# Humanoid Loco-Manipulation Planning Based on Graph Search and Reachability Maps

Masaki Murooka, Iori Kumagai, Mitsuharu Morisawa, Fumio Kanehiro,
Abderrahmane Kheddar

# Humanoid Loco-manipulation Planning based on Graph Search and Reachability Maps

Masaki Murooka, Iori Kumagai, Mitsuharu Morisawa, Fumio Kanehiro, and Abderrahmane Kheddar

*Abstract*—In this letter, we propose an efficient and highly versatile loco-manipulation planning for humanoid robots. Loco-manipulation planning is a key technological brick enabling humanoid robots to autonomously perform object transportation by manipulating them. We formulate planning of the alternation and sequencing of footsteps and grasps as a graph search problem with a new transition model that allows for a flexible representation of loco-manipulation. Our transition model is quickly evaluated by relocating and switching the reachability maps depending on the motion of both the robot and object. We evaluate our approach by applying it to loco-manipulation use-cases, such as a bobbin rolling operation with regrasping, where the motion is automatically planned by our framework.

*Index Terms*—Humanoid and Bipedal Locomotion; Manipulation Planning; Multi-Contact Whole-Body Motion Planning and Control.



Fig. 1. Humanoid loco-manipulation motions.

## I. INTRODUCTION

**M**OVING large objects is a typical task required for humanoid robots in large-scale manufacturing environments. As most of such objects are heavy, they need to be moved through manipulating them by taking advantage of the ground and any possible inertia properties. Therefore, to accomplish such a task, robots must autonomously plan loco-manipulation motion; that is to say, a multi-contact motion that alternates or simultaneously performs bipedal locomotion and object manipulation.

In this letter, we propose a versatile planning framework for loco-manipulation that has the following features: (i) determining the robot and object motion while considering obstacle avoidance, (ii) stepping a foot while moving an object, (iii) (re)grasping an object when necessary, and finally (iv) treating objects in sliding or rolling motion in a unified manner. Several studies of humanoid loco-manipulation exist, yet none of them has all of these features integrated [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12].

The key novelty of our framework is a transition model that can represent the complicated combinations of locomotion and manipulation primitives. This transition model can be evaluated efficiently while considering the movement of the grasping point due to the object motion by switching the reachability maps. We show that various loco-manipulation behaviors as shown in Fig. 1 are generated quickly and flexibly
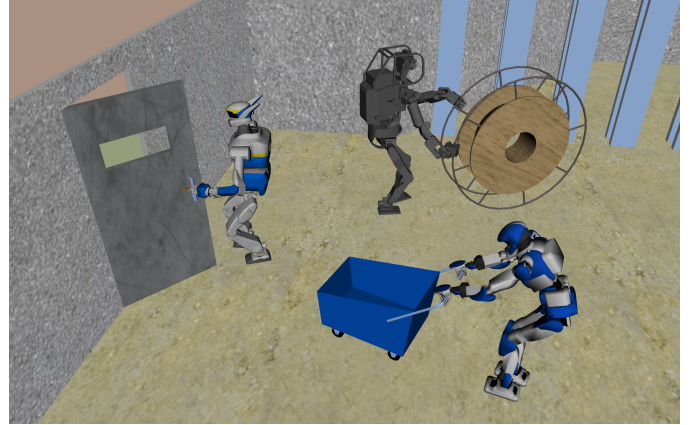
by applying this transition model to a sophisticated graph search algorithm. It is noteworthy that, to the best of our knowledge, this is the first study in which a humanoid robot automatically plans the motion of moving a large cylindrical object by the rolling operation with regrasping.

### A. Related Works

*1) Large Object Manipulation:* Various types of large object manipulation are achieved by life-size humanoid robots [1]. When an object is lifted and carried, only the bipedal footsteps are usually planned as the object can be moved in an arbitrary direction [2]. In non-prehensile manipulation such as pushing [3] and pivoting [4] and articulated environment operation such as door opening [5], it is necessary to consider the kinematic constraints of the object due to the ground contact or the joint. In many of these works, given a specific operation, ad-hoc rules (e.g., specifying the robot position with respect to the object) are applied [3], [4], [5], that cannot deal with the situation where the relative position of the robot and the object need to be changed to avoid obstacles. Our loco-manipulation planning method can cope with such situations and even with situations where object regrasping is necessary.

*2) Footstep and Multi-contact Planning:* In humanoid bipedal walking, left and right foot is switched alternately, so graph search that can handle discrete state transitions is often used [13], [14]. In a complicated environment, anytime graph search algorithms [15] are used that quickly acquire an initial suboptimal solution and gradually converge the solutions to the optimal one [16]. In multi-contact motion, which is a more flexible style of behavior, sampling-based and graph search-based methods are used widely [17], [18]. As a contrasting approach from these, optimization-based planning of bipedal
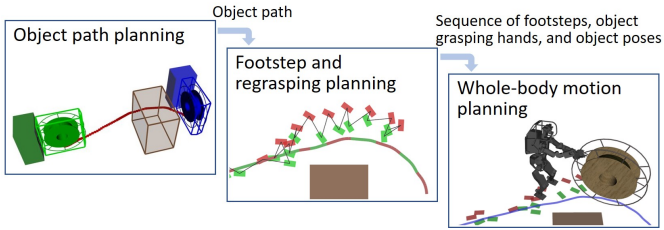
Fig. 2. Overview of the planning framework.
The loco-manipulation planning framework consists of three processes: object path planning (OP-planning), footstep and regrasping planning (FR-planning), and whole-body motion planning (WBM-planning).
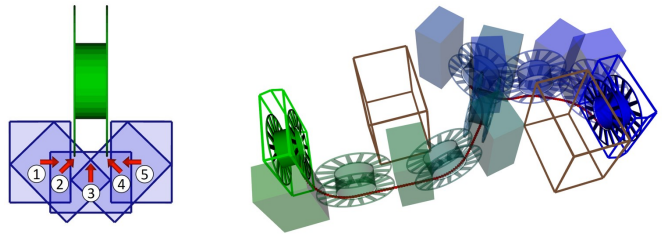


Fig. 3. An example of models and results in OP-planning.
The shape of the robot is approximated by a bounding box. Five candidates for the robot (in its bounding box) pose relative to the object are shown as an example. The robot pose is discretized and represented as the index of the candidates.

and multi-contact motion has been proposed recently with the advantage of being able to handle motion without discretization [19], [20]. However, in this letter, graph search is used to incorporate object motion-dependent reachability maps in a concise and low computational cost manner. We extend the transition model from the bipedal one [13], [14] so that motions of both robot and object are considered simultaneously. By limiting the target locomotion form to bipedal walking, loco-manipulation planning is appropriately divided and an efficient transition model is proposed.

