IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Multi-Contact Locomotion Planning for Humanoid Robot Based on Sustainable Contact Graph with Local Contact Modification

Iori Kumagai[1], Mitsuharu Morisawa[1], Shizuko Hattori[1], Mehdi Benallegue[1], Fumio Kanehiro[1]

***Abstract*— In this paper, we propose a graph-search based multi-contact locomotion planning method for humanoid robots, focusing on the sustainability of contacts as its key feature. We introduce the idea of sustainable contact area, which represents the area on which contacts can be maintained during contact transitions. This enables us to select feasible contact candidates along a given root path. Then, we compute all the possible combinations of these candidate contacts with every limb appearing at most once, which we call contact sets. The list of these contact sets can be regarded as a list of nodes in a graph structure representing transitions between sustainable contacts, which we name as the sustainable contact graph. We apply A* search on this graph, and evaluate the connectability of nodes by planning quasi-static motion sequences for their contact transitions. In this process, we locally modify the candidate contact to satisfy kinematics constraints and static equilibrium of the robot. The proposed method enables us to plan feasible contact transition motions without random sampling or manually designed contact transition models, and solves the problem of ignoring possible contact transitions, which is caused by the discretization in existing graph-search based planners. We evaluate our proposed method in both simulation and a real robot, and confirm that it contributes to improving the multi-contact locomotion abilities of a humanoid robot.**

I. Introduction

A humanoid robot is expected to take over big burden tasks on human workers like large scale manufacturing [1]. In these scenarios, the robot is required to traverse environments designed for humans utilizing its human-like body structure. The multi-contact locomotion planning method is critical for these real-world uses, and has been researched in the field of humanoid robotics for many years. For a legged robot, multi-contact locomotion is a sequence of contact transitions. However, it is still an open problem to find feasible contact transitions for a humanoid robot in a large search space of candidate contacts. This problem can be simplified by pre-defining contact transition candidates carefully designed by an operator. However, these manually discretized models may cause oversight of possible contact transitions, which can result in the failure of the locomotion planning.

In this paper, we propose a graph-search based contact transition planning method for multi-contact locomotion by a humanoid robot as shown in Fig. 1. The target of our paper is to reveal a method to find a sequence of feasible contact transition motion. In order to evaluate feasibility of the contact transition, we focus on the sustainability of a contact, based on the idea of "geometrical feasibility" [2]. A contact is defined as sustainable when it can be maintained by a robot during contact transitions. This concept enables us to effectively find the feasible contact transitions without random sampling or manually designed contact transition candidates, and to locally modify a target contact in order to solve the oversight problem while maintaining its feasibility. We also apply our proposed method to a humanoid robot in both simulation and real environments, which experimentally proves its practicality.

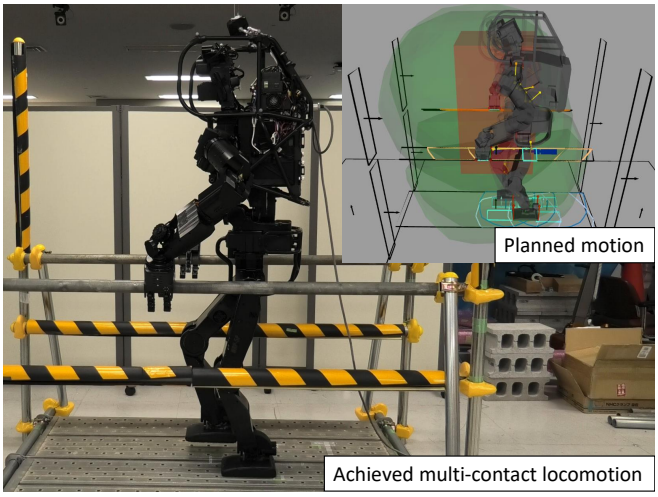

Fig. 1. The achieved multi-contact locomotion by the proposed method, where the humanoid robot traversed a corridor using handrails.

II. Related Works

A. Methodologies to find feasible contact transitions

Generally, a contact transition is regarded as feasible when there is an executable motion sequence for it. In this context, it is a challenging problem to find a sequence of such feasible contacts considering kinematics and stability constraints. The random sampling approach is one of the common solutions to this problem [3]. However, this kind of approach has the problem of increasing computational costs because of the very large search space for possible contact transitions. Recently, Tonneau et al. [4] proposed an efficient algorithm to plan multi-contact locomotion for a multiped robot. They utilized the reachability model, which consists of the bounding box of the trunk and the convex hulls of the reachable points for each limb, to find a global path where random sampling based planner could generate a sequence of multi-contact postures. However, there remains the problem that the resulting postures are not always executable. Fernbach et al. [5] proposed an efficient method to reject infeasible contacts based on the existence of a dynamic CoM trajectory

This work was partially supported by JSPS KAKENHI Grant Number 19K20380.

[1] I. Kumagai, M. Morisawa, S. Hattori, M. Benallegue and F. Kanehiro are with CNRS-AIST JRL (Joint Robotics Laboratory), IRL, National Institute of Advanced Industrial Science and Technology (AIST), 1-1-1 Umezono, Tsukuba, 305-8560 Ibaraki, Japan, iori-kumagai@aist.go.jp

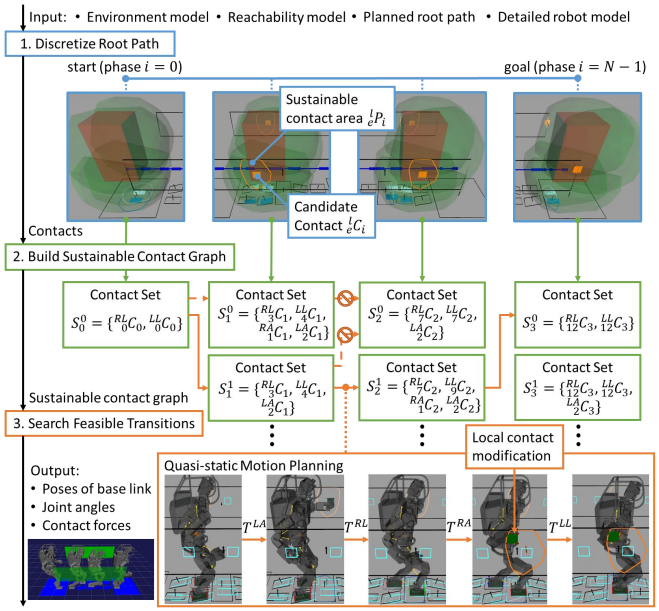

Fig. 2. Overview of the proposed multi-contact locomotion planning.

in a contact transition, but it is still difficult to find feasible contact transitions within reasonable computational costs while planning.

