2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
September 27 - October 1, 2021. Prague, Czech Republic

2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 978-1-6654-1714-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/IROS51168.2021.9636218

Dynamic Humanoid Locomotion Over Rough Terrain With Streamlined Perception-Control Pipeline

Moonyoung Lee[1], Youngsun Kwon[2], Sebin Lee[2], JongHun Choe[1], Junyong Park[3], Hyobin Jeong[4], Yujin Heo[1], Min-su Kim[1], Jo Sungho[3], Sung-Eui Yoon[2], Jun-Ho Oh[1]

***Abstract*— Vision aided dynamic exploration on bipedal robots poses an integrated challenge for perception and control. Rapid walking motions as well as the vibrations caused by the landing-foot contact-force introduce critical uncertainty in the visual-inertial system, which can cause the robot to misplace its feet placing on complex terrains and even fall over. In this paper, we present a streamlined integration of an efficient geometric footstep planner and the corresponding walking controller for a humanoid robot to dynamically walk across rough terrain at speeds up to 0.3 m/s. To handle perception uncertainty that arises during dynamic locomotion, we present a geometric safety scoring method in our footstep planner to optimally select feasible path candidates. In addition, the real-time performance of the perception pipeline allows for reactive locomotion such as generating a new corresponding swing leg trajectory in mid-gait if a sudden change in the terrain is detected. The proposed perception-control pipeline is evaluated and demonstrated with real experiments using a full-scale humanoid to traverse across various terrains.**

I. INTRODUCTION

A primary benefit of a bipedal design over tradition wheeled platform is the potential for greater mobility similar to that of humans. Increased mobility in legged locomotion opens the possibility for bipedal robots to traverse unstructured environments such as in disaster response scenarios. Although bipedal locomotion provides extended maneuverability, it requires careful planning and control to safely walk across unstable terrains without falling. This poses an integrated challenge for the robot to perceive and plan feasible footsteps in a fraction of a second, especially with dynamic motions where the robot may have to adapt its swing-leg trajectory on-the-fly in order to safely recover from sudden changes in the terrain. More specifically, the robot must be able to perceive, plan,and react to the given environment in real-time without falling down. To do so, we propose an efficient sampling-based geometric planner for generating feasible footsteps, and a robust walking controller that can handle dynamic footstep updates in real-time. To safely account the sensor uncertainty and motion estimation error that arise during dynamic locomotion, we implement a geometric safety scoring method that allows for heuristic selection of an optimal candidate out of the multiple feasible paths. Overall, our main contributions are:

- Streamlined integration of the mapping, planning, and control pipeline for continuous locomotion over rough terrain at speeds up to 0.3 m/s, which is faster than comparable published works to the best of the author's knowledge.
- Dynamic on-the-fly replanning of the landing footstep position in middle of the leg swing-phase for reactive locomotion robust to sudden changes in terrain.
- Extensive experimental results of the proposed perception-control pipeline using a full-sized humanoid in various environments featured by stepping stones, dynamically movable stepping stones, or narrow path.

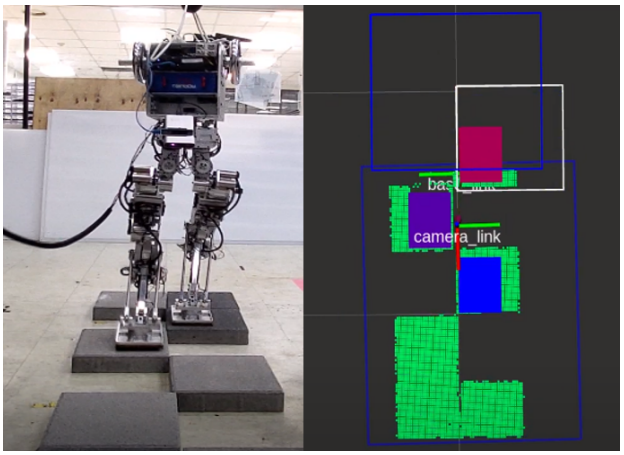


Fig. 1. Terrain reconstruction and footstep placements dynamically generated to traverse rough terrain. Terrain is represented as green octomap from planar regions collected from the depth camera. Feasible footsteps are continuously generated, where the change in color represents footstep sequence.

II. RELATED WORK

With recent availability of robust quadruped platforms, there has been much advancements in vision-aided dynamic exploration on legged robots [1]–[4]. We focus exclusively on bipedal research due to the fact that feasibility check in bipedal planners include a more complex foot surface area and orientation rather than simple point-contact commonly used in quadruped systems. One forerunner in this field [5] implemented an online footstep planner for the Honda ASIMO robot based on 2D A* search to navigate dynamic environments but utilized off-board motion capture system.