*3) Loco-manipulation Planning:* The motions covered by the previous works concerning humanoid loco-manipulation can roughly be divided into two categories: the motions that constrain the trajectory of the hands or the manipulated object [6], [7] and those that do not [8], [9]. The former motions, which are the focus of this paper, are required for tasks in which the object motion is explicitly constrained, such as opening a door and pushing an object along a trajectory, while the latter motions are useful for approaching and lifting objects. Sampling-based multi-modal planning has been proposed for motion that combines stepping and reaching [10], [11]. The motion along the given hand trajectory is planned by searching the footstep and object progression at the same time based on the loco-manipulation reachability acquired by data-driven learning [6]. Our approach is similar to this; however, by relocating and switching the pre-generated reachability maps [12], we propose a more general and faster method that can handle regrasping and object rolling.

### B. Contributions of this Letter

The contributions of our work are as follows: (i) a general loco-manipulation planning method using a flexible transition model with low evaluation cost, (ii) automatically generating a rolling operation with regrasping by switching the reachability map according to the object motion, (iii) showing that humanoid motions for various loco-manipulation tasks can be planned in a unified planning framework.

### II. METHOD OVERVIEW

To make loco-manipulation planning easier to treat, we divide it into three processes as shown in Fig. 2: object path planning (OP-planning), footstep and regrasping planning (FR-planning), and whole-body motion planning (WBM-planning).

In the following, OP-planning, FR-planning, and WBM-planning are presented in Sections III, IV, and V, respectively.

Especially, Section IV provides a detailed explanation of the transition model that is the main novelty of this letter. Section VI shows the application to various loco-manipulation tasks.

### III. OBJECT PATH PLANNING

We use the asymptotically-optimal sampling-based method RRT* [21] to plan an object path from the start to the goal. The planning state space is a compound space consisting of the object pose and the robot pose because it is necessary to consider collision avoidance with the environment not only for the object but also for the robot. The object pose is represented as the $SE(2)$ space state. For objects with car-like nonholonomic motion constraints, the Reeds-Shepp curve [22] is used for the edges connecting the states. To reduce the dimensions of the state space, the robot pose is represented as the index of the pose candidates relative to the object (Fig. 3).

### IV. FOOTSTEP AND REGRASPING PLANNING

#### A. Problem Settings

Footstep and regrasping planning (FR-planning) for loco-manipulation is defined as the problem of finding a sequence of footsteps and object poses that makes a transition from the initial state to the goal state. The formulation is made to allow the transition of regrasping the object from one hand to the other one.

The following information (Fig. 4) about the robot and object is assumed to be known in FR-planning:

- $\mathcal{A}(l_{foot}) = \{\,\boldsymbol{a}[i]\,|\,i = 1, 2, \cdots, N_{\mathcal{A}}\,\}$ ($l_{foot} \in \{L, R\}$): a set of discretized footstep actions. $N_{\mathcal{A}}$ is the number of actions. $L$ and $R$ represent the left and right foot, respectively. Each action $\boldsymbol{a} \in SE(2)$ represents the stepping position relative to the opposite foot.
- $\mathcal{P} = \{\,\boldsymbol{c}_{\mathcal{P}}[i]\,|\,i = 1, 2, \cdots, N_{\mathcal{P}}\,\}$: a sequence of the object poses that represents the discretized object path. Object pose $\boldsymbol{c}_{\mathcal{P}} \in SE(2)$ is represented by projecting it onto the ground plane. $N_{\mathcal{P}}$ is the sequence length.
- $\mathcal{M}(\boldsymbol{c}_{com}, l_{hand}) \subset SE(2)$: a reachability map of the object. The elements in a reachability map are the object poses such that the hand $l_{hand} \in \{L, R\}$ can reach the object when the CoM frame of the robot is $\boldsymbol{c}_{com} \in SE(2)$.

The object path $\mathcal{P}$ is obtained from the previous process, OP-planning. The procedures for determining the footstep action
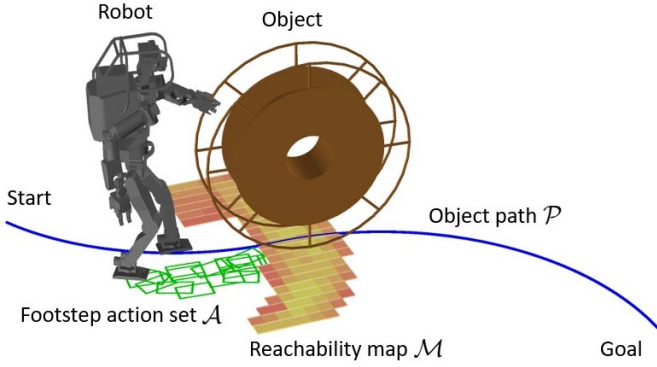
Fig. 4. Problem settings of FR-planning.
A robot is moving an object from the start position to the goal position. Green rectangle markers represent the footstep action set for the right foot; a blue line represents the object path; and markers with yellow to red gradation represent the reachability map.

set $\mathcal{A}$ and the reachability map $\mathcal{M}$ are described in Sections VI-B and IV-C, respectively.

### B. Formulation as Graph Search Problem

Since footstep and regrasping cause discrete state transitions, we formulate FR-planning as a graph path search problem that is suitable for treating discrete transition combinations. We focus on two improvements in graph search for FR-planning: (i) anytime approach that gradually improves the solution [15] and (ii) incremental replanning for graph updates [23]. We use the Anytime Dynamic A* (AD*) algorithm [24] that has these two features for FR-planning. The graph search algorithms introduced above are generalized so that a solution can be automatically obtained by giving a graph instance that represents the problem to be solved. Therefore, we do not explain the details of the graph search algorithm itself, see [15], [23], [24] and the source code [25]. We rather focus on graph components specific to FR-planning in the following.

*1) State:* In FR-planning, each graph node represents the state defined as follows:

$$\boldsymbol{s} = (\boldsymbol{c}_{st\text{-}foot}, l_{st\text{-}foot}, \boldsymbol{c}_{sw\text{-}foot}, l_{sw\text{-}foot}, \boldsymbol{c}_{obj}, l_{hand}) \quad (1)$$
$$\text{where} \quad \boldsymbol{c}_{st\text{-}foot}, \boldsymbol{c}_{sw\text{-}foot}, \boldsymbol{c}_{obj} \in SE(2)$$
$$l_{st\text{-}foot}, l_{sw\text{-}foot}, l_{hand} \in \{L, R\}$$
$$l_{sw\text{-}foot} = \bar{l}_{st\text{-}foot}$$

where $\boldsymbol{c}_*$ is the pose of the foot and object, and $l_*$ is the left/right label of the foot and the hand. *st-foot* and *sw-foot* represent a stance foot and a swing foot, respectively. *hand* represents a hand that grasps an object. $\bar{l}_*$ represents the opposite label (i.e., $\bar{L} = R, \bar{R} = L$). Here, we assume that the object is grasped and manipulated with one hand except during regrasping, see Section VI-A for both hands case.