Another approach is to manually design possible contact transitions in advance, which can simplify the feasibility problem in the contact transition planning [6], [7]. However, these contact planning algorithms based on contact transition models have the disadvantage of oversight in the possible contact transitions, because they can search only discretized nodes in the target graph. Griffin et al. [8] proposed a local footstep modification method for a discretized footstep planning system, but this kind of local modifications has not been introduced to multi-contact planners yet because it changes the number and positions of friction cones. Brossette et al. [9] proposed the method to allow a whole-body posture generator to consider a partial contact, but it only evaluated the stability of the resulting posture when the new contact did not support any contact forces.

Instead of search-based approaches, several methods to formulate the multi-contact planning problem as an optimization problem are also proposed [10], [11]. These optimization-based methods do not suffer from the limitations related to the discretization, but they are less robust than search-based methods, which can backtrack and search suboptimal but reasonable contact transitions.

B. Overview of the proposed approach

Based on the above discussion, we propose a graph-search based contact transition planning method for multi-contact locomotion of a humanoid robot. The overview of the proposed method is shown in Fig. 2. The proposed system assumes that the environment model, the reachability model [4] of a robot, detailed robot model and its root path are given as input. First, we discretize the input root path to find the candidate contacts along it. In this process, we compute a sustainable contact area, which represents the existing

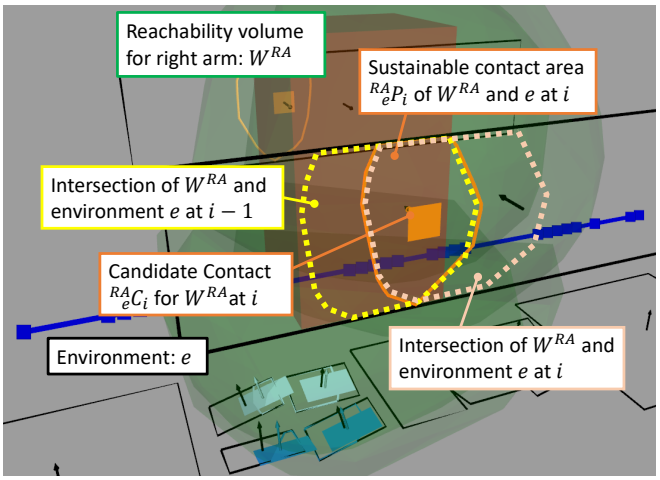

Fig. 3. Overview of the idea of geometrical feasibility. In this figure, we use the right arm as an example ($l = RA$).

area of the contacts that can maintained while the robot moves between certain points on the path. Then, we put the initial guesses for the candidate contacts on the centroid of each sustainable contact area, and compute all their possible combinations for each phase. We call these combinations contact sets. These contact sets are regarded as the nodes of a graph representing feasible contact transitions, which we refer to as the sustainable contact graph. We define the cost function based on the geometrical features of the contact set, and search for the appropriate contact transition sequence using A* search [12] on this graph. When connecting the contact sets, we evaluate their connectability by planning quasi-static motion sequence for their contact transition. In this process, we also search its transition orders and locally modify the candidate contacts to satisfy kinematics constraints and static equilibrium. The approach proposed by Escande et al. [13] also found candidate contacts for graph-search based contact planning by a quadratic programming based posture generator. In contrast to them, the novelty of our method is the concept of sustainable contact area, which contributes to efficiently finding feasible contact transitions without random sampling or pre-defined contact transition models. It also enables us to modify a contact locally while maintaining its sustainability during the contact transition.

III. Sustainable Contact Graph

We assume that the environment E consists of convex polygons, and the reachability model of the robot is generated in advance. In this paper, we generate the input path by the method proposed in our previous paper [2], which is optimized RB-RRT [4].

A. Generating sustainable contact areas from the root path

1) Sustainability of the contact: In general, at least one contact should be maintained in order to support the robot while some contacts are in transition. Therefore, it is important to consider this contact sustainability during the contact transition to achieve multi-contact locomotion. When estimating the contactable area, it is a reasonable approximation that the reachable area of a limb l in the environmental

Algorithm 1 Path discretization

Input: $path$ - Input path for the reachability model
Output: X - List of the sustainable contact areas

$\boldsymbol{p}_0 \leftarrow$ getStart($path$)
$i \leftarrow 0$
$X \leftarrow \emptyset$
while not reachedToGoal($path$, $\boldsymbol{p}_i$) **do**
  $\Delta d \leftarrow \Delta d^{def}$
  while $\Delta d \geq \Delta d^{min}$ **do**
    $\boldsymbol{p}_{i+1} \leftarrow$ goAlongThePath($path$, $\boldsymbol{p}_i$, Δd)
    $\{{}^l_e P_{i+1}\} \leftarrow$ getSustainableContactAreas($\boldsymbol{p}_i$, $\boldsymbol{p}_{i+1}$)
    if $\{{}^l_e P_{i+1}\} \neq \emptyset$ **then**
      add($\{{}^l_e P_{i+1}\}$, X)
      $i \leftarrow i + 1$
      break
    end if
    $\Delta d \leftarrow \frac{\Delta d}{2}$
  end while
  if $\Delta d < \Delta d^{min}$ **then**
    return $\emptyset$
  end if
end while
return X

polygon e is the intersection of the reachability volume of l and e [14]. However, this assumption cannot take the sustainability of resulting contact during the contact transition into account. In order to solve this problem, we proposed "geometrical feasibility" in our previous paper [2]. If we know the posture of a robot in the target phase i and previous phase $i-1$, the area where the sustainable contact ${}^l_e C_i$ for l and e exists is expected to be the common part of their reachable areas ${}^l_e P_i$, as shown in Fig. 3. In this paper, we introduce this idea to the graph-search based contact transition planning and show that it can also solve the problem of the discrete graph search.

2) Path discretization: In order to apply the idea of geometrical feasibility, we need to divide the locomotion process into discretized phases as shown on the top of Fig. 2. We generate these phases by discretizing the input root path based on the existence of sustainable contact areas. The algorithm for path discretization is shown in Alg. 1. The parameters Δd^{def} and Δd^{min} represent the default and minimum discretization distance respectively. We travel Δd along the path from the i-th point $\boldsymbol{p}_i \in \mathbb{R}^6$ on it, and place the reachability model there. Then, we compute the sustainable contact areas as shown in Fig. 3. We make the discretization step Δd half when there are no sustainable contact areas. If we cannot find any sustainable contact until Δd become smaller than Δd^{min}, this discretization process will return with failure. As a result of this path discretization, we obtain list of the sustainable contact areas X with the N discretized phases.