[1]Is with the Humanoid Robot Research Center, Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology, Daejeon, Korea jhoh@kaist.ac.kr
[2]Is with the Scalable Graphics, Vision & Robotics Lab, School of Computing, Korea Advanced Institute of Science and Technology, Korea
[3]Is with the Neuro-Machine Augmented Intelligence Laboratory, School of Computing, Korea Advanced Institute of Science and Technology, Korea
[4]Is with the Korea Atomic Energy Research Institute (KAERI), Korea

The work of [6] successfully implemented real-time perception and planning all on-board for stair-like terrains using stereo images by simplifying planar regions to straight edges with a line detector.

Our footstep planner resembles the geometric approach studied in [7]. This work differs from our work by first generating a 2D path to a known fixed global goal, and then individual footsteps along the path, mainly focusing on avoiding obstacles in 2D space. In contrast to navigating towards the global goal, the research in [8], [9] proposed continuous walking autonomy using short-horizon footstep planning within safe local regions to traverse over uneven terrain. Their approach, rather than discrete graph-based search, used continuous search space algorithm based on MIQP leading to optimal but long planning time of over 400 msec, which significantly reduced the overall pipeline.

The researches of [10], [11] are most similar to ours in that it utilizes discrete search technique with heuristically scored nodes over the generated 3D map of the terrain. However, there is no reactive modification during the leg swing phase once a feasible footstep is generated. Our approach differs in that our pipeline allows for on-the-fly replanning and dynamically updating the landing footstep in order to react to sudden detected changes in the terrain. A very recent work [12] similarly showed highly dynamic performances to quickly generate footsteps upon push-recoveries on rough terrain. This work simplified the visual processing by utilizing color features to determine steppable regions.

Our footstep planner generates feasible footsteps determined as best effort within a fixed sample time of 5 milliseconds, or one control cycle. Best effort here refers to scoring possible footstep array candidates, prioritizing candidates with higher safety score based on distance from the edges of the terrain.

III. System Overview

In this work, we use the Gazelle legged platform [13], which is a lightweight 13-DOF bipedal robot that is capable of walking at relatively high speeds with 30 cm stride with 0.5 sec step time (0.6 m/s). Previous works on this platform include dynamic walking without any perception capability [14] or push recoveries on flat grounds [15]. This is the first work to extend dynamic locomotion with vision-aided footstep planner to autonomously traverse rough terrain. To account for the new visual-inertial sensor suite, we modify the robot configuration to rotate the waist 180°so that the robot appears to walk backwards. While this new configuration has no affect on walking performance, it prevents the robot knees from obscuring the FOV of the depth camera used for terrain mapping.

The software framework is divided among two PCs for vision and motion control as shown in Figure 2. The vision PC utilizes ROS to handle camera data, such as the Intel T265 for high-speed VIO at 200Hz, and the Microsoft Kinect Azure for time-of-flight depth camera at 30Hz. The Motion PC runs on the custom PODO API software framework [16] to leverage real-time performance, with whole-body motion

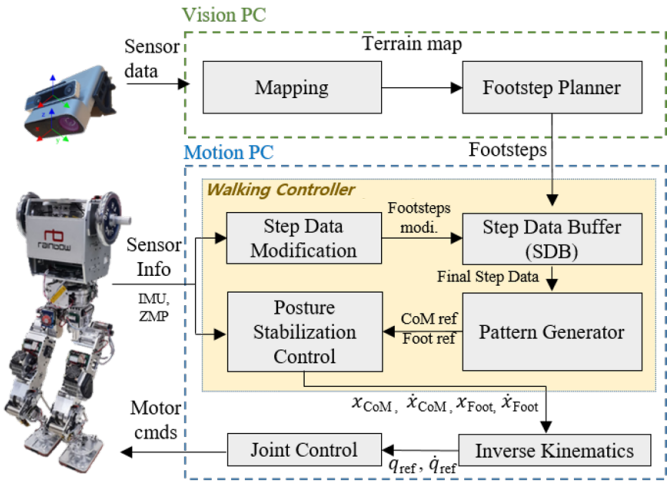


Fig. 2. Overall software diagram illustrating the flow of terrain map and walking controller modules used in the system.

controller iterating at 500 Hz to accommodate the generated footstep data. This integration of ROS-PODO framework [17], which was previously designed for a wheeled robot in our previous work, is extended for interfacing with a legged system.

