Cone-E: An Open Source Bimanual Mobile Manipulator for Generalizable Robotics

Enes Erciyes
New York University
enes.erciyes@nyu.edu

Haritheja Etukuru
New York University
hre7290@nyu.edu

Soumith Chintala
New York University

Nur Muhammad (Mahi) Shafiullah
New York University
mahi@cs.nyu.edu

Lerrel Pinto
New York University
lerrel@cs.nyu.edu

Abstract: Recent success of learning for robotics has spawned much interest and demand for capable robot platforms that may eventually approach human-level competence. However, current robot platforms for such research largely fall in one of two categories: *static bimanual* setups for manipulation, or *mobile bipedal* setups for locomotion – with a significant lack of bimanual mobile manipulators. We introduce Cone-E: an open-source, low-cost bimanual mobile manipulator designed to be a reliable general-purpose robotics research platform. Following the best practices in robot platform design for indoor environments, Cone-E integrates a compact swerve-drive base enabling omnidirectional motion, a telescopic lift mechanism affording a vertical reach range from the floor to high shelves, and dual 6-DoF arms to achieve whole-body mobility and manipulation. The design emphasizes modularity and reproducibility using off-the-shelf components, 3D printed parts, and open-source software, while remaining affordable with a bill of materials (BOM) cost of $12K.

Keywords: Whole-body Control, Bimanual Manipulation, Mobile Manipulators

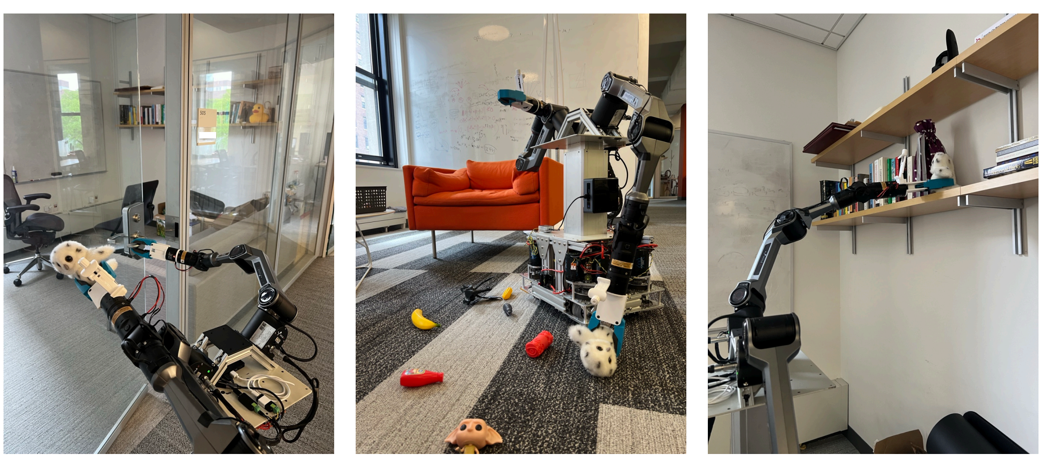

Figure 1: Cone-E is an open-source, bimanual mobile manipulator designed as a general-purpose research platform.

RSS 2025 Workshop on Whole-Body Control and Bimanual Manipulation (RSS2025WCBM).

1 Introduction

Applications of machine learning in robotics have made tremendous progress in recent years in robot navigation [1, 2, 3, 4, 5], locomotion [6, 7, 8, 9, 10], and manipulation [11, 12, 13, 14, 15, 16]. Such advances in robotics have been supported by accessible, low cost hardware such as the Unitree A1 and G1 robots, the Hello Robot: Stretch, or the Mobile Aloha open-source bimanual manipulator [17]. However, there is a noticeable gap in the currently available accessible platforms for mobile manipulation, in particular for bimanual mobile robots. Currently, such available platforms on the market tend to be inaccessible, hard to build upon, or have limited functionality due to, respectively, high-cost, closed source design, and hardware limitations.

In this work, we propose a new mobile manipulator design to accelerate generalizable robotics research – aiming to provide a reliable platform that will fuel indoor mobile manipulation research. Our most important considerations for this platform are to make it low-cost, and easy to build and repair with off-the-shelf parts, and easy to control in various ways. We identify some key needs for a good mobile manipulation research platform: dexterous bimanual arms, an omnidirectional base, and a vertically extended workspace that reaches both the floor and overhead. Beyond hardware capabilities, we identify quality-of-life developments for researchers, like having a long battery life, a small footprint, and a readable, fully open source software stack.

Our proposed mobile manipulator, Cone-E, is low-cost (with a bill of material cost $12-13K USD) and fully integrated to provide whole-body manipulation. With the publication of this work, we will open source the hardware design, including a BOM and an assembly guide, the full controller software stack, and a suite of our general-purpose “utility” policies. We believe our design will propel further research into whole-body and mobile manipulation by providing access to a stable platform with minimal dynamic constraints.

2 Hardware Design

In this section, we discuss how Cone-E achieves the hardware design goals identified in Section 1.

