

Loco-Manipulation Planning for Legged Robots: Offline and Online Strategies

Luca Rossini^{1,2}, Paolo Ferrari³, Francesco Ruscelli¹,
Arturo Laurenzi¹, Nikos G. Tsagarakis¹, Enrico Mingo Hoffman⁴

Abstract—The deployment of robots within realistic environments requires the capability to plan and refine the loco-manipulation trajectories on the fly to avoid unexpected interactions with a dynamic environment. This extended abstract provides a pipeline to offline plan a configuration space global trajectory based on a randomized strategy, and to online locally refine it depending on any change of the dynamic environment and the robot state. The offline planner directly plans in the contact space, and additionally seeks for whole-body feasible configurations compliant with the sampled contact states. The planned trajectory, made by a discrete set of contacts and configurations, can be seen as a graph and it can be online refined during the execution of the global trajectory. The online refinement is carried out by a graph optimization planner exploiting visual information. It locally acts on the global initial plan to account for possible changes in the environment. While the offline planner is a concluded work, tested on the humanoid COMAN+, the online local planner is still a work-in-progress which has been tested on a reduced model of the CENTAURO robot to avoid dynamic and static obstacles interfering with a wheeled motion task. Both the COMAN+ and the CENTAURO robots have been designed at the Italian Institute of Technology (IIT).

Paper Type – Recent Work [1][2]

I. INTRODUCTION

Humanoids have the potential to accomplish tasks that requires moving in complex and confined spaces by braking and establishing multiple contacts with the environment. Examples of these *multi-contact loco-manipulation tasks* include crawling, climbing a ladder and standing up using vertical wall (Fig. 1). Additionally, introducing collaborative robots alongside human beings in a domestic or working environment requires the capability of the robot to detect any change in the surrounding environment and react to that on the fly while executing a task.

To effectively fulfill these tasks, the robot must be able to autonomously decide and execute appropriate motion that respect several crucial constraints such as collision avoidance, balance and kinematic/dynamic limitations. This means addressing a complete Multi-Contact Planning and Control (MCPC) problem. In this extended abstract, we focus specifically on the planning aspects of this problem.

¹ Humanoid and Human Centered Mechatronics (HHCM) Lab, Istituto Italiano di Tecnologia (IIT), Via Morego 30, 16163 Genova, Italy. E-mail: name.surname@iit.it

² Dipartimento di Informatica, Bioingegneria, Robotica e Ingegneria dei Sistemi (DIBRIS), Università di Genova, Via Opera Pia 13, 16145 Genova, Italy.

³ Dipartimento di Ingegneria Informatica, Automatica e Gestionale (DIAG), Sapienza Università di Roma, Via Ariosto 25, 00185 Roma, Italy.

⁴ PAL Robotics, Carrer de Pujades, 77-79, 08005 Barcelona, Spain.



Fig. 1: The COMAN+ humanoid robot standing up taking advantage of a vertical wall placed in front of it.

Sample-based global planners revealed to be very efficient over the last decades, and they can easily be combined with perception algorithms avoiding any simplification of the environment [3]. In the case of dynamic environment, one possibility would be to continuously re-plan the global trajectory every time a difference in the environment is perceived. However, sample-based planners revealed to be strongly influenced by the complexity of the robot, and its computational cost increases exponentially depending on the number of degrees of freedom (DoFs), such as for legged robots. As a consequence of their massive computation cost, these methodologies are prevented to be used in real-time application (a.k.a. *curse of dimensionality*). On the other side, optimization-based motion planners can be employed in real-time but they can easily stuck in local minima getting impossible to find an optimal trajectory.

A possible solution is to couple a sample-based planner, generating offline a global initial trajectory, with an optimization-based motion planner exploiting visual information that acts locally while the robot executes the global trajectory (see Fig. 2). In this way, the local planner is globally guided by the global planner avoiding any local minima, while guaranteeing a computational efficiency that allows an online implementation during the execution of the assigned task. These kind of problems follow a common pipeline made by three modules: *perception*, *planning* and *control* modules [4].

