope ladder.

This method assumes that a bar of a rope ladder is straight and that the bar is roughly parallel to y-axis of the global coordinate system. These conditions are satisfied in only limited situation, so it is required to replace this recognition with more universal one as future task.

E. Reference Rope Tension Search Based on Current State

For search algorithm of reference rope tension, it is necessary to determine the angle of a slope and the direction of a rope. However, the assumed direction of a rope doesn't

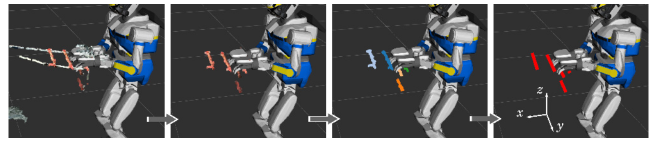

Fig. 12. Each Phase of Rope Ladder Recognition

always match the real direction. Therefore, we make a robot stand on a steep slope with initial reference rope tension determined by the first search based on the initial assumed parameters, then we execute the reference rope tension search again based on current state of the robot. The real direction of a rope is estimated by output of force sensors attached to the robot's hands as follows.

$$\boldsymbol{u}_{rope} = \frac{\boldsymbol{f}_{rh} + \boldsymbol{f}_{lh}}{|\boldsymbol{f}_{rh} + \boldsymbol{f}_{lh}|} \tag{18}$$

The robot executes the search described in Sec. IV based on the estimated direction of the rope and current positions of the robot's hands and legs, then it updates reference forces of its hands.

F. Experiment with HRP-2

With our proposed methods, we conducted experiments of walking on a steep slope using a rope by a life-size humanoid robot. The situation of the experiment is shown in Fig.I. We used HRP2[11] as a life-size humanoid robot, which weighs 60kg. We conducted the experiments on a 20 degrees slope and a 40 degrees slope.

The experiment starts from the state in which the robot grasping one side of a rope with both hands stands on the slope. The other side of the rope is fixed to a wall. The static friction coefficient between the slope and the robot's foot is about 0.72, which is smaller than $\tan 40°$. It means that the robot cannot stand on the 40 degrees slope statically without any support because of slip. In addition, the robot accelerates along the slope while walking, then there is possibility of slip during walking even if it can stand on the slope statically by pulling a rope because our method doesn't consider dynamic stability including acceleration of CoM. Therefore, we used 0.6 as the static friction coefficient for the search executed on the slope. By setting the static friction coefficient smaller than the real value, it is expected that friction constraint between foot and a slope remains satisfied when acceleration of CoM caused by walk occurs. We set the rope so that its direction roughly matched the front direction of the robot.

The results of the reference rope tension search are as shown in Fig.6 $\sim$ Fig.9 in Sec. IV when the direction of a rope is $(1, 0, 0)$ in the global coordinate system and static friction coefficient is 0.7. According to these results, we set initial reference force of each hand $(28, 0, 0)$ N for the 20 degrees slope and $(44, 0, 0)$ N for the 40 degrees slope in the global coordinate system. After the robot stands on the slope statically with this initial setting of the reference forces, it updates the reference forces based on its current state by our proposed search algorithm before it starts walking. Then