IV. Terrain Mapping

The terrain map is regenerated at every frame to allow fast updates of the current scene. Instead of using Simultaneous Localization and Mapping (SLAM) to create a global map expanding over the whole environment, we focus on creating a local map using only the most recent depth image. This strategy alleviates the drifting error caused by perception uncertainty, which can otherwise grow to be significant without loop-closure corrections, as well as leading to sparser local map that can more quickly detect dynamic changes in the environment.

The overall procedure of our terrain mapping process is shown in Fig. 3. The raw point cloud of the terrain, obtained through the camera, is first transformed to the robots base frame to fit the orientation of the world. The point cloud is then down-sampled and cropped to remove points outside the region of interest. We choose a 1 meter wide and 2 meter long horizontal box as our default region of interest.

These parameters are adjusted to fit the resolution and computation time trade-off depending on the traversing environment. RANSAC is used to segment the major planes

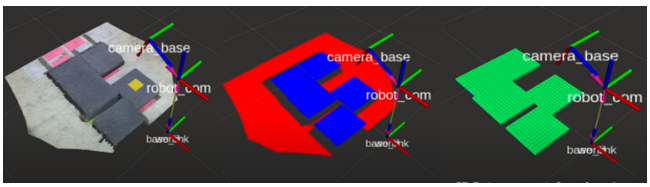
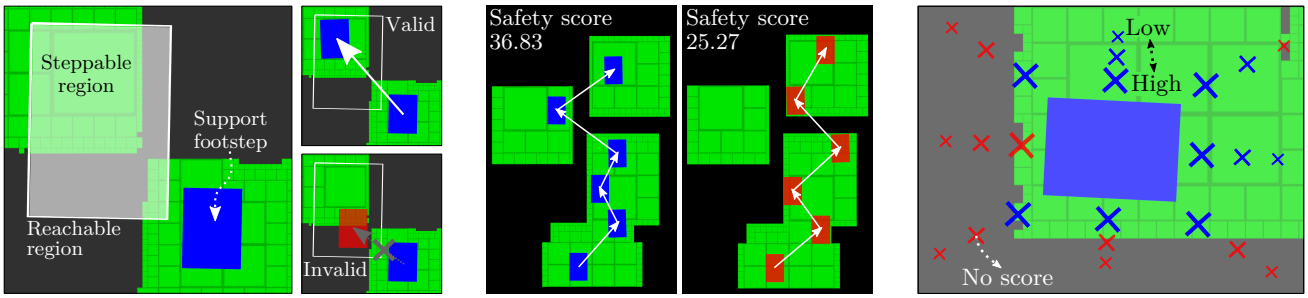


Fig. 3. Pictorial summary of obtaining the terrain map. Raw point cloud of the terrain (left), major planes segmented using RANSAC (middle), 3D octomap constructed for traversable regions (right).



(a) Footstep sampling (b) Footstep paths with different safety scores (c) Test points for computing a safety score

Fig. 4. These figures represent the main concept of our sampling-based footstep planner. (a) Our approach makes a footstep sample within the reachable region randomly and connects a valid sample to a support footstep. (b) The sampling process constructs a tree structure consisting of multiple footstep paths. This example shows two different paths colored by blue and red. (c) Our method computes the safety score using the multiple test points of a footstep. The blue and red X marks represent the points that receives some scores and no score, respectively. Different sizes of marks indicate various weight values of the safety scores.

Algorithm 1 FOOTSTEP PLANNING

Input: A steppable geometry G, a start footstep q_{init}
Output: A footstep path

$T \leftarrow \{q_{init}\}$
while termination condition is not satisfied **do**
  $q_{sup} \leftarrow RandomSupportFootstep(T)$
  $q_{rand} \leftarrow RandomFootstep(q_{sup})$
  if $ValidityTest(q_{rand}, G)$ **then**
    Connect q_{sup} with q_{rand}
    Insert q_{rand} to T
  end if
end while
$P \leftarrow FootstepPathCandidates(T)$
return $BestFootstepPath(P, G)$

in the point cloud. Planes outside an angular range of the plane of the floor are removed to ignore non-terrain planes such as walls. Finally, the planes classified as traversable are chosen and converted into an octomap for the planner. Overall, a local 3D octomap of the terrain is generated with 1cm resolution of rates up to 15 Hz.

V. FOOTSTEP PLANNING

Utilizing the continuously updated terrain map, we generate feasible footsteps while preserving the real-time performance for re-planning.