In this extended abstract, we present preliminary results of our offline and online planning strategies for loco-

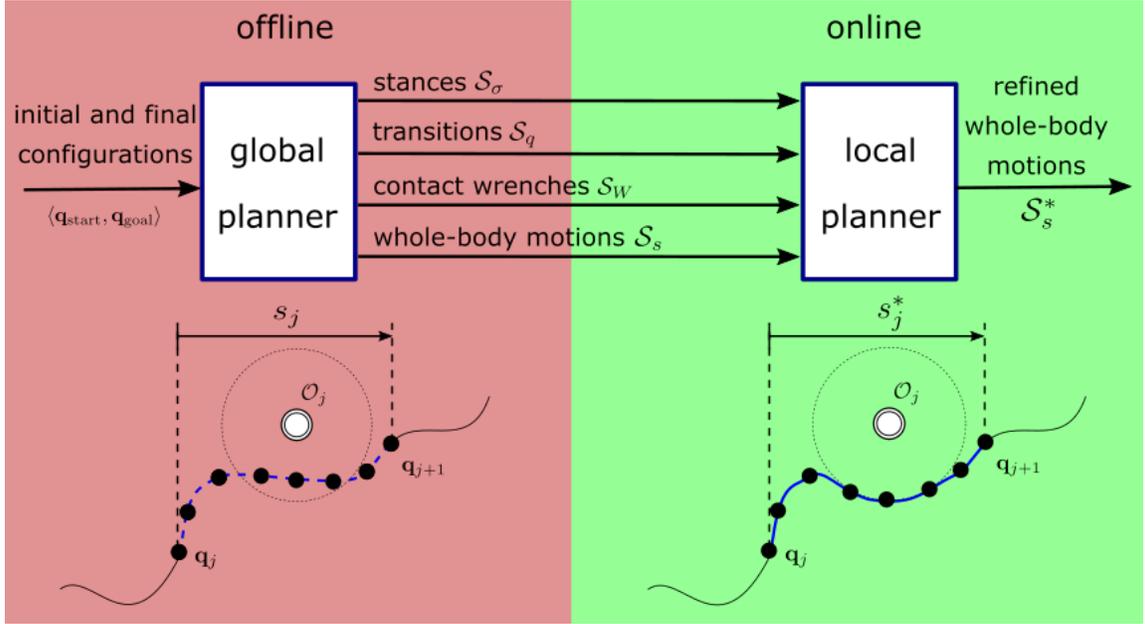


Fig. 2: Combination of global and local planning. The global planner works offline and provides an initial guess to the online local planner (blue dashed line on the left). While executing the trajectory, the local planner refines it to recover any infeasibility, generating a new feasible motion depending on the environment state. \mathcal{O}_j is a generic obstacle interfering with the global plan computed offline.

manipulation tasks, applied to legged robots. The perception module uses state-of-art algorithms to process a dense point cloud generated by the Intel® Realsense™ D435i depth camera embedded on the robot. The planner layer is accompanied by a control layer which moves the robot through an impedance controller using the references computed by the planner. At the current stage, the offline planning strategy has been tested on the real humanoid COMAN+ using a dedicated control layer that executes the planned trajectory once available. Recently, we started developing the online planning strategy, which has been successfully tested on the CENTAURO robot executing a wheeled-locomotion task. We are currently working towards the design of a unified planning and control framework, which is paramount skill for the success deployment and use of legged robots in challenging environments.

II. OFFLINE PLANNING

In this section, we consider the problem of generating appropriate motions for a torque-controlled humanoid robot that is assigned a multi-contact loco-manipulation task, i.e., a task that requires the robot to move within a *static* and *completely known* environment by repeatedly establishing and breaking multiple, non-coplanar contacts.

To solve such problem, we propose a offline planning scheme consisting of the two sequential sub-planner described in the following.

