# Implementation of Convex Optimization Control on the MIT Humanoid

Yanran Ding[1,2], Elijah Stanger-Jones[2] and Sangbae Kim[2]

***Abstract*— This paper summarizes the practical aspect of the application of a convex optimization control on the MIT Humanoid. The hierarchical control framework consists of a model predictive control (MPC) with reduced-order model, followed by a reactive whole-body control (WBC) with full-order model. Our work emphasizes the critical components, including automatic differentiation, inverse kinematics and multi-threading, that enabled successful simulation-to-real transfer to the hardware. Preliminary experimental results are presented, including push recovery and stable walking on the treadmill with a maximum walking speed of up to 0.4 m/s.**

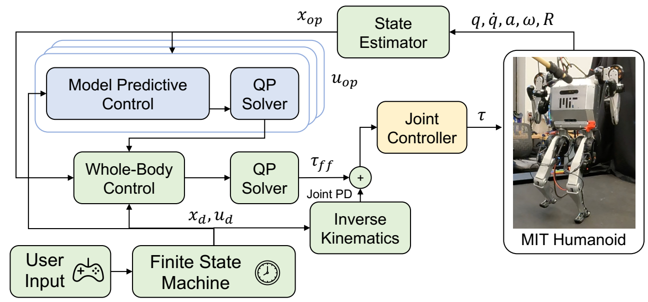


Fig. 1. The control architecture of the MIT Humanoid. A finite state machine takes in the user command and specifies the gait phases and reference states; the convex MPC tracks the reference state at 250 Hz by leveraging multi-threading; the MPC solution is taken as reference by the WBC, which computes the feedforward joint torque at 500 Hz; The inverse kinematic module calculates the joint PD target from the desired torso and swing foot poses; the state estimator produces robot state based on sensor readings from joint encoders and the IMU. The QP problems are solved using the open-source solver qpSWIFT [9]. On the right, the MIT Humanoid is recovering from external pushes while walking

## I. INTRODUCTION

Model predictive control (MPC) has proven to be one of the effective optimization-based control methods [1]. Its efficacy comes from the capability to predict state evolution over a horizon based on the system model. Reduced order models such as the Linear Inverted Pendulum [2] is widely used for humanoid control. In contrast, more complex models promise higher level of expressiveness. Whole-body MPC has shown agile and versatile behaviors on legged robots [3], [4], where the full system dynamics are leveraged. However, the high computation demand poses numerical challenges to the available solvers. To tackle this issue, model hierarchy predictive control (MHPC) [5] proposes a blend of complex and simple models over the horizon to achieve real-time control. Recent emergence of data-driven approaches [6] provides alternative solution for locomotion skill-acquisition. Although these methods have shown promising results in quadrupeds, hardware experiments on a full humanoid have yet to be demonstrated.

Hierarchical control architecture consisting of models with different complexity strikes a balance between model fidelity and computational load. In particular, the two-level control structure with MPC and whole-body control (WBC) has proven to be effective on legged robots [7], [8]. This framework is amenable for real-time execution since each sub-problem can be transcribed to a convex quadratic program (QP), which can be solved efficiently. Nevertheless, transferring such control architecture to real hardware still poses considerable challenges.

This paper addresses the implementation issues of the hierarchical convex optimization control framework on the MIT Humanoid platform. Aside from the optimal control modules, we focus on the auxiliary components that facilitate the transfer from simulation to the real hardware. Using this control framework, the MIT Humanoid is able to recover from external pushes while walking, as shown in Fig. 1.

## II. CONVEX OPTIMIZATION CONTROL

### A. Convex Model Predictive Control

The control of the MIT Humanoid [10] is posed as an optimal control problem (OCP). The robot is modeled as a single rigid body (SRB) since most of its mass is lumped at the torso.

$$\begin{aligned} \underset{\boldsymbol{x},\boldsymbol{u}}{\text{miminize}} \quad & \ell_N(\boldsymbol{x}_N) + \sum_{k=0}^{N-1} \ell_k(\boldsymbol{x}_k, \boldsymbol{u}_k) \\ \text{subject to} \quad & \boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k + \boldsymbol{d} \\ & \boldsymbol{x}_k \in \mathbb{X}, \boldsymbol{u}_k \in \mathbb{U}, \forall k = 0, ..., N-1 \\ & \boldsymbol{x}_t = \boldsymbol{x}(t), \end{aligned} \tag{1}$$

where $\boldsymbol{x}_t \in \mathbb{R}^n$ is the SRB state; $\boldsymbol{u} \in \mathbb{R}^m$ is the ground reaction force vector. $\ell_N$ and $\ell_k$ are the quadratic terminal and stage costs, respectively; $\mathbb{X}$ and $\mathbb{U}$ are the admissible state and control sets, respectively. The vector $\boldsymbol{d}$ and Jacobian matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ of the linear dynamics are constructed by linearizing around the current SRB state $\boldsymbol{x}_t$.

### B. Whole Body Control

A whole-body control (WBC) is employed as the reactive controller to exploit the full-order dynamics of the humanoid. Its objective includes centroidal state tracking, swing foot tracking and angular momentum minimization. The WBC can reason about the dynamic effects of moving limbs, which

[1] Robotics Department at the University of Michigan, MI - 48109, USA.
[2] Department of Mechanical Engineering at the Massachusetts Institute of Technology, MA - 02139, USA. email: yanrand@umich.edu, elijahsj@mit.edu, sangbae@mit.edu

is not captured by the SRB model in the MPC. The readers are referred to [8] for more detail.