This state definition is extended from the bipedal footstep planning where $\boldsymbol{s}$ is $(\boldsymbol{c}_{sw\text{-}foot}, l_{sw\text{-}foot})$ [13], [14] and the state-of-the-art loco-manipulation planning where $\boldsymbol{s}$ is $(\boldsymbol{c}_{st\text{-}foot}, l_{st\text{-}foot}, \boldsymbol{c}_{sw\text{-}foot}, l_{sw\text{-}foot}, \boldsymbol{c}_{obj})$ ($l_{hand}$ is missing) [6]. The state includes the stance foot, which is unnecessary

for bipedal footstep planning, to evaluate the state transition feasibility as described in Section IV-C.

*2) Successors:* The condition that a state $\boldsymbol{s}^{[k+1]}$ is a successor of the state $\boldsymbol{s}^{[k]}$ is represented as follows:

$$l_{st\text{-}foot}^{[k+1]} = l_{sw\text{-}foot}^{[k]} \quad (2a)$$
$$l_{sw\text{-}foot}^{[k+1]} = l_{st\text{-}foot}^{[k]} \quad (2b)$$
$$l_{hand}^{[k+1]} \in \{L, R\} \quad (2c)$$
$$\boldsymbol{c}_{st\text{-}foot}^{[k+1]} = \boldsymbol{c}_{sw\text{-}foot}^{[k]} \quad (2d)$$
$$\boldsymbol{c}_{sw\text{-}foot}^{[k+1]} \in \left\{ F_{apply}(\boldsymbol{c}_{st\text{-}foot}^{[k+1]}, \boldsymbol{a}) \,\middle|\, \boldsymbol{a} \in \mathcal{A}(l_{sw\text{-}foot}^{[k+1]}) \right\} \quad (2e)$$
$$\boldsymbol{c}_{obj}^{[k+1]} \in \left\{ \boldsymbol{c}_{\mathcal{P}} \left[ idx(\boldsymbol{c}_{obj}^{[k]}) + i \right] \,\middle|\, i = 0, 1, \cdots, N_{obj} \right\} \quad (2f)$$
$$F_{switchable}(\ldots) = true \quad (2g)$$
$$F_{movable}(\ldots) = true \quad (2h)$$
$$F_{no\text{-}collision}(\ldots) = true \quad (2i)$$

The superscript indicates the index of the state and its elements (e.g., $l_{st\text{-}foot}^{[k]}$ is an element of $\boldsymbol{s}^{[k]}$). The arguments of the functions in (2g)-(2i) are omitted.

The first three expressions (2a)-(2c) are the transitions of the left/right label of the foot and the hand. For the foot, the labels of the stance foot and the swing foot are interchanged, and for the hand, the transition to both the left and right is allowed. When $l_{hand}^{[k]}$ and $l_{hand}^{[k+1]}$ are the same, it means to keep grasping the object with the same hand; otherwise it means regrasping from one hand to the other one.

Expressions (2d) and (2e) are the transitions of the foot poses shown in Fig. 5 (A). Since the last swing foot becomes the new stance foot, the pose of the new stance foot ($\boldsymbol{c}_{st\text{-}foot}^{[k+1]}$) is the same as the landing pose of the last swing foot ($\boldsymbol{c}_{sw\text{-}foot}^{[k]}$). The landing pose of the new swing foot ($\boldsymbol{c}_{sw\text{-}foot}^{[k+1]}$) is generated from the footstep action set $\mathcal{A}$. $F_{apply}(\boldsymbol{c}_{st\text{-}foot}, \boldsymbol{a})$ returns the swing foot pose after applying the footstep action $\boldsymbol{a}$ when the pose of the opposite foot (i.e., the stance foot) is $\boldsymbol{c}_{st\text{-}foot}$; recall that the footstep action represents the stepping position relative to the opposite foot.

Expression (2f) is the transition of the object pose shown in Fig. 5 (B). The object path $\mathcal{P}$ is discretized, and the index is managed so that the object moves toward the goal by increasing it. According to the increment from the last index, successors representing the corresponding object pose are generated. $idx(\boldsymbol{c}_{obj})$ represents an index $i$ such that $\boldsymbol{c}_{\mathcal{P}}[i] = \boldsymbol{c}_{obj}$. $N_{obj}$ is the incremental limit of the index in one transition.

Expression (2g) is the condition for switching hands for regrasping, and expression (2h) is the condition for moving the object along the path. The detailed process for evaluating $F_{switchable}$ and $F_{movable}$ based on a reachability map is described in Section IV-C.

Expression (2i) triggers environment's obstacles avoidance. In FR-planning, geometric collision is detected on the basis of bounding box approximation of shape model.

State (1) and state transition (2) are general enough to flexibly capture loco-manipulation including regrasping. Yet, pure manipulation can occur since the footstep action set
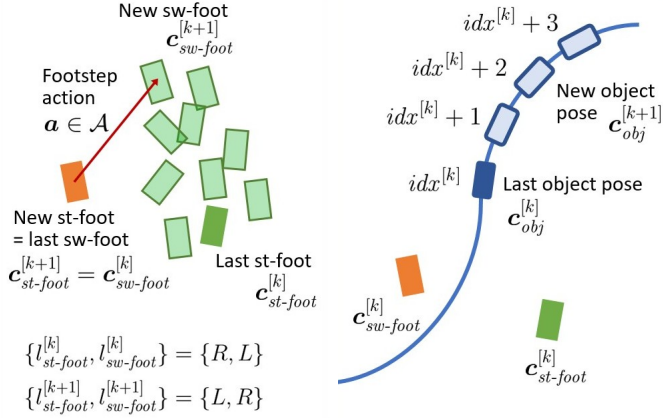
(A) Footstep transition (2d) and (2e).　(B) Object pose transition (2f).

Fig. 5. Transitions in state successors of FR-planning.
Orange markers represent the left foot; green markers represent the right foot; and blue markers represent the object. $idx^{[k]}$ is an abbreviation for $idx(c_{obj}^{[k]})$.

includes no-stepping in (2e); and pure locomotion is possible since zero is included in the index increments of the object pose in (2f).