2.1 Mobile Base

Cone-E has an omnidirectional base to allow flexible navigation, intuitive teleoperation, and simplified policy learning. Many current commercial robots, such as the Hello Robot: Stretch [18] or the Rainbow RB-Y1 [19], use a differential-drive base due to its simplicity and lower cost. While cheap, this type of drive is non-holonomic, meaning the state of the system is dependent on the path taken in order to achieve it.

Differential-drive constraint limits arbitrary position control which is important for closed-loop learned policies. Therefore, we designed our base as a swerve drivetrain with four wheels. We use readily available components from the FIRST Robotics Competition (FRC) ecosystem [20], similar to Tidybot++ [21]. A frame made of aluminum extrusions carry the four swerve modules, a power distribution block and the battery.

Unlike TidyBot++, we do not modify the swerve modules to create caster wheels. Our base is *non-holonomic* if modeled at the level of infinitesimally small timesteps. However, an abstraction of the system with discrete timesteps, longer than the steering duration, still renders a holonomic system. We find the maneuverability of swerve modules is enough for non-dynamic tasks in household environment while bypassing additional build complexity from machining caster modules.

We further refine our design to give the base a small footprint (34 x 42 cm) allowing navigation in household environments similar to humans. We designed this base to be more compact than TidyBot++ by simplifying the electrical circuitry and running all digital components from a single 24V 20Ah NMC battery. NMC batteries have higher power density compared to SLA and LiFePO4 batteries, providing long runtime in a compact form factor. Our base motors, the lift motor and the arms all run on 24V and can be directly powered from the power distribution block. The control module, an Intel NUC mini PC, runs on 19V and needs a step-down voltage regulator after the power distribution block. Practical splice connectors like Wago [22] and power distribution panel with many output channels keep the circuitry easy to build and customizable. In addition to the circuitry, we use SDS MK4c swerve modules instead of SDS MK4 due to their smaller footprint and lower chassis mounting.

2.2 Telescoping Lift

A core component to increase the vertical reach of mobile manipulators is a lift mechanism that provides a vertical degree of freedom. Lift mechanisms allow the robot to raise or lower its torso consisting of arm and sensors and expands its workspace beyond a fixed mounting height. This added vertical mobility is essential for household tasks like reaching high shelves, picking objects from the floor, or achieving optimal sensor viewpoints.

Most commercial lift designs are custom built per order and expensive, not readily available on the market. To keep our design low-cost and easy to build with the off-the-shelf parts, we re-purpose an adjustable height telescoping table (shown in Figure 2) as our robot lift. The table is a telescoping lead screw mechanism driven manually with a hand crank, which we motorize for our purposes. Inside, there are three lead screws that nest inside each other. The screws rotate in sequence. This allows for compact collapsed length and extended height adjustment. The screws are self-locking, thus, they hold position when unpowered. This makes Cone-E more efficient as the lift does not need to draw power when stationary. The outside of the lift are three telescoping aluminum columns that are held together by rubber friction pads. We use the thin steel panels on the top and bottom of the lift to mount the torso and attach the lift onto the base respectively.

We automate this table by motorizing the hand crank drive shaft. We create a timing belt pulley that fits on the shaft using nylon or metal 3-D printing. Then, using a timing belt and a BLDC servo motor, we can control the lift height. The lift is 30.5 cm at its lowest and 72 cm at its highest. We find the 41.5 cm stroke length to be enough for being able to reach the ground and also doing tabletop manipulation on high surfaces. In addition to providing extended reach, the lift acts as an extra degree of freedom that we can utilize in our inverse kinematics solver.

We use the integrated encoders inside the motor as a feedback to compute lift position. The lead screws inside the lift have a $6mm$ thread pitch. We use a 60-teeth and 18-teeth pulley on the lift and motor shaft respectively. To move the lift through its full-range, the motor needs to rotate approximately 225 rotations. To calibrate, the lift needs to "home" when the robot is turned on.

2.3 Arms and the Gripper

We build Cone-E as a bimanual robot that supports two 6DOF arms and custom grippers as manipulation tools. We choose AgileX Piper arms due to their low-cost (only $2,500) and light weight (4.2 kg). The arms are mounted onto an extruded aluminium torso with 45-degree shoulders. We choose this shape to balance the arms' forward and downward reach. The angled shoulders also prevent the elbows of the arms from colliding with each other even if their mounting points are close.

As the end-effector on the arm, we choose the NYU gripper introduced in [23]. The angular jaw design allows both precise manipulation and large force application. As an end-effector camera, we use an iPhone following the same work, and for data collection use the hand-held version of the NYU gripper with the mounted iPhone and the associated app, AnySense. The app records video, high-quality $SE(3)$ pose, and any supplementary information streamed over bluetooth, all at 30fps. We designed this tool to be more ergonomic and compact compared to other similar tool designs such as [23, 24, 25].