The *stance planner* is in charge of finding three sequences

$$\begin{aligned} \mathcal{S}_\sigma &= \{\sigma_0, \dots, \sigma_N\}, \\ \mathcal{S}_q &= \{\mathbf{q}_0, \dots, \mathbf{q}_N\}, \\ \mathcal{S}_W &= \{\mathbf{W}_{c,0}, \dots, \mathbf{W}_{c,N}\}, \end{aligned}$$

where \mathcal{S}_σ is the sequence of $N + 1$ stances leading to the desired final stance σ^{fin} , i.e., $\sigma_N = \sigma^{\text{fin}}$, while \mathcal{S}_q and \mathcal{S}_W are sequences of associated transitions [5] and contact wrenches, respectively. In particular, the generic stance $\sigma_j \in \mathcal{S}_\sigma$ is a set of contacts (each one specifying the pose of a certain end-effector in contact with the environment), transition $\mathbf{q}_j \in \mathcal{S}_q$ is a feasible configuration for both σ_{j-1} and σ_j , vector $\mathbf{W}_{c,j} \in \mathcal{S}_W$ collects the contact wrenches that guarantee static balance at \mathbf{q}_j using the contacts specified by σ_j . To produce the sequences \mathcal{S}_σ , \mathcal{S}_q and \mathcal{S}_W , our stance planner applies a RRT-like strategy; it iteratively constructs a tree \mathcal{T} in the search space, where a vertex $v = \langle \sigma, \mathbf{q}, \mathbf{W}_c \rangle$ consists of a stance σ , a configuration \mathbf{q} , and a vector \mathbf{W}_c of contact wrenches, while an edge going from vertex v to vertex v' indicates that \mathbf{q}' is a transition from σ and σ' . Transitions are efficiently generated during the planning process using the method presented in [6].

The *whole-body planner*, for each pair of consecutive configurations \mathbf{q}_j and \mathbf{q}_{j+1} ($j = 0, \dots, N - 1$) belonging to \mathcal{S}_q , computes a feasible whole-body motion s_j , i.e., a configuration space trajectory that connects the two and has duration δ_j . Then, the result of this planner consists in a sequence of motions

$$\mathcal{S}_s = \{s_0, \dots, s_{N-1}\}.$$

To produce the generic motion $s_j \in \mathcal{S}_s$, our whole-body planner adopts the two-stage approach presented in [7]: first, a path consisting of a sequence $\mathcal{S}_{q,j}$ of feasible configurations joining \mathbf{q}_j to \mathbf{q}_{j+1} is found using the AtlasRRT* algorithm [8]; then, a trajectory s_j interpolating the configurations in $\mathcal{S}_{q,j}$ is computed, simultaneously determining its duration δ_j , via optimization.

TABLE I: Averaged performance data of the stance planner.

Task	Planning time (s)	Transition generation time (s)	Number of iterations	Number of vertices in \mathcal{T}	Number of stances in \mathcal{S}_σ
Ladder climbing	43.60	37.64	1926.14	205.34	39.04
Parallel walls climbing	175.37	171.01	1557.37	151.28	44.12
Quadrupedal walking	62.69	55.99	2540.10	228.32	53.19
Standing up	7.56	6.02	459.24	62.19	17.00

To actually execute the planned motions, we have incorporated the presented offline planning scheme into a complete MCPC framework. This includes a control layer that is responsible for generating online the torque commands $\bar{\tau}$ for the humanoid actuators to execute the planned sequence \mathcal{S}_s of whole-body motions throughout the planned sequence \mathcal{S}_σ of stances.