*3) Start and Goal:* The start state $s^{[0]}$ and goal states $s^{[G]}$ are defined as follows:

$$c_{obj}^{[0]} = c_{\mathcal{P}}[1] \tag{3a}$$

$$c_{st\text{-}foot}^{[0]}, c_{sw\text{-}foot}^{[0]} = initial\ foot\ poses \tag{3b}$$

$$\{l_{st\text{-}foot}^{[0]}, l_{sw\text{-}foot}^{[0]}, l_{hand}^{[0]}\} = \{L, R, L\} \tag{3c}$$

$$c_{obj}^{[G]} = c_{\mathcal{P}}[N_{\mathcal{P}}] \tag{3d}$$

The initial foot poses are known. Left/right labels in the start state are given temporarily, and, if necessary, they will be swapped in successors. In the goal states, any pose and label are allowed for the foot and the hand.

*4) State Transition Cost:* The cost to transit from $s^{[k]}$ to $s^{[k+1]}$ is represented as follows:

$$c(s^{[k]}, s^{[k+1]}) = d(c_{obj}^{[k]}, c_{obj}^{[k+1]}) + c_{step} + c_{regrasp} \tag{4}$$

$d(c_{obj}^{[k]}, c_{obj}^{[k+1]})$ represents the distance along the object path. $c_{step}$ and $c_{regrasp}$ are the constant costs that are added only when stepping a foot and regrasping an object, respectively.

*5) Heuristic Function:* The AD* algorithm efficiently finds a solution by prioritizing searches based on heuristics. The heuristic function of the state $s^{[k]}$ is represented as follows:

$$h(s^{[k]}) = d(c_{obj}^{[k]}, c_{obj}^{[G]}) + N_{step}c_{step} + h_{nominal} \tag{5}$$

$N_{step}$ is the minimum number of footsteps from the current state to the goal state, calculated from the maximum stride length of the footstep action set $\mathcal{A}$. $h_{nominal}$ is a heuristic value according to the distance between the current foot poses and the nominal foot poses calculated from the current object pose. Without $h_{nominal}$, the heuristic is admissible [15]. The details and effectiveness of $h_{nominal}$ are described in Section VI-B.
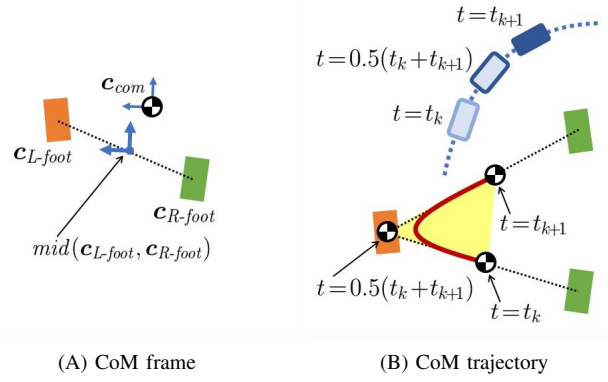


(A) CoM frame　　　(B) CoM trajectory

Fig. 6. Frame and trajectory of CoM
(A) $mid(c_{L\text{-}foot}, c_{R\text{-}foot})$ is the middle pose between the left and right foot poses. The position of CoM frame $c_{com}$ coincides with the robot's CoM, and its orientation coincides with the middle pose.
(B) The red curve illustrates an example of the CoM trajectory when stepping the right foot. To simplify the movability evaluation, the assumptions shown here are imposed on the poses of the robot's CoM and the object in the middle timing of the transition.

*C. Transition Evaluation based on Reachability Map*

*1) Reachability Map Generation:* To avoid repetitive calculation of computationally expensive whole-body inverse kinematics, a pre-computed reachability map is used for the feasibility evaluation of the transition. The reachability map $\mathcal{M}(c_{com}, l_{hand})$ is represented as the set of object poses relative to the robot's CoM frame defined as in Fig. 6 (A) to evaluate the reachability regardless of the foot poses.

Fig. 7 shows an example of the reachability map. By dividing the space of the object pose ($SE(2)$) into a grid and solving the inverse kinematics for each object pose $c_{obj}$, an object region is derived where the robot can reach the hand $l_{hand}$ to the grasping point $T_{hand}$ while keeping the CoM frame in the target $c_{com}$. The inverse kinematics calculation is the same as that in the WBM-planning described in Section V. The transformation from the object pose $c_{obj}$ to the grasping point $T_{hand}$ is treated as known information.

*2) Switchability Evaluation:* The switchable condition $F_{switchable}$ in (2g) depends on the last state of the feet and the object ($l_{st\text{-}foot}^{[k]}, c_{st\text{-}foot}^{[k]}, l_{sw\text{-}foot}^{[k]}, c_{sw\text{-}foot}^{[k]}, c_{obj}^{[k]}$) and the transition of hand label ($l_{hand}^{[k]}, l_{hand}^{[k+1]}$). If $l_{hand}^{[k]}$ and $l_{hand}^{[k+1]}$ are the same, it is always satisfied. Otherwise, when both feet and the object are fixed to the last poses and the hand can be switched from $l_{hand}^{[k]}$ to $l_{hand}^{[k+1]}$, the switchability is satisfied (Fig. 8 (A)).

This condition is formulated using the reachability map as follows:

$$
\begin{aligned}
F_{switchable}(\dots) = \\
c_{obj}^{[k]} \in \mathcal{M}\left(mid(c_{st\text{-}foot}^{[k]}, c_{sw\text{-}foot}^{[k]}), l_{hand}^{[k]}\right) \\
\wedge\ c_{obj}^{[k]} \in \mathcal{M}\left(mid(c_{st\text{-}foot}^{[k]}, c_{sw\text{-}foot}^{[k]}), l_{hand}^{[k+1]}\right)
\end{aligned} \tag{6}
$$

where $mid(c_1, c_2)$ represents the middle pose between $c_1$ and $c_2$. During regrasping, the robot's CoM is assumed to be on the middle of both feet. The first evaluation of the reachability map in (6) corresponds to (sw1) in Fig. 8 (A), and the second corresponds to (sw2).
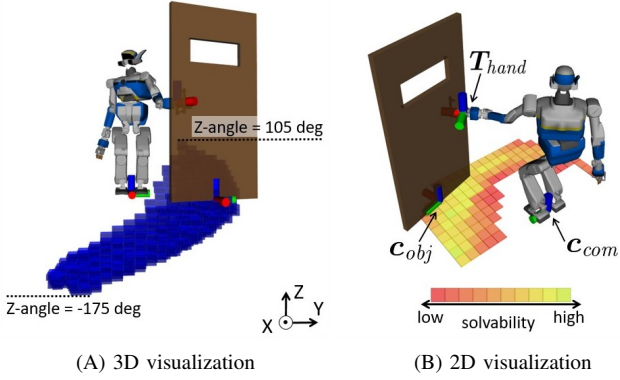
(A) 3D visualization      (B) 2D visualization

Fig. 7. An example of reachability map.
A reachability map for opening the door by grasping the doorknob with the left hand is shown. A reachability map is represented as a set of cells on a 3D grid consisting of the X and Y positions and the Z-angle (i.e., rotation angle around the Z-axis) of an object.
(A) The map is visualized so that the Z-position of the cell corresponds to the Z-angle of the object. It can be seen that the reachable Z-angle changes depending on the Y-position of the object.
(B) The reachable Z-angle information of the object is compressed into the cell color. The color of each cell represents the solvability (i.e., the number of reachable Z-angle cells) when the object is in the corresponding X and Y positions.