For each sustainable contact area ${}^l_e P_i$ in X, we define an initial guess of the contact ${}^l_e C_i$ at its centroid. Then, we compute the area of intersection between the contact surface of limb l at ${}^l_e C_i$ and the environment e. This intersection

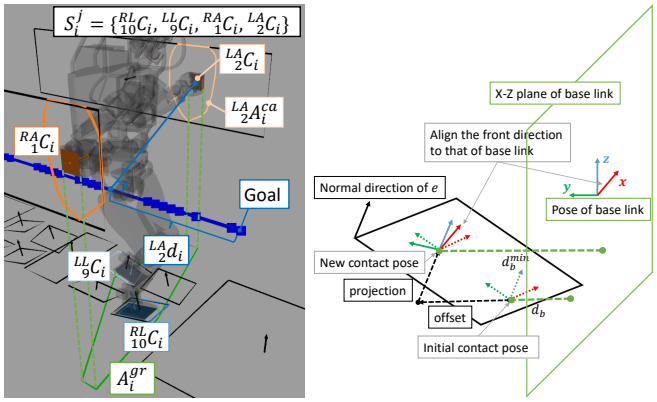

Fig. 4. Left: Definition of variables for the cost function $f(S_i^j)$. Right: The offset computation of the leg contact for a humanoid robot.

represents the support area for ${}^l_e C_i$. If the ratio of this support area to the entire area of contact surface of limb l is smaller than the pre-defined threshold r_{sp}^{th}, we reject ${}^l_e C_i$ and remove ${}^l_e P_i$ from X. By this contact generation process, we can obtain the list of candidate contacts $\{{}^l_e C_i\}$ for phase i.

B. Building sustainable contact graph

1) Generation of contact sets as graph nodes: Next, we compute the list of all possible combinations of $\{{}^l_e C_i\}$, where every limb l appears at most once. We define the j-th combination of candidate contacts in phase i as a contact set S_i^j as shown on the middle of Fig. 2. Empty contact sets are not allowed. The basic idea behind our contact transition graph generation is that the list of contact set $\{S_i^j\}$ can be regarded as a list of nodes. The contact sets in $\{S_i^j\}$ can be connected only from those in $\{S_{i-1}^j\}$, and to those in $\{S_{i+1}^j\}$. In addition, these contacts are expected to be sustainable in their transitions since all the contacts are inside of the sustainable contact area. Therefore, we define this graph as a sustainable contact graph, and formulate the problem of finding the feasible contact transition sequence as searching the valid path from $i = 0$ to $i = N - 1$ in it. This idea removes the necessity of pre-defined successors for the graph-search based planner to expand the child nodes.

2) Cost function for A graph search:* For the graph search algorithm, we adopt A* search. In order to apply this algorithm to the proposed sustainable contact graph, we need to define an appropriate cost function for the node S_i^j. In A* search, the cost of the node $f(S_i^j)$ is defined as the sum of the path cost $g(S_i^j)$ and heuristics cost $h(S_i^j)$.

First, we define the cost of a contact $f_c({}^l_e C_i)$ in the contact set S_i^j as (1). As shown on the left of Fig. 4, ${}^l_e A_i^{ca}$ is the square measure of the sustainable contact area ${}^l_e P_i$ and ${}^l_e d_i$ is the distance from the closest point from ${}^l_e C_i$ on the input root path to its goal. The weights w_{ca} and w_{ds} are pre-defined parameters. A small ${}^l_e A_i^{ca}$ means that the ${}^l_e P_i$ is near the boundary of the robot's reachability volume or the environmental polygon, which means that the manipulability is expected to be low, or highly accurate manipulation is required. Therefore, $f_c({}^l_e C_i)$ imposes a large cost to the contact which is difficult to be reached or far from the goal.

$$f_c({}^l_e C_i) = \frac{w_{ca}}{{}^l_e A_i^{ca}} + w_{ds}\, {}^l_e d_i \tag{1}$$

Next, we define the reaching cost, which represents the cost to reach the contact set S_i^j from its parent node, as (2). A_i^{gr} is the area of the convex polygon computed by all the vertices of contacts in S_i^j projected on the ground plane, as shown on the left of Fig. 4. The weight w_{sp} is a pre-defined parameter. This is not a strict support area, but can be used as a criterion of the supportability.

$$c(S_i^j) = \sum_{{}^l_e C_i \in S_i^j} f_c({}^l_e C_i) + \frac{w_{sp}}{A_i^{gr}} \tag{2}$$

Then, we define the path cost for the contact set $g(S_i^j)$ as (3). S_{i-1}^k is the parent node of S_i^j. The function $\alpha_f(S_i^j)$ represents a penalty cost, which increases in proportion to the number of past failures in the transition planning for S_i^j. The proportional coefficient a_f for $\alpha_f(S_i^j)$ is pre-defined.

$$g(S_i^j) = g(S_{i-1}^k) + c(S_i^j) + \alpha_f(S_i^j) \tag{3}$$

Finally, we define the heuristics cost $h(S_i^j)$ for the node S_i^j. Since we already know all the nodes in the sustainable contact graph, we can compute the smallest reaching cost c_i^{min} for each phase i. We use the sum of this c_i^{min} from phase $i+1$ to phase $N-1$ as the heuristics cost for the nodes in phase i as (4). This heuristics cost is obviously consistent, which guarantees the optimality of the A* search.

$$h(S_i^j) = \sum_{k=i+1}^{N-1} c_k^{min} \tag{4}$$

3) Additional assumptions for a humanoid robot: The sustainable contact graph proposed above can be applied to any type of robot which makes contacts for locomotion. However, since our target is a humanoid robot, we add some additional assumptions in order to make resulting motion more suitable for it. First, we assume that a robot makes contact with environment only by right arm (RA), left arm (LA), right leg (RL) and left leg (LL). In addition, we require that candidate contacts for both legs should be included in all of the contact set S_i^j. This also means that at least one sustainable contact area for both legs are required in the path discretization process. Finally, we keep the distance from the base link to the contact for a leg d_b larger than d_b^{min}, and we align its front direction to that of base link as shown on the right of Fig. 4. If this modified contact is out of its sustainable contact area, we reject it before computing contact sets.