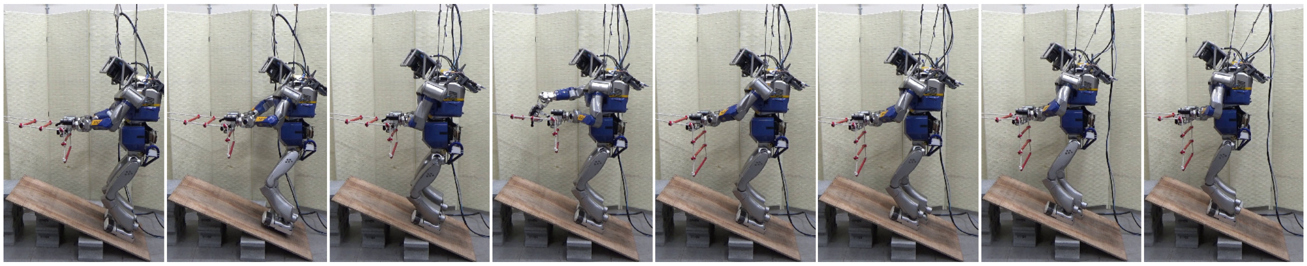

Fig. 13. HRP2 walking on a 20 degrees slope using a rope

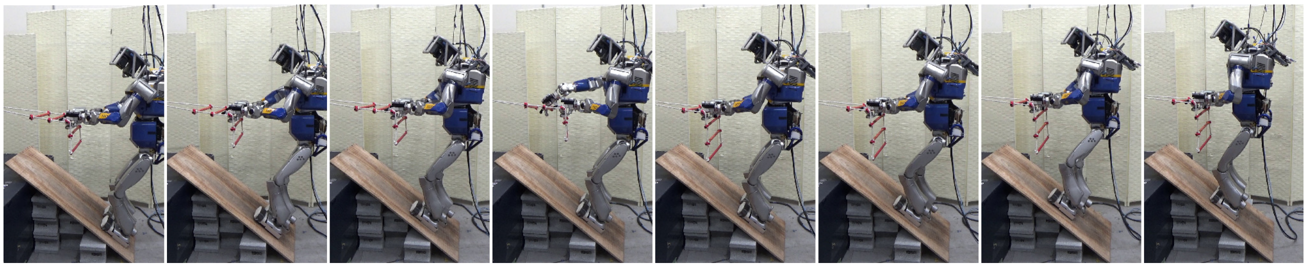

Fig. 14. HRP2 walking on a 40 degrees slope using a rope

it walks up on the steep slope and grasps the next bar of the rope ladder. The robot updates the reference forces again after it has grasped the new bar, and it walks up again. The other parameters are shown in Table I.

We show the snapshots of each experiment and graphs of tension of the rope in Fig.13 $\sim$ Fig.16. Because of impedance control to keep the tension, the tension of the rope exerted to the robot's hands was realized with an error within 5N from the reference tension. From $T = 80$s to $T = 140$s in Fig.15 and Fig.16, the actual forces of the robot's hands departed from the reference values because impedance control was disabled to grasp the next bar of the rope ladder during this period. With these experiments, we confirmed that the robot using a rope could walk on the slope on which the robot cannot stand without any support. In addition, the robot standing on a slope could successfully grasp the next bar of the rope ladder by simple recognition of the rope ladder.

In the experiment on a 20 degrees slope, the reference forces of the robot's hands decreased after twice reference rope tension search. It is because the robot can walk on a 20 degrees slope without any support in the first place, and the search algorithm selected to decrease reference forces based on the current state of the robot to minimize torque exerted to the robot's joints.

On the other hand, the search algorithm set the reference forces about 40N $\sim$ 55N in the experiment on a 40 degrees slope. The reference force of the right hand and the one of the left hand were not equal because the position of the rope ladder was not accurately center of the robot. With the determined parameters, the robot successfully walked up on the 40 degrees slope without slipping.

TABLE I

PARAMETERS OF THE EXPERIMENTS

Angle of slope	Direction of rope	Initial reference force	Length of stride	Time for one step	Step number
20°	(1, 0, 0)	28N	10cm	4s	8
40°	(1, 0, 0)	44N	10cm	4s	8

TABLE II

RESULT OF REFERENCE ROPE TENSION SEARCH FOR 20 DEGREES

	First Search	Second Search
Ref force (right hand)	(13.2, 0.38, 0.69)N	(4.9, 0.17, -0.21)N
Ref force (left hand)	(10.9, 0.32, 0.57)N	(3.4, 0.12, -0.04)N
Estimated rope direction	(0.998, 0.03, 0.05)	(0.998, 0.04, -0.04)
Max torque ratio	0.15	0.17
Range of motion of foot	210 mm	200 mm

TABLE III

RESULT OF REFERENCE ROPE TENSION SEARCH FOR 40 DEGREES

	First Search	Second Search
Ref force (right hand)	(55.5, 2.4, 4.4)N	(43.2, 1.5, 2.9)N
Ref force (left hand)	(40.2, 1.7, 3.2)N	(50.4, 1.8, 3.4)N
Estimated rope direction	(0.996, 0.04, 0.08)	(0.997, 0.04, 0.07)
Max torque ratio	0.23	0.21
Range of motion of foot	100 mm	100 mm

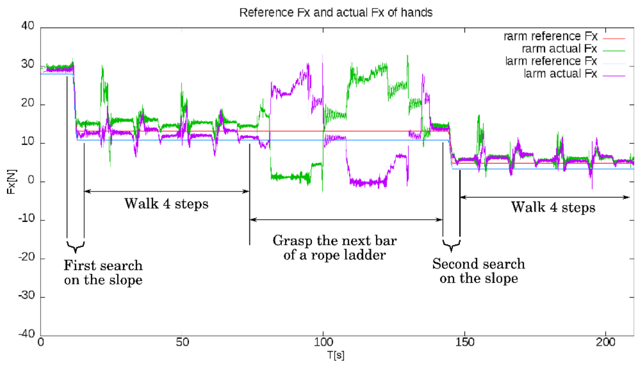

Fig. 15. Reference and actual tension exerted to the hands (20 degrees)

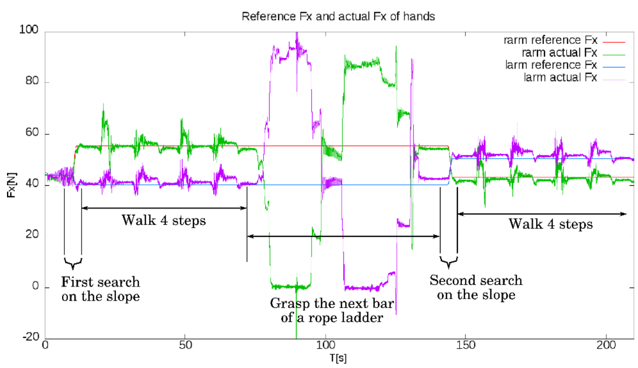

Fig. 16. Reference and actual tension exerted to the hands (40 degrees)

VI. CONCLUSION

A. Achievements

The achievements of this paper can be described as the following.

