Monte Carlo Tree Search Gait Planner for Non-Gaited Legged System Control

Lorenzo Amatucci[1], Joon-Ha Kim[2], Jemin Hwangbo[2] and Hae-Won Park[2]

***Abstract*— In this work, we show the implementation of a control framework for non-gaited legged system control. The presented approach decouples the gait optimization from the motion stabilization treating the contact phase as a decision variable. The system has been tested against the state-of-the-art Mixed Integer Quadratic Programming solver and verified in various simulation environments.**

***Paper Type* – Recent Work [1].**

I. MOTIVATION

This work assesses the open challenge of gait optimization for a legged robot system. Prior works showed the capabilities of Model Predictive Control (MPC) based controllers, like the one used in [2] and [3], to stabilize legged robots and perform dynamic and robust walking. However, these approaches do not include the contact sequence in the optimization problem to satisfy the real-time constraint. Precisely, including also the contacts implies the use of the linear complementarity constraint, which significantly affects the optimization's speed because it does not respect the linear independence constraint qualification [4]. Although there were researches like [5] and [6], which considered the contact sequence inside the optimization framework, they were restricted to only offline usage due to computation time. Besides, the proposed approach decouples the gait sequence optimization by considering the problem as a decision-making process. The redefined contact sequence problem is solved with significantly decreased computation time by utilizing a Monte Carlo Tree Search (MCTS) algorithm that exploits optimization-based simulations to evaluate the best search direction.

II. CONTROL FRAMEWORK

The overall framework is shown in Fig. 1. The user inputs are $\dot{x}$ and $\dot{y}$, which are respectively the target velocity in the x and y directions, and $\dot{\psi}$ is the target yaw rate. The reference trajectory for the system is created by integrating the user target speeds, and the footholds reference are generated by using a heuristic method as in [7]. The MCTS then uses the current contact information and robot states with the reference trajectory to generate the contact sequence used by the MPC. Now, MPC solves the optimization problem and generates the ground reaction forces (GRF) and future

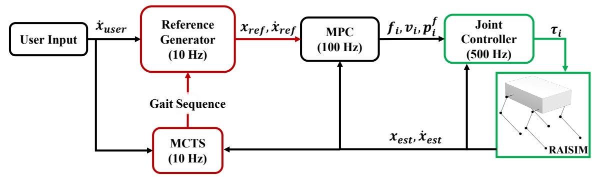

Fig. 1. Diagram of overall control framework. The $\dot{x}_{user}$ is defined as the x, y, yaw directional velocity input from the user. The x_{est} and $\dot{x}_{est}$ is the estimated state values (we used the true values from the simulation in this work).

footholds to track the desired trajectory. Finally, the GRF, feet positions, and velocities are fed into a joint space PD controller. Between these procedures, the MCTS is employed to optimize the gait sequence utilized in the overall control framework where its procedure is summarized in Fig. 2. The algorithm creates a tree search where each node represents one of the possible choices for the contact configuration. Starting from the root node representing the current contact situation, each node deeper in the tree represents the sequence of choices that constitute the gait prediction for the time horizon. At each iteration of the tree growing process, starting from the most promising node, new nodes are expanded considering all the available options (e.g., 16 combinations of possible contact configurations for quadrupeds). The proposed tree search algorithm implements two main policies: simulation and tree policies.

The simulation policy, which is used for node evaluation, solves a constrained optimization problem. The cost function is formulated as the weighted sum of the error on the state and control vector regarding the reference, while the approximated system dynamics define the constraints. The solution of the optimization, $\tilde{\boldsymbol{J}}_k$, is utilized in equation (1) to finally evaluate the contact sequence:

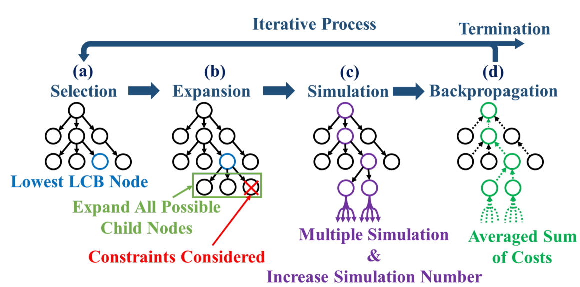

Fig. 2. Iterative growing process of the MCTS

[1] Lorenzo Amatucci was with the Humanoid Robot Research Center, Korea Advanced Institute of Science and Technology, Daejeon 34141, Korea. He is now with Dynamic Legged Systems Laboratory, Istituto Italiano di Tecnologia (IIT), 16163 Genova, Italy