IV. Motion Planning for Contact Transition

During A* search for the sustainable contact graph, we need to determine whether S_i^j and S_{i+1}^k are connectable or not. We consider that these nodes are connectable when a valid quasi-static motion sequence for the contact transition between them exists as shown on the bottom of Fig. 2.

A. Transition order between contact sets

1) Contact transition pair: In this paper, we assume that the robot moves only one limb at once. However, this assumption raises a problem of the order of contact transitions. Although each contact is inside of its sustainable contact area, it may not be reached because of the collisions or stability constraints, and these factors will depend on the sequence of contact transitions. In order to appropriately sort contact transitions from S_i^j to S_{i+1}^k, we define the contact transition pair $T^l = \{{}^l_n C_i, {}^l_m C_{i+1}\}$ for each limb l. This pair means that limb l will move from ${}^l_n C_i \in S_i^j$ to ${}^l_m C_{i+1} \in S_{i+1}^k$. When the limb l is not in contact, $\emptyset$ is put into T^l. The empty contact transition pair $T^l = \{\emptyset, \emptyset\}$ is ignored. In addition, when ${}^l_n C_i$ is inside of the sustainable contact area ${}^l_m P_{i+1}$ of the target contact ${}^l_m C_{i+1}$, and the displacements of both translation and rotation from ${}^l_n C_i$ to ${}^l_m C_{i+1}$ are less than the thresholds ΔC_t^{th} and ΔC_r^{th}, we assume that ${}^l_n C_i$ can be maintained in phase $i+1$ and do not plan motion for T^l.

2) Searching transition order: First, we sort the set of these contact transition pairs $T = \{T^l\}$ in descending order of $f_c({}^l_m C_{i+1})$. $f_c(\emptyset)$ is regarded as the infinity cost. This sorted order of the contact transition pairs is used as the initial guess of the transition order. Then, we plan contact transition motion for T^l from the top of T. When this motion planning for T^l is failed, the order of contact transition pairs in T is reorganized in lexicographic order at its index, and we re-try planning quasi-static motion from the top of the new T as shown in Fig. 5. If the planner reaches the last permutation of the index order, we consider that S_i^j and S_{i+1}^k are not connectable.

B. Motion generation with local contact modification

1) Quasi-static whole-body motion planning: We plan the quasi-static whole-body motion sequence for the contact transition T^l to evaluate its feasibility. Prete et al. [15] proposed the linear programming formulation to evaluate static equilibrium with friction coefficient μ. We introduce the improved formulation of their work, which can compute statically stable CoM at the same time [2]. Then, in order to consider the kinematic constraints of a robot, we use inverse kinematics based on the prioritized quadratic programming [16]. We introduced the collision avoidance constraints [17] with the efficient GJK algorithm [18], 6D posture constraints for contacting end-effectors, joint angle limitation, and joint velocity limitation as priority 0, which is the highest priority. We also add the constraint for CoM position, which is computed to satisfy the static equilibrium condition [2], as priority 1. However, since the next target contact ${}^l_m C_{i+1}$ is just put on the centroid of ${}^l_m P_{i+1}$, it may not be feasible when the kinematics constraints of a robot are taken into account as shown in the lower right of Fig. 6.

2) Local contact modification: We need to find the feasible target contact to satisfy kinematics constraint of a robot, but we also need to keep its sustainability for following contact transitions. Since we already know ${}^l_m P_{i+1}$, it is reasonable to assume that the modified target contact ${}^l_m \hat{C}_{i+1}$ is expected to be sustainable if it is kept in ${}^l_m P_{i+1}$. Therefore, we define the inequality constraint to make the contact point $\boldsymbol{r} \in \mathbb{R}^3$ of the limb l in the convex polygon ${}^l_m P_{i+1}$. We assume that the vertices $\{\boldsymbol{v}_0, \cdots, \boldsymbol{v}_K\}$ of ${}^l_m P_{i+1}$ are ordered

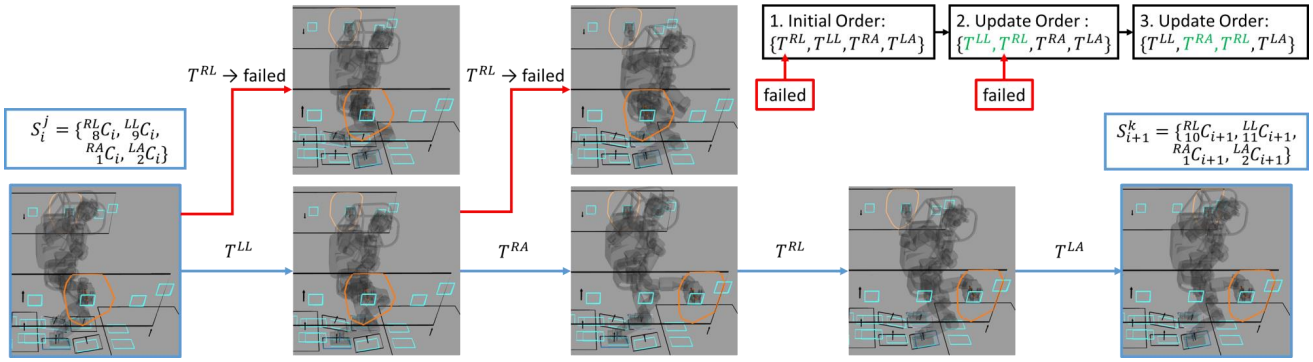

Fig. 5. Example of searching contact transition order from S_i^j to S_{i+1}^k. In this example, the planner failed to plan quasi-static motion for T^{RL} twice, which caused reorganization of T as shown in the green characters.

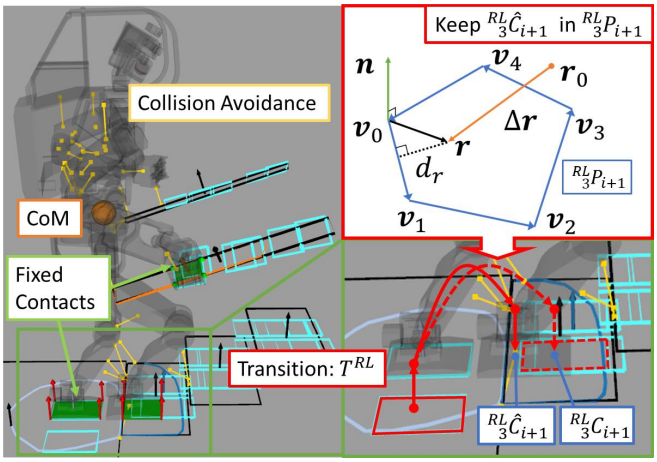

Fig. 6. Left: Kinematics constraints to generate a quasi-static multi-contact motion sequence with local contact modification. Right: The constraint to keep sustainability in the local contact modification. The original ${}^{RL}_{3}C_{i+1}$ will make the collision between the right leg and the second step.