As shown in Fig. 4-(a), we define the steppable region as the intersection of two parts: 1) geometric representations of steppable objects and 2) pre-defined reachable region where the robot can kinematically reach by taking one step from the support footstep. Our framework obtains the geometry of steppable objects from the vision system as a grid-based volumetric representation (Fig. 3), and uses kinematic conditions fine-tuned with empirical results to obtain the reachable region with respect to the supporting footstep position. Computing the explicit representation of the steppable region, however, becomes an overhead in a real-time system.

We, therefore, adopt a sampling approach instead of explicitly determining the steppable region as shown in (Alg. 1). The proposed footstep planner randomly samples a next footstep within the reachable region of the support footstep.Then our method then checks the geometric validity of the sampled footstep as shown in Fig. 4-(a). The resulting volumetric representation of the footstep resides within the geometry of the steppable environment, indicating a geometrically and kinematically feasible footstep. Our approach connects the valid footstep to the support footstep but discards invalid samples. As shown in Fig. 4-(b), repeating this random sampling process generates multiple viable footstep paths in a tree structure. Each path of the footstep tree consists of the sequential footsteps linked from a root to a leaf.

With multiple feasible paths generated, the proposed planner chooses the best candidate based on safety score using geometric information of the foot and steppable region. Safety scoring accommodates for the perception uncertainty in the terrain map or motion estimation error by quantifying and selecting the safest path furthest from the edges of the terrain. For example, two footstep paths shown in Fig. 4-(b) are computed with different safety scores. The blue footstep path, deemed the safer path, receives a higher safety score than the red path because the generated footsteps in the blue path are located further from the terrain edges. Our method heuristically computes the safety of a footstep using its surrounding test points as shown in (Fig. 4-(c)), efficiently checking whether the footstep is located nearby the edges.

Lastly, our method selects the footstep path having the best safety score:

$$p^* = \underset{p \in P}{\operatorname{argmax}}\ f_{safety}(p), \tag{1}$$

where $P = \{p_1, p_2, \ldots, p_N\}$ indicates a set of the N footstep paths that our planner generates. We design the function $f_{safety}(p)$ that computes a safety score of the footstep path p consisting of the M footsteps, $p = \{q_1, q_2, \ldots, q_M\}$:

$$f_{safety}(p) = \sum_{m=1}^{M} \sum_{k=1}^{K} w_k^m \, \mathbb{I}(\mathbf{x}_k^m \text{ is onto } G), \tag{2}$$

where $\mathbf{x}_k^m$ represents k-th test point of m-th footstep associated with the footstep path p.

If a test point $\mathbf{x}_k^m$ is onto the steppable objects G, our planner makes a score as a weight w_k^m depending on the distance between the center of footstep and the test point; e.g., we can use the radial basis function (RBF) kernel. Otherwise, we give no score value at the test point. Note that the indicator function $\mathbb{I}(\cdot)$ in Eq. 2 returns 1 if the input state is true and 0 if not. Based on the scoring policy, our method selects the final footstep path with the highest safety score, consisting of the geometry-aware safe footsteps.

Finally, the planner terminates when robot's odometry approximately reaches a desired distance or a desired number of footsteps from the starting point. As we focused on local perception and footstep generation, we did not implement a global navigation strategy towards a specific goal. Rather, we embedded a forward-driven-bias into our planner to continuously generate steps until the robot reaches the end of the course. This approach of planning only in local view also helped speed up the pipeline by reducing the maximum number of feasible steps in comparison to available methods that account for global paths.

VI. Walking Pattern-Generator

Given the collision-free and kinematically admissible geometric path, we generate CoM and foot trajectories over time that maintains stability of the compliant LIPM. The overall walking-pattern generator used for Gazelle platform is composed of CoM & footstep trajectory generation and posture-stabilization-controller. Walking patterns are generated based on the robot model, and the controllers are constructed based on sensor feedback. This section will briefly introduce the walking pattern generation method and the required stabilization controller.

A. CoM & Footstep Trajectory

The pattern-generator is extended upon Preview Control method [18] in order to create robot's CoM trajectories from determined footsteps from our planner. For the preview control to be stable, the controller requires information of the next two footsteps in advance. Therefore, for dynamic walking, the planner is bounded to generate a minimum footstep array length of two.

To accommodate on-the-fly replanning of the landing footstep, we utilize an efficient vector-type struct called Step Data Buffer (SDB) to store current and future footstep positions and timing. The SDB can be dynamically modified any time during the gait cycle by the planner or balancing controller, even in middle of the swing-foot phase.This is enabled through our specific implementation of a short-cycle preview controller [19] to stably re-create CoM trajectories at control interval of 500Hz.