III. ONLINE PLANNING

To produce the global configuration space trajectory \mathcal{S}_f , the offline planning scheme previously described assumes that the environment is static. However, a typical working environment for legged robots, is likely to change during the execution of an assigned task, and the robot must be able to rapidly adapt to it to avoid unexpected collisions. To this end, we designed a configuration space local planner based on graph optimization that allows to store the complete trajectory and efficiently perform local modifications whenever needed. Indeed, the online planner possibly refines the motion s_j to be executed, instead of replanning the global motion \mathcal{S}_s , avoiding useless optimizations of future motions that are likely to be affected by further changes of the environment. Additionally, the solver avoids to get stuck in local minima being guided by the global information coming from the offline planner. Using the same constraints both for the offline and online planner, they will generate whole-body trajectories projected on the same manifold, thus guaranteeing the output of the offline planner to be feasible also for the online one.

The idea behind the proposed methodology is based on the representation of a global plan as a *hyper graph* where each vertex is a configuration, while each hyper edge represents the error term associated to a generic constraint connecting a certain number of vertices. The aim of this optimization is to move the vertices inside the feasible configuration space to minimize the sum of the considered error terms.

We discretize the trajectory s_j into a sequence \mathcal{Q}_j of equispaced configurations. Then, a vertex is created in correspondence of each configuration $q \in \mathcal{Q}_j$, while a set of edges is created to build connections between them. In particular, the k -th edge will connect a sub-sequence of vertices $\mathcal{Q}_{j,k} \subset \mathcal{Q}_j$. Edges involve joint limits, velocity limits, balance, and collision avoidance as well as a cost term to track the global plan. A continuous error function $e_k(\mathcal{Q}_{j,k})$ is assigned to each edge to penalize those solutions that violate the constraints. At each iteration, the planner locally refines the global trajectory solving the Non-Linear

Least Squared Optimization problem (NLLSO):

$$\mathcal{Q}_j^* = \underset{\mathcal{Q}_j}{\operatorname{argmin}} F(\mathcal{Q}_j) \quad (1)$$

where the cost function is

$$F(\mathcal{Q}_j) = \sum_{k=0}^m e_k^T(\mathcal{Q}_{j,k}) \Omega_k e_k(\mathcal{Q}_{j,k}) = \sum_{k=0}^m \mathbf{F}_k(\mathcal{Q}_{j,k})$$

with Ω_k being a diagonal square weight matrix having the same number of rows and columns as the associated error function $e_k(\mathcal{Q}_{j,k})$. A solution to (1) can be found using the Levenberg-Marquardt algorithm which finds the optimal step size $\Delta \mathcal{Q}_j^*$

$$(\mathbf{H} + \lambda \mathbf{I}) \Delta \mathcal{Q}_j^* = -\mathbf{b} \quad (2)$$

with λ being a damping factor used to control the step size $\Delta \mathcal{Q}_j^*$ on non-linear surfaces, $\mathbf{b} = \sum_{k=0}^m e_k^T \Omega_k \mathbf{J}_k$, and $\mathbf{H} = \sum_{k=0}^m \mathbf{J}_k^T \Omega_k \mathbf{J}_k$ being the $(n \cdot N) \times (n \cdot N)$ information matrix of the system (i.e., Hessian) and is sparse by construction, since the objective functions involved depend on small subsets of neighbours states. \mathbf{J}_k is the $(n \cdot N) \times (n \cdot N)$ Jacobian of $e_k(\mathcal{Q}_j)$ evaluated in \mathcal{Q}_j . The optimal refined trajectory is then found updating the initial guess recursively:

$$\mathcal{Q}_j^* = \mathcal{Q}_j + \Delta \mathcal{Q}_j^*. \quad (3)$$

The optimized sequence of configurations \mathcal{Q}_j^* is then provided to the robot and used as a new refined reference.

IV. RESULTS

Both the offline and online planner were tested independently. Results are collected in the following section.

A. Offline Planning

We now present some simulation and experimental results obtained using our offline planning scheme on COMAN+, a torque-controlled humanoid robot designed at Istituto Italiano di Tecnologia.

To analyze the performance of our offline planning scheme, we performed a simulation campaign on an Intel Core i7-7500U CPU running at 2.70 GHz. We considered four different multi-contact loco-manipulation tasks (Fig. 4) described in the following.