Algorithm 2 Local contact modification

Input: ${}^{l}_{m}C_{i+1}$ - next target contact, $\boldsymbol{x}_0$ - initial robot state
Output: ${}^{l}_{m}\hat{C}_{i+1}$ - modified target contact

$S \leftarrow$ getContactSet($\boldsymbol{x}_0$)
removeContactForLimb(l, S)
addContact(${}^{l}_{m}C_{i+1}$, S)
$\boldsymbol{c}_{ref} \leftarrow$ getStaticallyStableCoM(S)
$count \leftarrow 0$
while $count < maxCount$ **and not** $\boldsymbol{c}_{ref} = NaN$ **do**
  $\boldsymbol{x}_{kp} \leftarrow$ solveIK(S, $\boldsymbol{c}_{ref}$)
  if not $\boldsymbol{x}_{kp} = NaN$ **then**
    $S \leftarrow$ getContactSet($\boldsymbol{x}_{kp}$)
    $\boldsymbol{c} \leftarrow$ getCoM($\boldsymbol{x}_{kp}$)
    if $\|\boldsymbol{c}_{ref} - \boldsymbol{c}\| < \epsilon$ **then**
      ${}^{l}_{m}\hat{C}_{i+1} \leftarrow$ getContactForLimb(l, S)
      return ${}^{l}_{m}\hat{C}_{i+1}$
    end if
    $\boldsymbol{c}_{ref} \leftarrow$ getStaticallyStableCoM(S)
  end if
  $count \leftarrow count + 1$
end while
return NaN

in counter-clockwise as shown in the upper right of Fig. 6. For the local coordinate system of ${}^{l}_{m}P_{i+1}$, the normal vector $\boldsymbol{n} \in \mathbb{R}^3$ represents the direction of its z-axis, and its x-axis and y-axis are same as those of ${}^{l}_{m}C_{i+1}$. Then, the constraint for $\boldsymbol{r}$ to be kept in ${}^{l}_{m}P_{i+1}$ is written as (5). d_r^{min} is the allowable minimum distance from $\boldsymbol{r}$ to the each edge.

$$\begin{cases} \boldsymbol{n}^T\{(\boldsymbol{v}_1 - \boldsymbol{v}_0) \times (\boldsymbol{r} - \boldsymbol{v}_0)\} & > d_r^{min}\|\boldsymbol{v}_1 - \boldsymbol{v}_0\| \\ & \vdots \\ \boldsymbol{n}^T\{(\boldsymbol{v}_0 - \boldsymbol{v}_K) \times (\boldsymbol{r} - \boldsymbol{v}_K)\} & > d_r^{min}\|\boldsymbol{v}_0 - \boldsymbol{v}_K\| \end{cases} \tag{5}$$

Using the jacobian matrix J_l for limb l, the previous contact point of the end-effector $\boldsymbol{r}_0$ and the selection matrix $W = [I_{1\times3}\ O_{1\times3}]$, the end-effector position $\boldsymbol{r}$ can be represented by the configuration of a robot $\boldsymbol{q}$ as (6).

$$\boldsymbol{r} = \boldsymbol{r}_0 + WJ_l\Delta\boldsymbol{q} \tag{6}$$

By substituting (6) for (5), we can obtain the inequality constraint for $\Delta\boldsymbol{q}$ to keep $\boldsymbol{r}$ in ${}^{l}_{m}P_{i+1}$ as (7).

$$A_pWJ_l\Delta\boldsymbol{q} < \boldsymbol{b}_p - A_p\boldsymbol{r}_0 \tag{7}$$

A_p and $\boldsymbol{b}_p$ are obtained from (5) as (8).

$$\begin{cases} A_p &= \begin{bmatrix} \boldsymbol{n}^T[\boldsymbol{v}_0 - \boldsymbol{v}_1]\times \\ \vdots \\ \boldsymbol{n}^T[\boldsymbol{v}_K - \boldsymbol{v}_0]\times \end{bmatrix} \\ \boldsymbol{b}_p &= \begin{bmatrix} \boldsymbol{n}^T(\boldsymbol{v}_0 \times \boldsymbol{v}_1) \\ \vdots \\ \boldsymbol{n}^T(\boldsymbol{v}_K \times \boldsymbol{v}_0) \end{bmatrix} - d_r^{min} \begin{bmatrix} \|\boldsymbol{v}_0 - \boldsymbol{v}_1\| \\ \vdots \\ \|\boldsymbol{v}_K - \boldsymbol{v}_0\| \end{bmatrix} \end{cases} \tag{8}$$

By using (7) and constraining the displacement on the z-axis position, its rotation around x-axis and its rotation around y-axis for the end-effector of the limb l relative to the local coordinate system of ${}^{l}_{m}P_{i+1}$ to 0, we keep the limb l contact inside of the ${}^{l}_{m}P_{i+1}$ while solving prioritized inverse kinematics. Their priorities are defined as 0. We also constrain the 2D pose on the local x-y plane of ${}^{l}_{m}P_{i+1}$ to ${}^{l}_{m}C_{i+1}$ with the lowest priority.

Then, we obtain the new target contact ${}^{l}_{m}\hat{C}_{i+1}$. The algorithm for the local contact modification is shown in Alg. 2. The initial robot state $\boldsymbol{x}_0$ is the last robot state obtained from the previous contact transition planning. Since the static equilibrium evaluation requires the vertices of the support polygon for each contact, modifying a target contact makes static equilibrium evaluation a non-linear problem. We solved

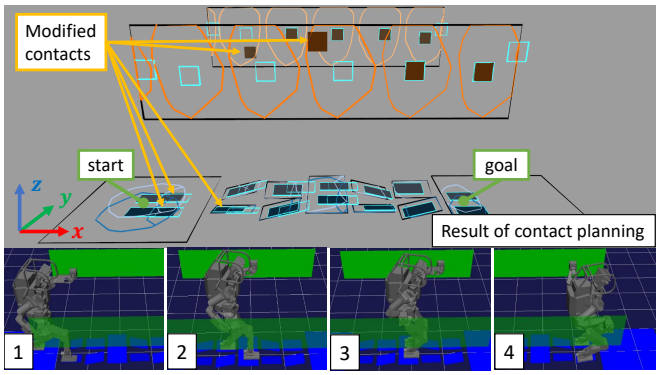

Fig. 7. The result of contact planning in traversing a terrain. Upper: The generated contact candidates (light blue borders) and resulting contact sequence (filled polygons). Lower: The resulting multi-contact motion in a kinematics simulation.