B. Posture-Stabilization Controller

Main purpose of posture stabilization control is to achieve stable walk according to the footstep information in SDB. Posture stabilization is implemented using contact force control and vibration damping control [13] summarized below.

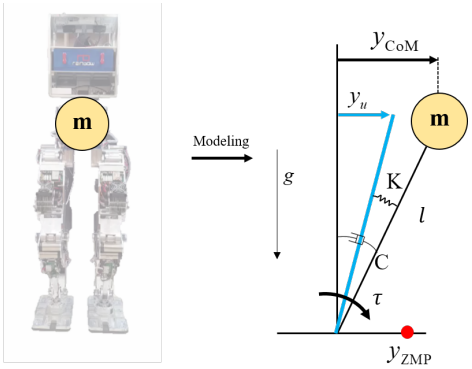


Fig. 5. Compliant-LIPM modeling of the real robot. Compliant-LIPM is preferred over the standard LIPM model for improved controller performance because in reality, the robot does not behave exactly like a ridged body due to joint, structure stiffness and compliance between the ground and the foot.

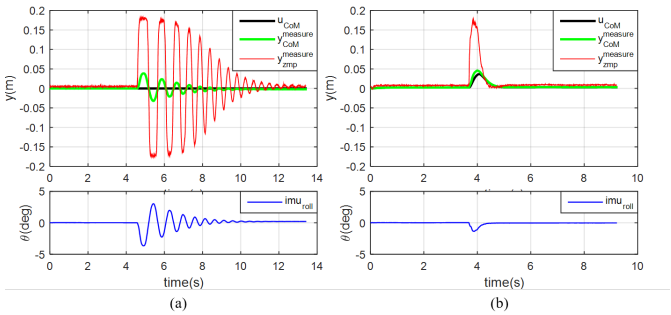


Fig. 6. CoM damping control performance tested through robot's response to being pushed in the lateral direction. Graphs show the sensor values and the CoM position of the robot. (a) Without control, the body vibrates upon push because of high stiffness. (b) The damping control raises system damping and becomes robust to impact.

1) Vibration Damping Control: When a sudden external impact is applied to a standing robot, the robot will oscillate with a damped sinusoidal form because of the stiffness of each leg joint, including the stiffness of the structure, and the rubber pad attached under the feet. Such oscillation can negatively affect perception accuracy of visual sensors due to shaking motions and blurred images. By walking, however, the robot receives such external impact continuously due to contact force between its landing foot and hard ground. Although the vibrations from these impact will damp out after a certain period of time, the damping is not enough to stabilize the robot during the continuous walking. Therefore, it requires to increase the damping and decrease the stiffness of the system. We used simple LIPM model for the walking pattern generation; here compliant-LIPM model is be applied for damping controller (Fig. 5).

From the state space form modeling of Compliant-LIPM that has input of y_u and output of y_{ZMP}, the full state feedback controller can be constructed. The full state feedback is designed to specify new stiffness and damping of the system, and estimated states are used for feedback. Fig. 6 presents the result of the vibration damping controller. Detailed description about modeling and state estimation are

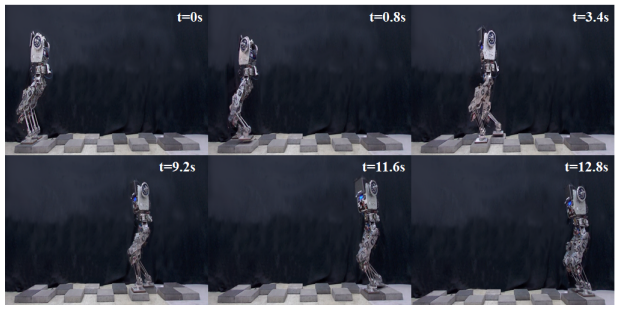


Fig. 7. Terrain mapping, footstep planning, and trajectory generation pipeline integrated to walk continuously over uneven terrain at high speeds of 0.3 m/s.

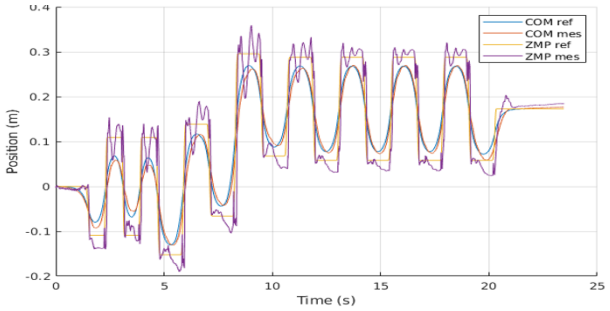


Fig. 8. Measured ZMP and CoM track the desired trajectory created by the walking pattern generator resulting in stable balancing of the robot.

available in [13].