[2] Joon-Ha Kim, Jemin Hwangbo and Hae-Won Park are with the Humanoid Robot Research Center, Korea Advanced Institute of Science and Technology, Daejeon 34141, Korea. haewonpark@kaist.ac.kr

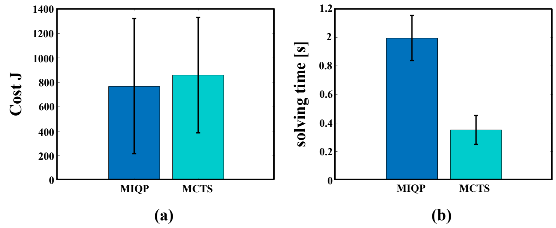

Fig. 3. The average cost of the MIQP and the MCTS on graph (a) and the relative solving time for the two algorithms on graph (b).

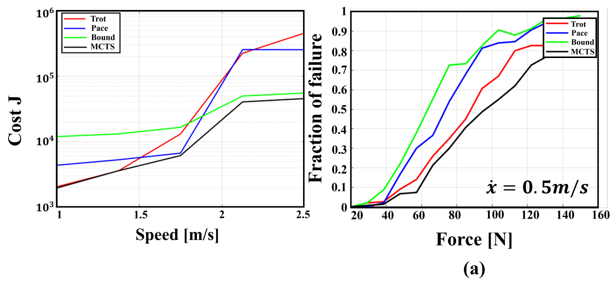

Fig. 4. (a) Cost comparison of MCTS with predefined gaits over target speed variation. The x-axis is the target speed while the y axis, in logarithmic scale, is the average running cost of the MPC evaluated at steady state. (b) Fraction of failed simulation at a defined disturbance magnitude with the system target speed at $0.5m/s$

$$\boldsymbol{J} = \sum_{k=1}^{N_{step}} \{\tilde{\boldsymbol{J}}_k + R_c \cdot (N_{leg} - \sum_{i_{leg}=1}^{N_{leg}} c_{i_{leg},k})\}, \qquad (1)$$

where $\tilde{\boldsymbol{J}}_k$ R_c is the weight relative to the contact cost, and $c_{i_{leg},k}$ is the binary variable representing the contact state for the i^{th} leg in k^{th} step. Additional term $N_{leg} - \sum_{i_{leg}=1}^{N_{leg}} c_{i_{leg},k}$ in equation (1) minimizes the leg's air-time, which increases the stability of the solution at lower target speeds.

The other policy defined in the MCTS is the tree policy. This is used to traverse the tree and so define the most favourable searching direction. In our approach, inspired by [8] and [9], we defined the low confidence boundary policy (LCB), equation (2), to direct the tree policy.

$$LCB_{i_{node}} = \bar{\boldsymbol{J}}_{i_{node}} - c \cdot \sqrt{\frac{\log N_{i_{node}}}{n_{i_{node}}}}, \qquad (2)$$

This is a simple yet effective heuristic equation that links the number of nodes that has been simulated, with the confidence we have in its evaluation. Thank to this approach the MCTS is able to balance between exploration and exploitation of the search space.

TABLE I

ROBOT AND FRAMEWORK PARAMETER

Parameter	Value	Parameter	Value
m	19 kg	*Body Width*	0.2 m
$\boldsymbol{I}$	*diag*(1e-2[9 60 67])Kgm^2	*Body Length*	0.6 m

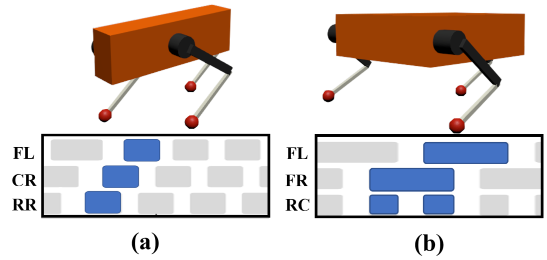

Fig. 5. Robots with different limb layout show different gait resulting from the MCTS

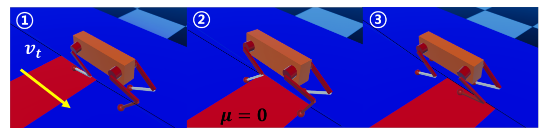

Fig. 6. Simulation of the robot on a treadmill, the floor under the left legs move at a speed v_t. The portion of the treadmill highlighted in red has a 0 friction coefficient