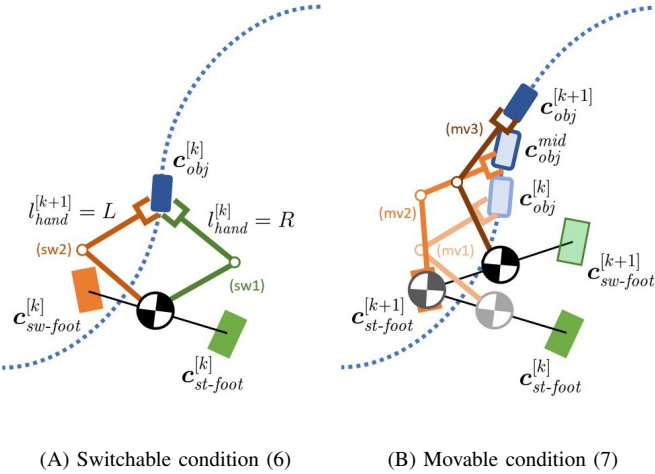(A) Switchable condition (6)      (B) Movable condition (7)

Fig. 8. Transition evaluation based on reachability map.
(A) Switchability when regrasping an object from the right hand to the left hand is evaluated.
(B) Movability of an object with stepping the right foot is evaluated. (mv1), (mv2), and (mv3) correspond to $t = t_k$, $t = 0.5(t_k + t_{k+1})$, and $t = t_{k+1}$ in Fig. 6 (B), respectively.

*3) Movability Evaluation:* The movable condition $F_{movable}$ in (2h) depends on the new state of the stance foot and the hand ($l_{st\text{-}foot}^{[k+1]}, c_{st\text{-}foot}^{[k+1]}, l_{hand}^{[k+1]}$), the transition of the swing foot ($l_{sw\text{-}foot}^{[k+1]}, c_{st\text{-}foot}^{[k]}, c_{sw\text{-}foot}^{[k+1]}$), and the transition of the object ($c_{obj}^{[k]}, c_{obj}^{[k+1]}$). It is satisfied when the object can be manipulated from $c_{obj}^{[k]}$ to $c_{obj}^{[k+1]}$ by the hand $l_{hand}^{[k+1]}$ while fixing the foot $l_{st\text{-}foot}^{[k+1]}$ to $c_{st\text{-}foot}^{[k+1]}$ and stepping the foot $l_{sw\text{-}foot}^{[k+1]}$ from $c_{st\text{-}foot}^{[k]}$ to $c_{sw\text{-}foot}^{[k+1]}$ (Fig. 8 (B)).

In [6], the movable condition is obtained by generating a dataset in the loco-manipulation motion space and training a neural network. However, due to the large dimensions of

the loco-manipulation motion space, the computational cost of dataset generation is large, even if it is calculated offline.

In this study, by making an assumption about the motions of the robot's CoM and the object, the movable condition is derived based on the reachability map evaluations. Fig. 6 (B) shows these motions during the loco-manipulation transition. The CoM trajectory depends on the motion properties (e.g., the CoM height and the stepping duration) and the generation algorithm (e.g., [26], [27]); however, in general, it can be approximated as being inside a triangular area connecting the stance foot and the CoM before and after the transition (visualized in yellow in Fig. 6 (B)). As the most severe case in terms of reachability, we assume the CoM is on the stance foot at the middle timing of the transition; it corresponds to $t = 0.5(t_k + t_{k+1})$ in Fig. 6 (B). Assuming that the CoM and the object move synchronously, the object is considered to be in the middle pose of the transition. Based on these considerations, the movable condition is formulated as follows:

$$
\begin{aligned}
F_{movable}(\dots) = \\
& c_{obj}^{mid} \in \mathcal{M}\left(c_{st\text{-}foot}^{[k+1]}, l_{hand}^{[k+1]}\right) \\
& \wedge \quad c_{obj}^{[k+1]} \in \mathcal{M}\left(mid(c_{st\text{-}foot}^{[k+1]}, c_{sw\text{-}foot}^{[k+1]}), l_{hand}^{[k+1]}\right) \quad (7) \\
& \text{where} \quad c_{obj}^{mid} = c_{\mathcal{P}}\left[\left(idx(c_{obj}^{[k]}) + idx(c_{obj}^{[k+1]})\right) / 2\right]
\end{aligned}
$$

$c_{obj}^{mid}$ represents the object pose in the middle timing of the transition. The first evaluation of the reachability map in (7) corresponds to (mv2) in Fig. 8 (B), and the second corresponds to (mv3); (mv1) has already been checked in the last switchability evaluation.

*4) Extension of Transition Evaluation to Rolling Objects:* The problem with applying the transition evaluation based on the reachability map to a rolling object is that the reachability map is not constant because the transformation from the object pose $c_{obj}$ to the grasping point $T_{hand}$ changes according to the motion of the object. In this study, this problem is solved by generating multiple reachability maps according to the rolling angles of the object and switching them according to the moving distance of the object.

Fig. 9 shows an example of the reachability maps for a rolling object. The reachable area gradually changes according to the object moving distance converted from the object-rolling angle. The set of these maps is represented as $\mathcal{M}(c_{com}, l_{hand}, d)$ where $d$ is the distance the object has moved since the last regrasping. By extending state (1) to add an index of the object pose at the last regrasping and selecting the reachability map according to the distance the object has moved from there, the FR-planning can be applied to the rolling object.