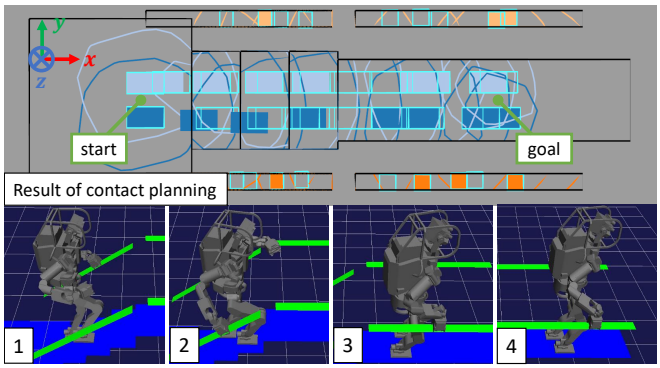

Fig. 8. The result of contact planning in traversing a scaffold. Upper: The generated contact candidates (light blue borders) and resulting contact sequence (filled polygons). Lower: The resulting multi-contact motion in a kinematics simulation.

this problem by dividing the motion planning process into CoM based static equilibrium evaluation and whole-body inverse kinematics. First, we compute the reference CoM $\boldsymbol{c}_{ref}$ from static equilibrium evaluation, and sequentially solve inverse kinematics to obtain the the keypose $\boldsymbol{x}_{kp}$, which satisfies the kinematics constraints discussed above. When the resulting $\boldsymbol{x}_{kp}$ is not NaN, which means inverse kinematics is successfully solved, and the difference between $\boldsymbol{c}_{ref}$ and the resulting CoM $\boldsymbol{c}$ from $\boldsymbol{x}_{kp}$ converges within the threshold ϵ, we compute the posture of the end-effector for the limb l from $\boldsymbol{x}_{kp}$. We use it as the new target contact ${}^{l}_{m}\hat{C}_{i+1}$ for the motion planning of T^l. If $\boldsymbol{c}_{ref}$ is NaN, which means that a statically stable CoM cannot be found, or the resulting CoM cannot converge within the maximum trial count, the local contact modification process returns with failure and the motion planning for T^l fails. The resulting ${}^{l}_{m}\hat{C}_{i+1}$ is not only feasible for kinematics constraints, but is also still sustainable because of (7).

3) Generating contact transition motion: After the local contact modification step, we plan the quasi-static contact transition for moving limb l to ${}^{l}_{m}\hat{C}_{i+1}$ along a cubic-spline trajectory. In this phase, target contacts are not modified any longer. The modified contact ${}^{l}_{m}\hat{C}_{i+1}$ is used only for this contact transition, and the original contact set S^k_{i+1} is not modified. We generate contact transition motion for one iteration in 20ms. More details are available in [2].

V. Experimental Evaluation

We applied the proposed method to our humanoid robot HRP-5P [19]. In the following experiments, the proposed planner was executed with Intel(R) Xeon(R) CPU E5-1680 v4. We used Choreonoid [20] as simulation environment. In the real world experiment, we sent the resulting motion to the multi-contact stabilization controller [21]. The parameters for each experiment are shown in TABLE I.

A. Simulation experiments

1) Traversing a terrain: We conducted a simulation experiment to plan multi-contact locomotion to traverse a terrain. The terrain consisted of 30×20cm rectangles rotated at 0.25rad around various directions and two vertical walls. The result of this experiment is shown in Fig. 7. The proposed planner succeeded in connecting 8 contact sets without failure. It successfully generated motions for 7 out of 19 contact transitions. There are also 6 modified contacts out of 30 planned contacts in total due to the kinematics constraints. The computational time was 20.62s.

2) Traversing a scaffold: We also planned multi-contact locomotion to traverse a scaffold in a simulation. This scaffold is a series of three 30cm×60cm steps which have 15cm height, and a 1.8m×50cm walkway. Each part has two handrails with 1m spacing. The result of this experiment is shown in Fig. 8. The proposed planner connected 9 contact sets without failure, where it succeeded to plan 8 contact transitions out of 9, and modified 8 contact out of 21. The total computational time was 12.64s. This experiment showed that the proposed planner can efficiently plan multi-contact motion in a series of different environments as well.

3) Performance evaluations: We quantitatively evaluated the success rate and computational efficiency of the proposed planner in the above two simulation environments. We tested the entire proposed planner (ALL), the one without searching transition order (W/O TO) and the one without local contact modification (W/O CM). We applied random offsets in the range of $\pm$ 10cm to the x and y position of pre-defined start and goal for each of the 50 trials in each scenario. The motion planning was regarded as failed if it took over 90s to find a solution. The results are shown in TABLE II. We only consider successful trials to obtain the average computational time and the average effective branching factor b^* [12]. In both environments, the entire planner succeeded in finding a solution for 90% of the trials. As a comparison, the average computational time of the "ALL" scenario in the terrain outperforms the state-of-the-art planner in similar uneven platforms with comparable success rate [5] (17.83s planning time and 83.5% success rate). It should be noted that our motion is quasi-static but kinematically more complex because of arm contacts. In terms of efficiency, b^* was close to 1.0 for both environments, which proved that proposed heuristics can find a solution without expanding less unnecessary nodes than the existing method [13] ($b^* = 1.12$ for climbing a ladder for example). TABLE II also showed that the success rates dropped and b^* increased for the "W/O TO" scenario in both environments. This proved

TABLE I

PARAMETERS FOR THE EXPERIMENTAL EVALUATION

Parameter	Δd^{def}	Δd^{min}	r_{sp}^{th}	w_{ca}	w_{ds}	w_{sp}	a_f	d_b^{min}	ΔC_t^{th}	ΔC_r^{th}	d_r^{min}	ϵ	μ
Terrain	0.4m	0.05m	0.5	1.0	2.0	5.0	50.0	0.1m	0.05m	0.087rad	0.05m	0.01m	0.6
Scaffold	0.32m		0.8		8.0	7.0		0.1m	0.07m		0.01m		
Stairs	0.34m		0.5		3.0	6.0		0.06m	0.12m		0.01m		
Corridor	0.3m		0.8		1.0	10.0		0.08m	0.05m		0.01m		