2) Contact Force Control: Foot contact force control is necessary for the robot to walk on uneven ground or in the case of perception error of the ground height. Such instance results in unexpected, early landing or late landing of the stepping foot, causing the robot to fall without proper stabilization. To solve this problem, the reference force and torque of the foot should be obtained from the reference trajectory of the robot and the current state. In [14], we proposed a method for generating the reference torque of a foot using capture point feedback. The error between the reference capture point and the measured capture point led us to obtain the desired ZMP (cZMP). This cZMP was used to generate the ankle torque reference.

Foot force control was performed in the vertical direction using leg length control. The cZMP from the capture point feedback was used to generate the reference vertical force of both feet. According to relative position of cZMP about foot position, the weight to be distributed to both feet is determined. This was then transformed to the desired length of both legs. Detailed description of ankle torque and force control are in [20].

VII. EXPERIMENTAL RESULTS

We conduct various experiments to verify the performance of our streamlined perception-control pipeline. The first experiment was of the robot traversing scattered stepping stones for three meters without stopping as our validity test of the proposed system. We then present results of dynamic locomotion of the robot updating swing-leg trajectories on-the-fly as the terrain changes, and how the controller responds to stabilize the body. We lastly conduct experiments on a narrow path the accuracy of the local mapping and footstep planner. Of 20 repeated trials in the stepping stone and narrow path, the robot successfully traversed 100% and 95% of the time. The dynamic locomotion showed lower repeatability rate of 60% as discussed below.

The computation and performance, running on the mini PC (6th gen i7 at 2.6GHz on 4 cores), of our proposed system can be summarized below. Note that these modules ran asynchronously on separate threads, and in the slowest-case scenario where each module is fed serially, the total pipeline computation is approximately 0.12 sec. Time for each module are as below:

- Depth Image acquisition: 33 msec (30 FPS, Wide FOV)
- Planar region segmentation and mapping: 67 msec (15Hz, 1cm resolution grid)
- Footstep planning: 5 msec (max 4 footsteps)
- Communication: 10 msec (ROS and PODO between 2 PCs)

A. System Performance Test

The first experiment was traversing across three meters of scattered stepping stones displaced up to 0.3m apart for each footstep as seen in Fig. 7. Using the proposed strategy, the robot successfully traversed across at a high speed of 0.3 m/s with 100% success rate (tested 20 times). This default test scene was simplest compared to following experiments because the scene environment was static and thus did not need on-the-fly replanning of foot trajectories. In addition, the footstep positions generated from the planner were often located at the center of the stepping stone terrain, showing successful selection of path candidate with highest safety score as implemented. With planner that prioritized footstep positions at center of the terrain, traversing across stepping stones were more tolerant to perception error. For example, given the width of the stone and robot foot, the foot width covered only 50% of the stone while the foot width covered 85% of the narrow path below.

B. Dynamic Walking on Changing Terrain

As an extension to the previous test, the dynamic locomotion capability was tested by introducing a terrain disturbance as the robot is walking. During the robot's swing phase, the position of the stone where the immediate landing foot would have landed was changed approximately 8cm by pulling on a rope. Of 10 repeated trials, the robot reacted to the disturbance only 60% of the time. Low rate is reflected by the low repeatability of the dynamic disturbance in the test setup, where the rope would be pulled either too far or too late for the robot to adjust in swing-leg trajectory. As seen in Fig. 9 (a), during the short time-frame of the robot's swing phase, the pipeline was able to detect change in terrain, and updated its swing-leg trajectory under 0.12 second to readjust its landing footstep. Without such dynamic replanning capability, as shown in Fig. 9 (b), the robot would

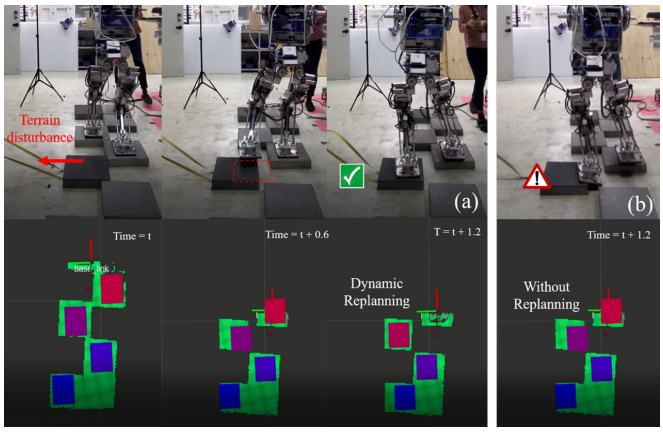


Fig. 9. (a) Terrain disturbance is introduced during the robot's swing leg phase. Dynamic replanning enables updated feasible footsteps on-the-fly. (b) Without replanning, the robot is susceptible to terrain disturbances and steps over the edge.