## V. WHOLE-BODY MOTION PLANNING

The trajectory of whole-body joint position is generated by the QP-based inverse kinematics (IK) calculation [28]. The target of the IK consists of three types of tasks as shown in Fig. 10: $SE(3)$ pose, CoM, and posture. The $SE(3)$ pose tasks represent the target poses of the stance foot, swing foot, and hand grasping the object. The swing foot trajectory is
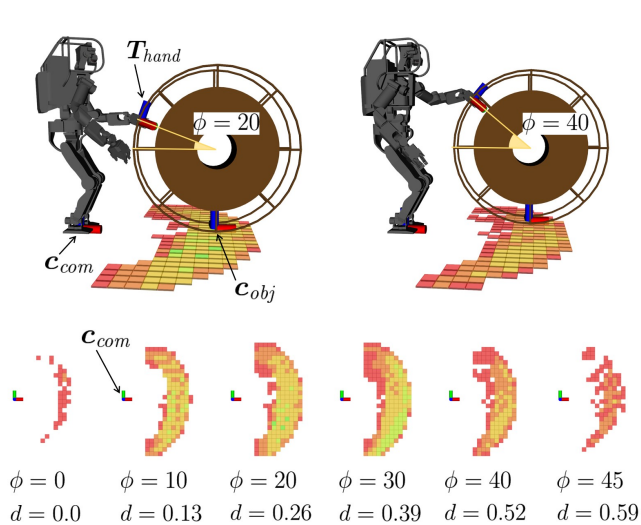
Fig. 9. An example of reachability maps for a rolling object.
For each sampled object-rolling angle $\phi$, a reachability map is generated for the corresponding grasping point $\boldsymbol{T}_{hand}$. The object moving distance $d$ is calculated from the rolling angle and the rolling radius. It can be seen that the reachable area gradually changes according to the object moving distance.
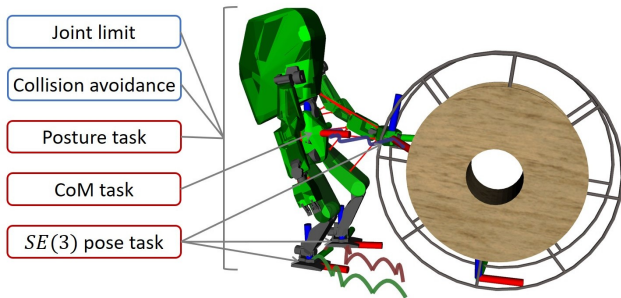


Fig. 10. Tasks and constraints in WBM-planning.
Whole-body IK consists of three types of equality tasks (boxed in red) and two types of inequality constraints (boxed in blue). Links checked for self-collision are displayed in their green strictly convex hulls.

generated by interpolating the foot poses before and after stepping and the midpoint of the 50 mm height with a cubic spline curve. In order to keep the torso link upright, the task of orientation in the roll and pitch directions is imposed with a small weight. The target CoM is generated by the preview control [26] from the ZMP trajectory determined from the planned footstep sequence. The task of half-sitting posture is imposed with a small weight for regularization. The joint position limits and self-collision avoidance are formulated as inequality constraints. The link shapes of the robot are represented by strictly convex hulls [29].

## VI. APPLICATION TO LOCO-MANIPULATION TASKS

The planning process was implemented in C++ and exchanges data via ROS communication [30]. As external libraries, OMPL [31] was used for RRT* [21] in OP-planning, and SBPL [25] was used for the AD* algorithm [24] in FR-planning.
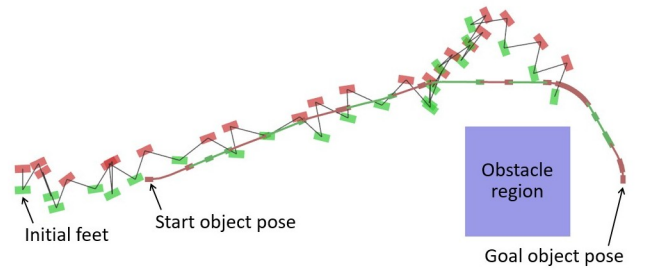


(A) Result of OP-planning and FR-planning (top view)
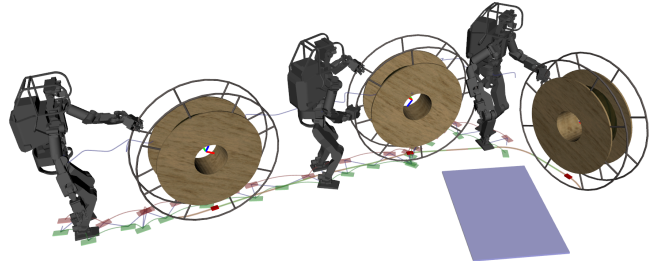


(B) Result of WBM-planning

Fig. 11. Bobbin rolling task.
HRP-5P humanoid carries a bobbin by rolling. The rectangle markers represent the footsteps of the left foot (red) and right foot (green). The sections of the object path that correspond to grasping with the left hand and right hand are drawn in red and green, respectively. The blue rectangle markers represent the obstacles.

### A. Planning Results of Loco-manipulation Motions

We applied our proposed planning method to three types of loco-manipulation tasks.

*1) Bobbin Rolling Task:* Fig. 11 shows the planning result of HRP-5P [32] moving-by-rolling a cable bobbin with a handle of 1.5 m in diameter and 0.3 m in width. Table I shows the computation time for each process. A detailed analysis of FR-planning is provided later.

TABLE I
COMPUTATION TIME OF LOCO-MANIPULATION PLANNING

| OP-planning | FR-planning | WBM-planning | Total |
|---|---|---|---|
| 0.51 s (0.10 s) | 1.15 s (0.07 s) | 3.90 s | 6.19 s |

The anytime algorithm is used for OP-planning and FR-planning, and the values in parentheses are the time until the initial solution is obtained.

*2) Door Opening Task:* Fig. 12 shows the planning result of HRP-2Kai opening a door. Without obstacle, a motion without regrasping is planned with a small cost of the graph path. Conversely, with an obstacle, the motion with regrasping from left to right hand is automatically planned to avoid the obstacle. In this way, the proposed planner generates appropriate motions according to the environment.

*3) Cart Pushing Task:* Fig. 13 shows the planning result of HRP-4 pushing a cart with both hands. By setting $l_{hand}$ in state (1) to always be $L \wedge R$ and using the intersection of the reachability maps of the left and right hands in the transition evaluation, the proposed planner handles easily both hands grasp manipulations.
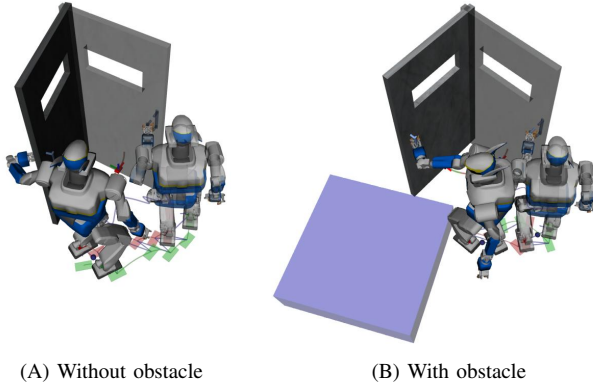
(A) Without obstacle  (B) With obstacle

Fig. 12. Door opening task.
In (A), HRP-2Kai humanoid opens the door with the left hand until the end, whereas in (B), the robot switches from left to right hand to avoid the obstacle visualized in blue.
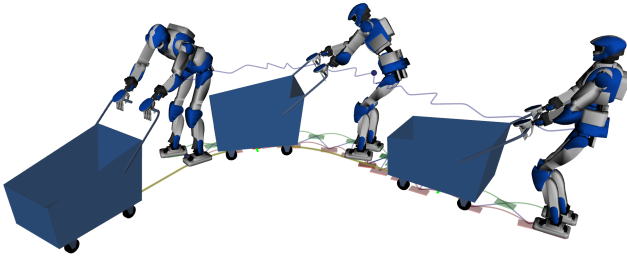


Fig. 13. Cart pushing task.
HRP-4 humanoid carries the cart by pushing it with both hands.