TABLE II

PERFORMANCE EVALUATION RESULT IN 50 TRIALS

Scenario	Terrain			Scaffold		
	ALL	W/O TO	W/O CM	ALL	W/O TO	W/O CM
Success	90%	72%	88%	90%	26%	60%
Time	15.70s	25.43s	15.91s	20.94s	45.61s	33.40s
b^*	1.01	1.14	1.02	1.05	1.27	1.13

that searching transition order also played an important role in improving efficiency of the planner. Although the "W/O CM" scenario did not make a large difference in the terrain, it worsened the performance of the planner in the scaffold. This is because the collision avoidance constraints were severe in the scaffold, especially while stepping up stairs. On the other hand, the sustainable contact area cannot guarantee a solution of inverse kinematics because it also depends on the initial state and constraints such as collision avoidance. This limitation can be shown by the fact that the major reason of the failures was timeout due to a series of unsuccessful inverse kinematics. TABLE II suggests that although the sustainable contact area has potential to efficiently find promising contact candidates, we need transition order search and local contact modification to properly generate valid multi-contact motions.

B. Real world experiments

1) Stepping up stairs: We conducted a real world experiment to step up stairs with handrails. The stairs consisted of three steps, whose height was 15cm and depth was 28cm, and two handrails. Although the spacing of the handrails was 64cm and it was notably narrow for HRP-5P, the proposed planner succeeded in connecting 5 contact sets out of 8, and it successfully generated motion for 4 contact transition orders out of 17 in total. In addition, the local contact modification, which modified 22 contacts out of 28 in total, was also critical for collision avoidance as shown on the left of Fig. 9. The total computational time in this experiment was 20.01s. We also confirmed that the resulting motion enabled the real HRP-5P to climb up the staircase using a handrail as shown on the right of Fig. 9.

2) Walking on a corridor: Finally, we conducted a real world experiment to walk on a corridor with handrails. Since the robot needed to use its arms to keep its balance on an elastic corridor, the parameters were assigned high priorities on increasing A_i^{gr}. The planning result for this experiment is shown on the left of Fig. 10. The proposed planner succeeded in connecting 3 contact sets and generated motion for 2 contact transitions without failure. The local contact modification did not occur in 8 planned contacts. The total computational time in this experiment was 7.10s. The resulting motion was successfully executed by the real HRP-5P as shown on the right of Fig. 10, which utilized both arms to support the robot as intended.

VI. DISCUSSION

A. Computational complexity

In the worst case scenario, the time complexity of the proposed planner for A* search is $\mathcal{O}(b^N)$, where b is the branching factor [12]. Therefore, when the traveling distance becomes longer or Δd^{def} becomes smaller, the time complexity can grow exponentially. In addition, b depends on the number of contact sets in each phase. This suggests that when the number of contactable surfaces or limbs is large, the number of contact sets may suffer from a combinatorial explosion, which will make b large. The proposed method efficiently avoid these problems by properly designing costs and heuristics, which was experimentally proved by the low effective branching factor as discussed in Subsection V-A.

B. Limiting assumptions

In this paper, we employed the environmental polygons as the input to the proposed planner, which also means that we pre-defined the contact direction for each surface. We can mitigate this limitation by using more accurate meshes for environment models, but that will cause a combinatorial explosion of possible contact sets. In addition, since our planner assumes that the robot will move forward following the given root path, it is difficult to plan in-place contact repositioning. It can be possible to approximate some in-place contact sequence by making discretization step of the input path smaller, but it may also increase the time complexity as the trade-off factor.

C. Static equilibrium evaluation

Since we modified the contact transition order and the target contacts while motion planning, we need to compute the reference CoM and evaluate the stability criterion repeatedly. Therefore, in order to reduce computational costs, we adopted static equilibrium evaluation, which can compute a statically stable CoM and contact forces from the current target contacts by the quadratic programming [2]. Although this assumption makes resulting motion quasi-static, we experimentally proved that our planner can achieve practical multi-contact locomotion with the dynamic stabilization controller [21]. However, in order to perform more challenging locomotion, we will introduce a dynamic stability criterion such as proposed by Lin et al. [22] in the future.

VII. CONCLUSION

In this paper, we proposed a contact transition planning method for multi-contact locomotion by a humanoid robot. This method searches the graph representing the possible transitions of sustainable contact sets. This sustainable contact graph can efficiently find feasible contacts without predefined successors or random sampling. Then, we search for

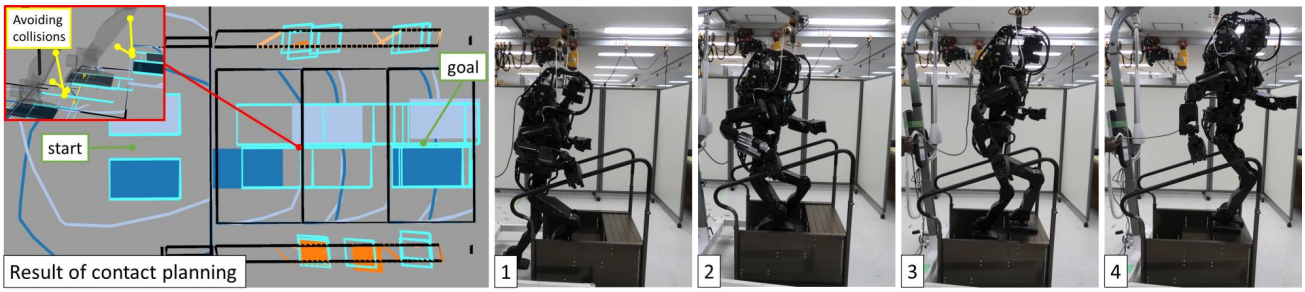

Fig. 9. Left: The contact candidates (light blue borders) and the resulting contact sequence (filled polygons). Some contacts were modified to avoid collisions (emphasized by yellow lines). Right: The resulting multi-contact motion sequence of the real HRP-5P to step up stairs with handrails.