- 1) *Ladder climbing*. The robot must climb a ladder that is located in front of it.
- 2) *Parallel walls climbing*. The robot must climb vertically between two parallel walls located on its left and right flanks.
- 3) *Quadrupedal walking*. The robot must go through a narrow passage that can not be navigated by standard bipedal walking.

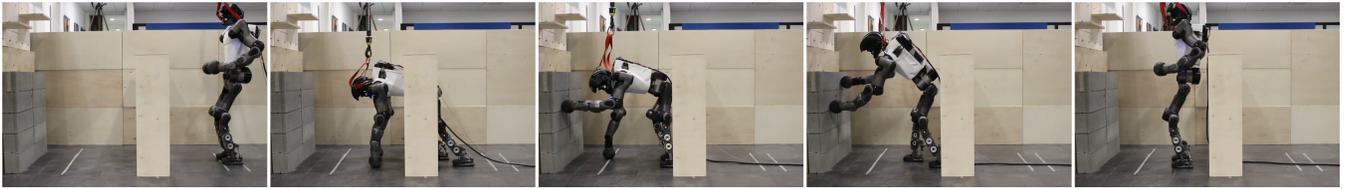


Fig. 3: COMAN+ sequentially performs the quadrupedal walking and standing up tasks.

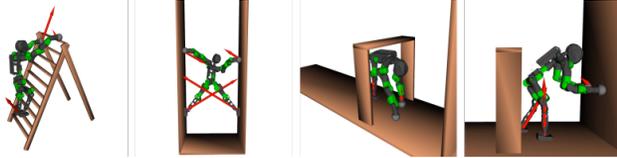


Fig. 4: The four considered tasks. For each of them, a snapshot from a typical planning solution is provided.

4) *Standing up*. The robot must stand upright, possibly exploiting a wall located in front of it as support.

Since our planning scheme is randomized (as both the stance and whole-body planners rely on probabilistic strategies), we performed 100 runs for each of the four tasks. Table I collects the most significant performance data, averaged over the 100 runs, of the proposed stance planner. For each task, we report the time needed by the stance planner to find a solution, the time spent in generating transitions, the number of performed iterations, the number of vertices in the constructed tree \mathcal{T} , and the number of stances in the solution sequence \mathcal{S}_σ . To show the feasibility of the planned motions on the real robotic platform, we performed a complete experiment in which COMAN+ was requested to sequentially perform the quadrupedal walking and standing up tasks in a purposely constructed scenario (Fig.3) ¹.

B. Online Planner

The proposed perception-based online planner was evaluated through a number of simulation studies and experimental trials carried out on the CENTAURO robot [9] reduced considering the joints of the lower body as the only active joints, resulting in a 24 DoFs robot. The robot is equipped with two Intel® RealSense™ D435i depth cameras mounted on its pelvis and head. No other external sensor has been used to perceive the environment.

The online planner is tested independently from the offline global planner and a straight wheeled motion is used as the initial global plan, moving the robot 4m forward. In this scenario, the stance does not change and the robot moves simply rolling and steering its wheels instead of taking steps. The resulting trajectory is discretized into 50 configurations and the online planner works locally with a moving horizon of 20 vertices, thus ignoring the furthest ones (Fig.5a). However, the global trajectory can be arbitrarily long, including a higher number of vertices without under performing the local planner, which depends on the number of local vertices involved in the refinement only. During the execution of the