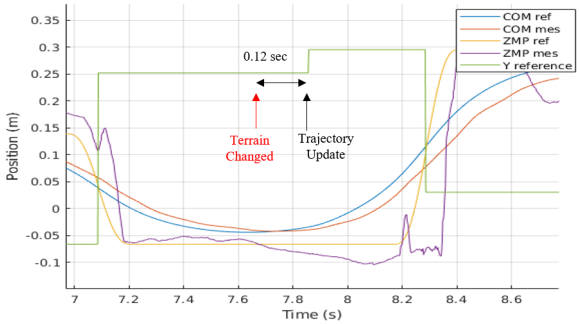


Fig. 10. Trajectory of the swing-foot updated on-the-fly within 0.12 seconds after terrain change is introduced. ZMP and CoM trajectory is dynamically modified, yet is continuous and stable.

be susceptible to sudden terrain changes because it would not be able to update its landing footstep position on-the-fly, causing the robot to fall over. The upper constraint on how far the robot could readjust its swing-leg trajectory on-the-fly was experimentally determined to be approximately 0.5m euclidean distance from its previous footstep. For test values exceeding that bound, the robot would fall over in mid foot-swing in which the measured ZMP could not closely track the newly generated reframe ZMP. For sudden terrain change below 0.1m, the streamlined perception could generate newly feasible trajectory within 0.12sec, where the new CoM and ZMP reference would smoothly update with previous trajectory as shown in Fig. 10. As such, we set upper bound on how much the robot could adjust its landing foot position in mid-swing without falling over.

C. Precision Walking on Narrow Path

In our final experiment, we verify the precision of our perception as well as our controller. To do so, we set up an environment consisting of two platforms spanned by a narrow path of length three meters and width 0.3m, which is approximately half the width of the robot Fig. 11. In contrast to previous stepping stone walking pattern, the narrow path prevents the robot from placing both feet side-by-side, and constrains the robot to place one foot in front of the other. The small subset of kinematically reachable area and geometrically steppable region requires significant precision of the pipeline in order to not fall over the edge. To do so, the map was densely generated by increasing the resolution of the map to 0.5cm while narrowing the region of interest in octomap in order to maintain mapping rate. The robot successfully traversed narrow path with 90% accuracy (tested 20 times), the two times failing due experimental mishandling where the robot's tethered power-cord got stuck on the narrow path setup.

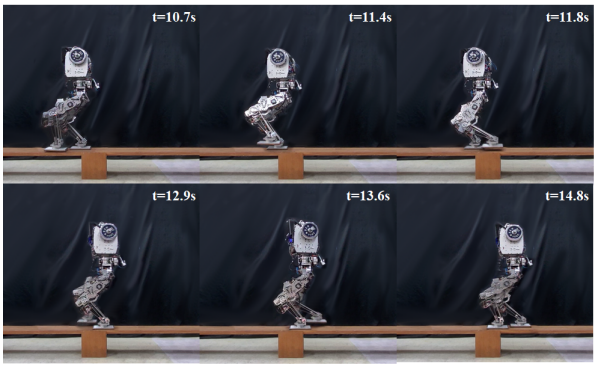


Fig. 11. Walking continuously over narrow beam path to verify perception and motion precision

VIII. CONCLUSION

We present an efficient geometric footstep planner and the corresponding walking controller that enables dynamic humanoid locomotion over uneven terrain. We show dynamic locomotion first through the real-time perception pipeline, and second through the on-the-fly re-planning of the landing footstep position in middle of the swing phase during the robot gait cycle. The proposed perception-control pipeline is demonstrated on a full-scale humanoid using only on-board sensors and computing. For evaluation, we conduct experiments where the robot traverses at high speeds of 0.3 m/s across uneven terrains including static stepping stones, dynamically movable stepping stone, and narrow path. Our current perception pipeline handles only uneven terrain with flat surfaces. For future work, we aim to cover terrains with varying heights and surface normal. We believe the current framework will extend nicely without accruing significant computation time.