### B. Evaluation of Planning Results

*1) Computation Time:* Table II shows the detailed results of FR-planning in the examples described in Section VI-A. Compared with the state-of-the-art loco-manipulation planning [6], it can be seen that our planner can quickly generate complicated and long-term motions. In these planning examples, sufficiently high-quality final solutions are obtained within the computation time limit set to a few seconds. In addition, the initial solution is obtained in an extremely short time of approximately 100 ms; this is an important feature for the robot to continue moving without stopping during planning. The reachability maps are pre-generated with grid sizes of 100 mm and 10 degree; the computation time is approximately 300 s. For the rolling objects, multiple reachability maps are generated for object-rolling angles from 0 degree to 45 degree in 5 degree increments.

*2) Footstep Actions:* As mentioned in Section IV-A, in FR-planning, footstep action set $\mathcal{A}$ is prepared in advance. We use the Halton sequence [33], which has deterministic and quasirandom properties, to generate the footstep action set (Fig. 14 (A)). Fig. 14 (B) shows the relationship between the number of actions $N_{\mathcal{A}}$ and the length of the footstep sequence planned in Section VI-A1. The plan is robust regardless of the number of actions; however, it can be seen that a high-quality initial solution is obtained when the number of actions is large.

*3) Heuristic Function:* We evaluate the effectiveness of $h_{nominal}$ in the heuristic function (5) for the planning use-

### TABLE II
### DETAILED RESULTS OF FR-PLANNING

|  |  | (a) | (b) | (c) | (d) | (e) |
|---|---|---|---|---|---|---|
| Bobbin | Initial sol. | 0.07 s | 82.00 | 11.63 | 51 | 6524 |
|  | Final sol. | 1.15 s | 3.60 | 11.02 | 45 | 45673 |
| Door w/o obstacle | Initial sol. | 0.04 s | 40.20 | 3.02 | 13 | 2634 |
|  | Final sol. | 0.47 s | 3.60 | 2.01 | 13 | 21727 |
| Door w/ obstacle | Initial sol. | 0.04 s | 75.80 | 3.02 | 13 | 2634 |
|  | Final sol. | 0.05 s | 7.20 | 3.02 | 13 | 2669 |
| Cart | Initial sol. | 0.13 s | 81.20 | 7.58 | 41 | 9849 |
|  | Final sol. | 1.88 s | 1.80 | 6.18 | 29 | 74991 |

FR-planning performance for Section VI-A use-cases: (a) computation time, (b) heuristics inflation factor [24], (c) cost from start to goal, (d) length of the planned footstep sequence, (e) number of expanded states. The heuristics inflation factor qualifies the solution optimality: the closer it is to 1, the more optimal the solution is. The use-case of door with obstacle only expands a few more states in the final solution than the initial solution because the initial solution is already high-quality comparable to the final solution.



(A) Footstep action set  (B) Computation time and solution quality
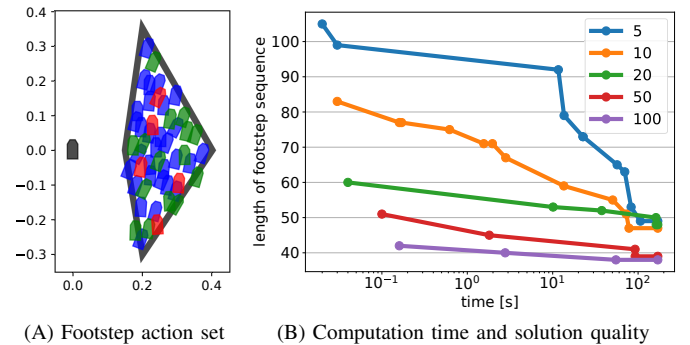
Fig. 14. Footstep actions in FR-planning.
(A) The black foot marker represents the stance foot, and the area surrounded by the black line represents the allowable area for the landing position of the swing foot. The red, green and blue foot markers represent the footstep actions generated by the Halton sequence (red: $1, \cdots, 5$; green: $6, \cdots, 20$; blue: $21, \cdots, 50$).
(B) The transition of the length of the planned footstep sequence is shown for the number of actions $N_{\mathcal{A}}$ of 5, 10, 20, 50, and 100, The smaller the length of the footstep sequence, the higher the quality of the solution. For large number of actions, it takes time to obtain the initial solution, yet its quality is high.

cases of Section VI-A1. The nominal pose of the foot is set to the pose 1.2 m behind the current object. Table III shows a comparison of the computation time until the initial solution is obtained with and without $h_{nominal}$. Especially for difficult problems such as moving a large object along a curved path, this heuristic avoids increasing the computation time by preferentially expanding the states leading to the goal.

### TABLE III
### COMPUTATION TIME WITH AND WITHOUT HEURISTICS

|  | straight path | curved path |
|---|---|---|
| w/ heuristics of nominal pose | 0.03 s | 0.07 s |
| w/o heuristics of nominal pose | 0.43 s | 17.41 s |

The time until the initial solution is obtained is shown. FR-planning was performed on straight and curved object paths.

*4) ZMP Evaluation:* Although the robot motions in Section VI-A were only animated with the visualization software Rviz [30], we checked the dynamics validity of the motion based on ZMP. Fig. 15 shows the ZMP trajectory when the robot moves forward several steps while rolling the bobbin.
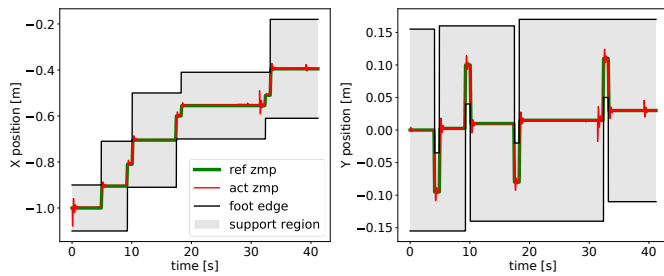
Fig. 15. ZMP trajectory of the planned motion.