## III. Implementation

### A. Automatic Differentiation

Automatic differentiation (AD) [11] is useful for calculating the Jacobian matrices of the constraints of an MPC problem (1). Compared with matrix indexing in the manual construction of Jacobian matrices, AD provides high flexibility of modifying the MPC formulation with minimum overhead. In this work, we used the package autodiff [12] with forward mode for the construction of (1).

### B. Inverse Kinematics

Inverse kinematics (IK) control serves as a complementary component to the MPC-WBC control. MPC solution is the contact wrench and WBC solution is the joint torque, both of which are control inputs to the second-order system. In swing foot tracking, the gait timing is sensitive to the swing foot height tracking error. Relying solely on the WBC requires high gain, which compromises other task objectives. By leveraging the IK, the swing tracking performance can be improved with low WBC swing-tracking gain, improving the numerical stability of the optimization. In this work, we use numerical IK based on the Levenberg–Marquardt Algorithm (LMA) [13].

### C. Multi-threading

Solving the MPC-QP requires intensive computational resources, which becomes the system performance bottleneck. We employed multi-threading to exploit the available computing resources by using a threadpool [14]. Each cycle initiates an optimization task to available instances, which do not share memory. The most recently solved optimization solution is used for control.

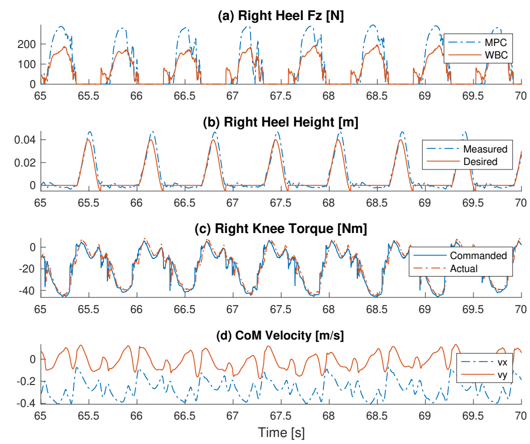


Fig. 2. MIT Humanoid Walking Experiment Data. (a) Right heel vertical force, solved from MPC (dashed) and WBC (solid); (b) Right heel swing height, measured (dashed) and desired (solid); (c) Right knee torque, commanded (solid) and actual (dashed); Center of mass velocity in the x-direction (dashed) and y-direction (solid).

## IV. Experiment Results

### A. Walking

The convex optimization control framework enabled the MIT Humanoid to walk on a treadmill with a maximum walking speed of 0.4 m/s. As presented in Fig. 2 (a), the ground reaction force generated by MPC is modified by the WBC to account for the full-order dynamics. The combination of WBC swing tracking and IK control (Section III-B) enabled small swing tracking error, as shown in Fig. 2 (b). Fig. 2 (c) shows that the maximum knee torque for flat ground walking is 45 Nm, way below the maximum knee motor torque.

## V. Conclusion and Future Work

This paper presents the practical application of convex optimization control on the MIT Humanoid. Our work focuses on the critical components such as AD, IK and multi-threading for the successful implementation of the controller on the hardware platform. Preliminary walking and push recovery results on the MIT Humanoid are presented. Future work involves extending this control framework to achieve omni-directional walking on various terrains.

## References

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

[2] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3d linear inverted pendulum mode: A simple modeling for a biped walking pattern generation," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, vol. 1. IEEE, 2001, pp. 239–246.

[3] 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.

[4] C. Mastalli, W. Merkt, G. Xin, J. Shim, M. Mistry, I. Havoutis, and S. Vijayakumar, "Agile maneuvers in legged robots: a predictive control approach," *arXiv preprint arXiv:2203.07554*, 2022.

[5] H. Li, R. J. Frei, and P. M. Wensing, "Model hierarchy predictive control of robotic systems," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3373–3380, 2021.

[6] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019.

[7] D. Kim, J. Di Carlo, B. Katz, G. Bledt, and S. Kim, "Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control," *arXiv preprint arXiv:1909.06586*, 2019.

[8] Y. Ding, C. Khazoom, M. Chignoli, and S. Kim, "Orientation-aware model predictive control with footstep adaptation for dynamic humanoid walking," in *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*. IEEE, 2022, pp. 299–305.

[9] A. G. Pandala, Y. Ding, and H.-W. Park, "qpSWIFT: A real-time sparse quadratic program solver for robotic applications," *IEEE Robot. Automat. Lett. (RA-L)*, vol. 4, no. 4, pp. 3355–3362, 2019.

[10] M. Chignoli, D. Kim, E. Stanger-Jones, and S. Kim, "The MIT humanoid robot: Design, motion planning, and control for acrobatic behaviors," in *Proc. IEEE/RAS Int. Conf. Humanoid Robot. (Humanoids)*, 2021.

[11] L. B. Rall, *Automatic differentiation: Techniques and applications*. Springer, 1981.

[12] A. M. M. Leal, "autodiff: automatic differentiation in C++ couldn't be simpler," https://github.com/autodiff/autodiff/, 2023.

[13] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 1999.

[14] B. Shoshany, "BS::thread_pool: a fast, lightweight, and easy-to-use C++17 thread pool library," https://github.com/bshoshany/thread-pool/tree/v3.5.0, 2023.