IX. ACKNOWLEDGEMENT

This work was supported by the Technology Innovation Program (or Industrial Strategic Technology Development Program (0070171, Development of core technology for advanced locomotion/manipulation based on high-speed/power robot platform and robot intelligence) funded By the Ministry of Trade, industry & Energy(MI, Korea).

REFERENCES

[1] D. Kim, J. Carballo, J. Di Carlo, B. Katz, G. Bledt, B. Lim, and S. Kim, "Vision aided dynamic exploration of unstructured terrain with a small-scale quadruped robot," in *IEEE International Conference on Robotics and Automation*, 2020.

[2] T. Dudzik, M. Chignoli, B. Lim, A. Miller, D. Kim, and S. Kim, "Robust autonomous navigation of a small-scale quadruped robot in real-world environments," in *IEEE International Conference on Intelligent Robots and Systems*, 2020.

[3] P. Fankhauser, M. Bjelonic, D. Bellicoso, T. Miki, and M. Hutter, "Robust rough-terrain locomotion with a quadrupedal robot," in *IEEE International Conference on Robotics and Automation*, 2018.

[4] A. Winkler, C. Mastalli, I. Havoutis, M. Focchi, D. Caldwell, and C. Semini, "Planning and execution of dynamic whole-body locomotion for a hydraulic quadruped on challenging terrain," in *IEEE International Conference on Robotics and Automation*, 2015.

[5] J. Chestnutt, M. Lau, G. Cheung, J. Kuffner, J. Hodgins, and T. Kanade, "Footstep planning for the honda asimo humanoid," in *IEEE International Conference on Robotics and Automation*, 2005.

[6] M. Asatani, S. Sugimoto, and M. Okutomi, "Real-time step edge estimation using stereo images for biped robot," in *IEEE International Conference on Intelligent Robots and Systems*, 2011.

[7] P. Karkowski and M. Bennewitz, "Real-time footstep planning using a geometric approach," in *IEEE International Conference on Robotics and Automation*, 2016.

[8] M. Fallon, P. Marion, R. Deits, T. Whelan, M. Antone, J. McDonald, and R. Tedrake, "Continuous humanoid locomotion over uneven terrain using stereo fusion," in *IEEE International Conference on Humanoid Robots*, 2015.

[9] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *IEEE International Conference on Humanoid Robots*, 2014.

[10] R. Griffin, G. Wiedebach, S. McCrory, S. Bertrand, I. Lee, and J. Pratt, "Footstep planning for autonomous walking over rough terrain," in *IEEE International Conference on Humanoid Robots*, 2019.

[11] S. Bertrand, I. Lee, M. B., D. Calvert, J. Pratt, and R. Griffin, "Detecting usable planar regions for legged robot locomotion," in *IEEE International Conference on Intelligent Robots and Systems*, 2020.

[12] Y. Kojio, Y. Omori, K. Kojima, F. Sugai, Y. Kakiuchi, K. Okada, and M. Inaba, "Footstep modification including step time and angular momentum under disturbances on sparse footholds," in *IEEE Robotics and Automation Letters, VOL. 5, NO. 3*, July 2020.

[13] H. Jeong, K. K. Lee, W. Kim, I. Lee, , and J.-H. Oh, "Design and control of the rapid legged platform gazelle," in *Mechatronics 66, 102319*, January 2020.

[14] H. Jeong, I. Lee, J. Oh, K. K. Lee, and J.-H. Oh, "A robust walking controller based on online optimization of ankle, hip, and stepping strategies," in *IEEE TRANSACTIONS ON ROBOTICS, VOL. 35, NO. 6*, December 2019.

[15] H. Jeong, J.-H. Kim, O. Sim, and J.-H. Oh, "Avoiding obstacles during push recovery using real-time vision feedback," in *IEEE International Conference on Robotics and Automation*, 2019.

[16] M. Lee, Y. Heo, S. Cho, H. Park, and J. Oh, "Motion generation interface of ros to podo software framework for wheeled humanoid robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, in press 2019.

[17] M. Lee, Y. Heo, J. Park, H. Yang, P. Benz, H. Jang, H. Park, I. Kweon, and J. Oh, "Fast perception, planning, and execution for a robotic butler: Wheeled humanoid m-hubo," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, in press 2019.

[18] 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*, 2003.

[19] K. Nishiwaki and S. Kagami, "High frequency walking pattern generation based on preview control of zmp," in *IEEE International Conference on Robotics and Automation*, 2006.

[20] H. Jeong, O. Sim, H. Bae, K. Lee, J. Oh, and J.-H. Oh, "Design and control of the rapid legged platform gazelle," in *EEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.