The planned motion is dynamically feasible as the ZMP is within the support region.

## VII. Conclusion

In this letter, we proposed a versatile planning framework for loco-manipulation. In order to quickly and flexibly generate complex loco-manipulation motion involving object regrasping and obstacle avoidance, we introduced a transition model that can be evaluated efficiently based on reachability maps in a sophisticated graph search algorithm. We have shown the effectiveness of our planning framework by applying it to various loco-manipulation tasks such as rolling an object by a humanoid robot. Future challenges include improving parallelization of locomotion and manipulation, consideration of reaction forces from the object, and real-time replanning based on object tracking.

## References

[1] M. Murooka, S. Noda, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba, "Manipulation strategy decision and execution based on strategy proving operation for carrying large and heavy objects," in *IEEE International Conference on Robotics and Automation*, 2014, pp. 3425–3432.

[2] I. Kumagai, M. Morisawa, S. Nakaoka, T. Sakaguchi, H. Kaminaga, K. Kaneko, and F. Kanehiro, "Perception based locomotion system for a humanoid robot with adaptive footstep compensation under task constraints," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018, pp. 713–719.

[3] M. Stilman, K. Nishiwaki, and S. Kagami, "Humanoid teleoperation for whole body manipulation," in *IEEE International Conference on Robotics and Automation*, 2008, pp. 3175–3180.

[4] E. Yoshida, M. Poirier, J.-P. Laumond, O. Kanoun, F. Lamiraux, R. Alami, and K. Yokoi, "Pivoting based manipulation by a humanoid robot," *Autonomous Robots*, vol. 28, pp. 77–88, 2009.

[5] 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 3417–3424.

[6] S. J. Jorgensen, M. Vedantam, R. Gupta, H. Cappel, and L. Sentis, "Finding locomanipulation plans quickly in the locomotion constrained manifold," in *IEEE International Conference on Robotics and Automation*, 2020, pp. 6611–6617.

[7] S. Dalibard, A. Nakhaei, F. Lamiraux, and J. Laumond, "Manipulation of documented objects by a walking humanoid robot," in *IEEE-RAS International Conference on Humanoid Robots*, 2010, pp. 518–523.

[8] P. Ferrari, M. Cognetti, and G. Oriolo, "Humanoid whole-body planning for loco-manipulation tasks," in *IEEE International Conference on Robotics and Automation*, 2017, pp. 4741–4746.

[9] A. Settimi, D. Caporale, P. Kryczka, M. Ferrati, and L. Pallottino, "Motion primitive based random planning for loco-manipulation tasks," in *IEEE-RAS International Conference on Humanoid Robots*, 2016, pp. 1059–1066.

[10] K. Hauser and V. Ng-Thow-Hing, "Randomized multi-modal motion planning for a humanoid robot manipulation task," *The International Journal of Robotics Research*, vol. 30, no. 6, pp. 678–698, 2011.

[11] M. Kallmann, Y. Huang, and R. Backman, "A skill-based motion planning framework for humanoids," in *IEEE International Conference on Robotics and Automation*, 2010, pp. 2507–2514.

[12] N. Vahrenkamp, D. Berenson, T. Asfour, J. Kuffner, and R. Dillmann, "Humanoid motion planning for dual-arm manipulation and re-grasping tasks," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 2464–2470.

[13] J. Chestnutt, J. Kuffner, K. Nishiwaki, and S. Kagami, "Planning biped navigation strategies in complex environments," in *IEEE-RAS International Conference on Humanoid Robots*, 2003.

[14] N. Perrin, "Biped footstep planning," in *Humanoid Robotics: A Reference*. Springer, 2018, pp. 1–21.

[15] M. Likhachev, G. J. Gordon, and S. Thrun, "ARA*: Anytime A* with provable bounds on sub-optimality," in *Advances in Neural Information Processing Systems 16*, 2004, pp. 767–774.

[16] A. Hornung, A. Dornbush, M. Likhachev, and M. Bennewitz, "Anytime search-based footstep planning with suboptimality bounds," in *IEEE-RAS International Conference on Humanoid Robots*, 2012, pp. 674–679.

[17] K. Bouyarmane and A. Kheddar, "Humanoid robot locomotion and manipulation step planning," *Advanced Robotics*, vol. 26, no. 10, pp. 1099–1126, 2012.

[18] S. Tonneau, A. Del Prete, J. Pettr, C. Park, D. Manocha, and N. Mansard, "An efficient acyclic contact planner for multiped robots," *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 586–601, 2018.

[19] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, 2014.

[20] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous Robots*, vol. 40, no. 3, pp. 429–455, 2016.

[21] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[22] J. A. Reeds and L. A. Shepp, "Optimal paths for a car that goes both forwards and backwards." *Pacific J. Math.*, vol. 145, no. 2, pp. 367–393, 1990.

[23] S. Koenig and M. Likhachev, "D* lite," in *Eighteenth National Conference on Artificial Intelligence*. American Association for Artificial Intelligence, 2002, pp. 476–483.

[24] M. Likhachev, D. Ferguson, G. Gordon, A. T. Stentz, and S. Thrun, "Anytime dynamic A*: An anytime, replanning algorithm," in *International Conference on Automated Planning and Scheduling*, 2005.

[25] SBPL. Accessed: 2020-9-30. [Online]. Available: https://github.com/sbpl/sbpl

[26] 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 *IEEE International Conference on Robotics and Automation*, vol. 2, 2003, pp. 1620–1626.

[27] J. Englsberger, C. Ott, and A. Albu-Schffer, "Three-dimensional bipedal walking control based on divergent component of motion," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 355–368, 2015.

[28] O. Kanoun, F. Lamiraux, and P. Wieber, "Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 785–792, 2011.

[29] A. Escande, S. Miossec, M. Benallegue, and A. Kheddar, "A strictly convex hull for computing proximity distances with continuous gradients," *IEEE Transactions on Robotics*, vol. 30, no. 3, pp. 666–678, 2014.

[30] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng, "ROS: an open-source Robot Operating System," in *IEEE International Conference on Robotics and Automation Workshop on Open Source Software*, 2009.

[31] I. A. Şucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, 2012.

[32] K. Kaneko, H. Kaminaga, T. Sakaguchi, S. Kajita, M. Morisawa, I. Kumagai, and F. Kanehiro, "Humanoid Robot HRP-5P: An electrically actuated humanoid robot with high-power and wide-range joints," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1431–1438, 2019.

[33] J. H. Halton, "On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals," *Numerische Mathematik*, vol. 2, pp. 84–90, 1960.