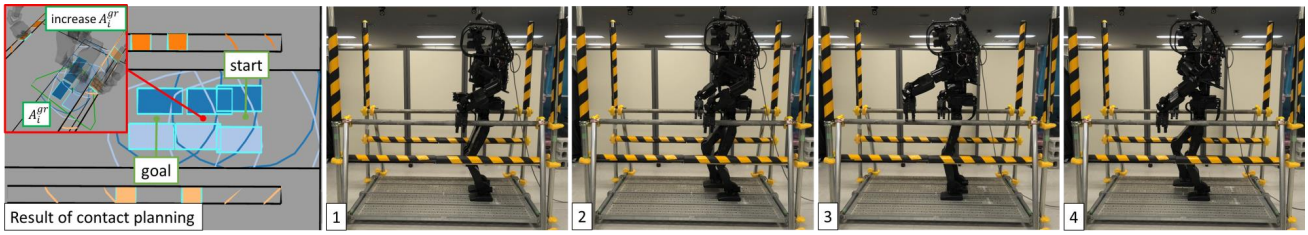

Fig. 10. Left: The contact candidates (light blue borders) and the resulting contact sequence (filled polygons). The parameters preferred to use contact sets which had larger A_i^{gr} (green polygon) in this experiment. Right: The resulting multi-contact motion of the real HRP-5P to traverse a corridor.

a sequence of quasi-static contact transitions by applying A* search to it. In this process, we locally modify the next target contact to satisfy the kinematics constraints and static equilibrium conditions, which avoids the oversight of possible contact transitions caused by the discretization for the graph-search algorithm. We experimentally confirmed that the proposed method could provide feasible motions for HRP-5P to traverse simulated and real environments, and concluded that the proposed method contributes to improving multi-contact locomotion abilities of a humanoid robot.

REFERENCES

[1] I. Kumagai, M. Morisaswa, T. Sakaguchi, S. Nakaoka, K. Kaneko, H. Kaminaga, S. Kajita, M. Benallegue, R. Cisneros, and F. Kanehiro. Toward Industrialization of Humanoid Robots: Autonomous Plasterboard Installation to Improve Safety and Efficiency. *IEEE Robotics and Automation Magazine*, (October):2–11, 2019.

[2] I. Kumagai, M. Morisawa, M. Benallegue, and F. Kanehiro. Bipedal Locomotion Planning for a Humanoid Robot Supported by Arm Contacts Based on Geometrical Feasibility. In *IEEE-RAS International Conference on Humanoid Robots*, pages 140–147, 2019.

[3] K. Hauser, T. Bretl, J.C. Latombe, K. Harada, and B. Wilcox. Motion planning for legged robots on varied terrain. *International Journal of Robotics Research*, 27(11-12):1325–1349, 2008.

[4] S. Tonneau, A. Del Prete, J. Pettré, C. Park, D. Manocha, and N. Mansard. An Efficient Acyclic Contact Planner for Multiped Robots. *IEEE Transactions on Robotics*, 34(3):586–601, 2018.

[5] P. Fernbach, S. Tonneau, O. Stasse, J. Carpentier, and M. Taix. C-CROC: Continuous and Convex Resolution of Centroidal Dynamic Trajectories for Legged Robots in Multicontact Scenarios. *IEEE Transactions on Robotics*, 36(3):676–691, jun 2020.

[6] Y. C. Lin and D. Berenson. Using previous experience for humanoid navigation planning. *IEEE-RAS International Conference on Humanoid Robots*, pages 794–801, 2016.

[7] A. Dornbush, K. Vijayakumar, S. Bardapurkar, F. Islam, and M. Likhachev. A Single-Planner Approach to Multi-Modal Humanoid Mobility. In *IEEE International Conference on Robotics and Automation*, pages 4334–4341, 2018.

[8] R. J. Griffin, G. Wiedebach, S. McCrory, S. Bertrand, I. Lee, and J. Pratt. Footstep Planning for Autonomous Walking Over Rough Terrain. In *IEEE-RAS International Conference on Humanoid Robots*, pages 9–16, 2019.

[9] S. Brossette, A. Escande, J. Vaillant, F. Keith, T. Moulard, and A. Kheddar. Integration of non-inclusive contacts in posture generation. In *IEEE International Conference on Intelligent Robots and Systems*, pages 933–938, 2014.

[10] B. Ponton, A. Herzog, S. Schaal, and L. Righetti. A convex model of humanoid momentum dynamics for multi-contact motion generation. In *IEEE-RAS International Conference on Humanoid Robots*, pages 842–849, 2016.

[11] J. Arreguit, S. Faraji, and A. Jan Ijspeert. Fast Multi-Contact Whole-Body Motion Planning with Limb Dynamics. *IEEE-RAS International Conference on Humanoid Robots*, pages 25–32, 2018.

[12] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, USA, 3rd edition, 2009.

[13] A. Escande, A. Kheddar, and S. Miossec. Planning contact points for humanoid robots. *Robotics and Autonomous Systems*, 61(5):428–442, 2013.

[14] P. Fernbach, S. Tonneau, A. Del Prete, and M. Taix. A kinodynamic steering-method for legged multi-contact locomotion. *IEEE International Conference on Intelligent Robots and Systems*, pages 3701–3707, 2017.

[15] A. Del Prete, S. Tonneau, and N. Mansard. Fast Algorithms to Test Robust Static Equilibrium for Legged Robots. In *IEEE International Conference on Robotics and Automation*, pages 1601–1607, 2016.

[16] O. Kanoun, F. Lamiraux, and P. Wieber. Kinematic control of redundant manipulators: generalizing the task priority framework to inequality tasks. *IEEE Transactions on Robotics*, 27(4):785–792, 2011.

[17] F. Kanehiro, F. Lamiraux, O. Kanoun, E. Yoshida, and J.P. Laumond. A Local Collision Avoidance Method for Non-strictly Convex Polyhedra. *Robotics: Science and Systems*, 2008.

[18] M. Benallegue, A. Escande, S. Miossec, and A. Kheddar. Fast C1 proximity queries using support mapping of sphere-torus-patches bounding volumes. In *IEEE International Conference on Robotics and Automation*, pages 483–488, 2009.

[19] 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*, 4(2):1431–1438, April 2019.

[20] S. Nakaoka. Choreonoid: Extensible virtual robot environment built on an integrated GUI framework. In *IEEE/SICE International Symposium on System Integration*, pages 79–85, 2012.

[21] M. Morisawa, M. Benallegue, R. Cisneros, I. Kumagai, A. Escande, K. Kaneko, and F. Kanehiro. Multi-Contact Stabilization of a Humanoid Robot for Realizing Dynamic Contact Transitions on Non-coplanar Surfaces. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2252–2258, 2019.

[22] Y. C. Lin, B. Ponton, L. Righetti, and D. Berenson. Efficient Humanoid Contact Planning using Learned Centroidal Dynamics Prediction. *IEEE International Conference on Robotics and Automation*, pages 5280–5286, 2019.