III. SIMULATION RESULT

The proposed algorithm is verified in various simulation environments with a quadrupedal robot simulated in the Raisim simulator [10], where the parameters of the robot are summarized in Table I. To show the benefit of the MCTS, we compared the resultant cost from the MCTS with the one calculated by a mixed-integer quadratic programming solver, GUROBI [11]. This solver has been chosen for comparison since it can handle the same optimization problem described for the simulation policy of the gait planner. As shown in figure 3 while the resulting cost is comparable, the solving time of the MCTS is significantly lower, up to 3 times. On the other hand, we also show the benefit of the proposed framework compared to the conventional fixed gait algorithms in terms of efficiency and robustness. In figure 4 we show the comparison with fixed gaits like trot, pace, and bounding. Tracking and efficiency of the system at various speeds were considered a metric for comparison. Furthermore, to tackle the robustness of the controller, we perturbed the robot with pushing forces of increasing magnitude and recorded the number of failures. In both tests, the MCTS outperformed the fixed gait on all metrics. The simulation campaign also included the test on robots with different limb layouts. In particular, we show in figure 5 two examples of different periodic gaits naturally emerging. The MCTS could exploit and adapt to the different layouts without any expert tuning. Finally, the framework's robustness has been further tested using a treadmill with different sections, as shown in figure 6. The left portion of the treadmill has a velocity $v_t = 0.5m/s$ where a portion highlighted in red has no friction, while the right side has no velocity. The robot has no information about the environment and relies only on proprioceptive data but still tracks the desired trajectory with no difficulties. Those simulations and further result are shown in the video at `https://youtu.be/4nWE4-Q4pBk`

IV. CONCLUSION

In this work, we presented a novel framework for non-gaited legged locomotion, where the contact sequence problem is defined as a decision-making process. In this way, the gait generation is decoupled from the ground reaction forces and foothold optimization. The contact sequence optimization is then tackled by utilizing an MCTS-based approach. The modified MCTS exploits the prediction capabilities of the MPC for the exploration of the possible phases combination of contacts over a fixed time horizon. The shown simulation results highlight the potential of this approach against the state-of-the-art MIQP solvers. The results show that the proposed method can discover periodic gaits and adapt to external disturbances and unknown terrain morphology and characteristics, therefore increasing the system's robustness. Finally, it is shown that the framework is easily adaptable to various robot layouts.

REFERENCES

[1] L. Amatucci, J.-H. Kim, J. Hwangbo, and H.-W. Park, "Monte carlo tree search gait planner for non-gaited legged system control," in *IEEE International Conference on Robotics and Automation*, 2022.

[2] S. Hong, J.-H. Kim, and H.-W. Park, "Real-time constrained nonlinear model predictive control on so(3) for dynamic legged locomotion," in *IEEE International Conference on Intelligent Robots and Systems*, 2020.

[3] G. Bledt and S. Kim, "Implementing regularized predictive control for simultaneous real-time footstep and ground reaction force optimization," in *Conference: International Conference on Intelligent Robots and Systems (IROS) 2019*, 11 2019.

[4] M. Neunert, M. Stäuble, M. Giftthaler, C. D. Bellicoso, J. Carius, C. Gehring, M. Hutter, and J. Buchli, "Whole-body nonlinear model predictive control through contacts for quadrupeds," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1458–1465, 2018.

[5] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1560–1567, 2018.

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

[7] C. Gehring, S. Coros, M. Hutter, M. Bloesch, M. A. Hoepflinger, and R. Siegwart, "Control of dynamic gaits for a quadrupedal robot," in *2013 IEEE International Conference on Robotics and Automation*, pp. 3287–3292, 2013.

[8] L. Kocsis and C. Szepesvári, "Bandit based monte-carlo planning," in *Machine Learning: ECML 2006* (J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, eds.), (Berlin, Heidelberg), pp. 282–293, Springer Berlin Heidelberg, 2006.

[9] L. Kocsis, C. Szepesvári, and J. Willemson, "Improved monte-carlo search," 2006.

[10] J. Hwangbo, J. Lee, and M. Hutter, "Per-contact iteration method for solving contact dynamics," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 895–902, 2018.

[11] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2021.