- The method to check the stability of a robot grasping a rope on a slope.
- The algorithm to search proper reference tension of a rope considering torque and range of motion of the robot's legs.

We conducted experiments on a 20 degrees slope and a 40 degrees slope by a life-size humanoid robot. Especially, we confirmed that the robot using a rope could walk on the 40 degrees slope which is originally hard to climb without a rope. It shows the usefulness of use of a rope for walking on a steep slope.

B. Future Work

In this paper, we assumed that the angle and the static friction coefficient of a slope are known. However, the situation in which such parameters are known is rare, so a robot has to recognize the state of a slope for more universal walking on a slope. For example, detection of slip during walking on a slope will be required to estimate static friction coefficient. In addition, our method considers only static stability on a slope, so there is possibility of slipping caused by acceleration of walking. In order to keep dynamic stability without slipping, it is necessary to calculate and update reference tension of a rope in real time during walking.

In the experiments, the robot started walking from the state in which it stood on the slope, but it has to get on a slope by itself when it works in real environment. With our current method, the robot standing on the ground couldn't get on the 40 degrees slope because its foot couldn't reach the slope from the state in which it was pulling a rope strongly. The robot requires a long stride to get on a slope from the ground, but it cannot move its foot forward enough because its hip position is behind its feet. To solve this problem, we have to improve our method so that the robot can shift smoothly from the state in which it stands on the ground without pulling a rope to the state in which it stands on a steep slope by pulling a rope.

REFERENCES

[1] J. Vaillant, A. Kheddar, H. Audren, F. Keith, S. Brossette, K. Kaneko, M. Morisawa, E. Yoshida, and F. Kanehiro. Vertical ladder climbing by the hrp-2 humanoid robot. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pp. 671–676, Nov 2014.

[2] M. Kanazawa, S. Nozawa, Y. Kakiuchi, Y. Kanemoto, M. Kuroda, K. Okada, M. Inaba, and T. Yoshiike. Robust vertical ladder climbing and transitioning between ladder and catwalk for humanoid robots. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2202–2209, Sept 2015.

[3] M. Kudruss, M. Naveau, O. Stasse, N. Mansard, C. Kirches, P. Soueres, and K. Mombaur. Optimal control for whole-body motion generation using center-of-mass dynamics for predefined multi-contact configurations. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pp. 684–689, Nov 2015.

[4] Weiwei Huang, Chee-Meng Chew, Yu Zheng, and Geok-Soon Hong. Pattern generation for bipedal walking on slopes and stairs. In *Humanoids 2008 - 8th IEEE-RAS International Conference on Humanoid Robots*, pp. 205–210, Dec 2008.

[5] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, Vol. 2, pp. 1620–1626 vol.2, Sept 2003.

[6] S. Kajita, M. Morisawa, K. Miura, S. Nakaoka, K. Harada, K. Kaneko, F. Kanehiro, and K. Yokoi. Biped walking stabilization based on linear inverted pendulum tracking. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4489–4496, Oct 2010.

[7] S. Nozawa, I. Kumagai, Y. Kakiuchi, K. Okada, and M. Inaba. Humanoid full-body controller adapting constraints in structured objects through updating task-level reference force. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3417–3424, Oct 2012.

[8] T. Doi, R. Hodoshima, S. Hirose, Y. Fukuda, T. Okamoto, and J. Mori. Development of a quadruped walking robot to work on steep slopes, titan xi (walking motion with compensation for compliance). In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2067–2072, Aug 2005.

[9] H.J. Ferreau, C. Kirches, A. Potschka, H.G. Bock, and M. Diehl. qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, Vol. 6, No. 4, pp. 327–363, 2014.

[10] R. B. Rusu and S. Cousins. 3d is here: Point cloud library (pcl). In *2011 IEEE International Conference on Robotics and Automation*, pp. 1–4, May 2011.

[11] K. Okada, T. Ogura, A. Haneda, J. Fujimoto, F. Gravot, and M. Inaba. Humanoid motion generation system on hrp2-jsk for daily life environment. In *IEEE International Conference Mechatronics and Automation, 2005*, Vol. 4, pp. 1772–1777 Vol. 4, July 2005.