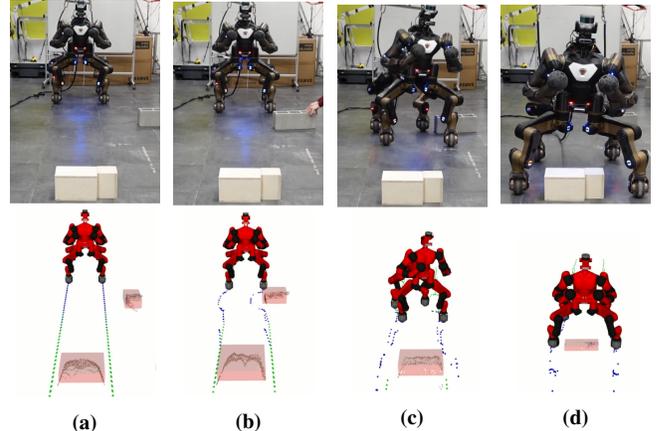


Fig. 5: Screenshots from the experiment carried out with the CENTAURO robot. Blue and green dots are the poses of the four wheels corresponding to vertices inside and outside the moving horizon of the planner.

task, obstacles are dynamically introduced to force the robot to refine the whole-body trajectory, successfully avoiding any collision (Figs. 5b-5c-5d) ².

The computation time fluctuates around 0.07s during the execution of the trajectory.

V. CONCLUSION

This extended abstract presents our recent work on planning strategies to allow autonomous motion generation of legged robots in cluttered environments. Specifically, our planners explore the feasible configuration space. The offline planner finds a feasible global plan, that can be used as an initial guess for the online planner, which refines the whole-body trajectory whenever required.

Experiments and simulations were carried out on the two planning modules separately, using two different robotic platforms, i.e., the COMAN+ humanoid and the CENTAURO robot. Preliminary results confirmed the possibility to use such a offline and online planning strategy to generate and adjust a feasible configuration space trajectory in general dynamic environments. Future work will focus on merging the two presented offline and online planners into a unified framework allowing the real-time generation of appropriate motions to accomplish loco-manipulation tasks with legged robots.

¹<https://youtu.be/zS44CegGqow>

²<https://youtu.be/qaBNzVrZtF4>

REFERENCES

- [1] P. Ferrari, L. Rossini, F. Ruscelli, A. Laurenzi, G. Oriolo, E. Mingo Hoffman, and N. Tsagarakis, "Multi-contact planning and control for humanoid robots: design and validation of a complete framework," *The International Journal of Robotics Research*, Under Review.
- [2] L. Rossini and N. Tsagarakis, "From offline to online: A perception-based local planner for dynamic obstacle avoidance applied to humanoid robots," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Under Review.
- [3] M. Görner, R. Haschke, H. Ritter, and J. Zhang, "Moveit! task constructor for task-level motion planning," 05 2019, pp. 190–196.
- [4] S. Haddadin, A. De Luca, and A. Albu-Schäffer, "Robot collisions: A survey on detection, isolation, and identification," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1292–1312, 2017.
- [5] K. Hauser, T. Bretl, J.-C. Latombe, K. Harada, and B. Wilcox, "Motion planning for legged robots on varied terrain," *The International Journal of Robotics Research*, vol. 27, no. 11-12, pp. 1325–1349, 2008.
- [6] L. Rossini, E. Mingo Hoffman, A. Laurenzi, and N. Tsagarakis, "NSPG: An efficient posture generator based on null-space alteration and kinetostatics constraints," *Frontiers in Robotics and AI*, p. 236, 2021.
- [7] F. Ruscelli, M. P. Polverini, A. Laurenzi, E. Mingo Hoffman, and N. G. Tsagarakis, "A multi-contact motion planning and control strategy for physical interaction tasks using a humanoid robot," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020, pp. 3869–3876.
- [8] L. Jaillet and J. M. Porta, "Asymptotically-optimal path planning on manifolds," in *Robotics: Science and Systems*, 2012, pp. 1–8.
- [9] N. Kashiri, L. Baccelliere, L. Muratore, A. Laurenzi, Z. Ren, E. M. Hoffman, M. Kamedula, G. F. Rigano, J. Malzahn, S. Cordasco, P. Guria, A. Margan, and N. G. Tsagarakis, "Centauro: A hybrid locomotion and high power resilient manipulation platform," *IEEE Robotics and Automation Letters (RAL)*, vol. 4, no. 2, pp. 1595–1602, 2019. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8630605>