2.3.1 Compliant Controller

A compliant controller is essential to absorb unexpected forces encountered during manipulation and ensure safety in learned policy deployments. Therefore, we implement a joint stiffness controller

Figure 2: Cone-E is modular and easily customizable with different arms, end-effectors and sensors.

with two layers. The low-level real time controller runs at 200 Hz, while the policy sets targets for this controller at much lower frequencies. The typical joint stiffness controller objective is

$$-\tau_g(q) + K_p(q_{ref} - q) + K_d(\dot{q}_{ref} - \dot{q})$$

where q is the measured joint positions and q_{ref} is the target position set by the upstream controller. The system acts like a spring-damper around the reference position with stiffness coefficient K_p and damping coefficient K_d. The feedforward torque gravity compensation allows us to set stiffness gains lower, resulting in compliant movement.

3 Applications of Cone-E

3.1 Teleoperation

We teleoperate Cone-E using a Quest VR device following [26]. The VR controllers are re-mapped to robot control in the following way. We re-target the left and right controller poses to the corresponding arm's end-effector pose. The left and right joysticks on the controllers are used to command rotational and translational velocities to the base in the planar $SE(2)$ workspace respectively. The trigger buttons on the joysticks are used to control the lift height. Quest controller commands are published to the robot mini PC over WiFi. We find 30 Hz to be the ideal VR command frequency to balance robot responsiveness against network delays.

3.2 Policy Learning

Following Etukuru et al. [23], we used our hand-held data collection tool with an iPhone Pro to collect demonstrations for a general pick-up task. Our portable hand-held tool enables us to collect demonstrations in diverse environments. We collected approximately 5,000 demonstrations to train a general pick-up policy. We use a VQ-BeT [27] model with 30M parameters, which runs entirely on the CPU of Cone-E's mini PC. The pick-up model predicts the $SE(3)$ relative action in the current end-effector frame and the absolute gripper pose. This end-effector pose is then fed to our arm differential inverse kinematics controller, which calculates the next joint positions for the robot.

The policy takes in camera observations and predicts new actions at 2Hz, predicting the desired end-effector pose. In contrast, our low-level joint stiffness controller runs at 200Hz. To bridge this frequency gap and ensure smooth motion, we interpolate the joint commands to reach the target pose within 1 second. The policy issues a new command when the preceding one is halfway completed (every 0.5s), thereby enabling continuous and smooth robot control.

4 Conclusion

In this work, we introduce Cone-E, an open-source bimanual mobile manipulator robot platform. While we believe it offers a great balance between cost and functionality, there are certain affordances, such as a head camera and twisting neck and torso, that are not present in the current version. By open sourcing our design, we hope that the community can customize the platform to their needs while iterating on future such platforms in an open and collaborative way.

Acknowledgments

If a paper is accepted, the final camera-ready version will (and probably should) include acknowledgments. All acknowledgments go at the end of the paper, including thanks to reviewers who gave useful comments, to colleagues who contributed to the ideas, and to funding agencies and corporate sponsors that provided financial support.

References

[1] A. Sridhar, D. Shah, C. Glossop, and S. Levine. Nomad: Goal masked diffusion policies for navigation and exploration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 63–70. IEEE, 2024.

[2] P. Liu, Z. Guo, M. Warke, S. Chintala, C. Paxton, N. M. M. Shafiullah, and L. Pinto. Dynamem: Online dynamic spatio-semantic memory for open world mobile manipulation, 2024. URL https://arxiv.org/abs/2411.04999.

[3] J. Yang, C. Glossop, A. Bhorkar, D. Shah, Q. Vuong, C. Finn, D. Sadigh, and S. Levine. Pushing the limits of cross-embodiment learning for manipulation and navigation, 2024. URL https://arxiv.org/abs/2402.19432.

[4] T. Gervet, S. Chintala, D. Batra, J. Malik, and D. S. Chaplot. Navigating to objects in the real world. *Science Robotics*, 8(79):eadf6991, 2023.

[5] N. M. M. Shafiullah, C. Paxton, L. Pinto, S. Chintala, and A. Szlam. Clip-fields: Weakly supervised semantic fields for robotic memory. *arXiv preprint arXiv:2210.05663*, 2022.

[6] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In A. Faust, D. Hsu, and G. Neumann, editors, *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 91–100. PMLR, 08–11 Nov 2022. URL https://proceedings.mlr.press/v164/rudin22a.html.

[7] A. Agarwal, A. Kumar, J. Malik, and D. Pathak. Legged locomotion in challenging terrains using egocentric vision. In K. Liu, D. Kulic, and J. Ichnowski, editors, *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 403–415. PMLR, 14–18 Dec 2023. URL https://proceedings.mlr.press/v205/agarwal23a.html.

[8] X. Cheng, K. Shi, A. Agarwal, and D. Pathak. Extreme parkour with legged robots. *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11443–11450, 2023. URL https://api.semanticscholar.org/CorpusID:262826068.

[9] Z. Fu, X. Cheng, and D. Pathak. Deep whole-body control: Learning a unified policy for manipulation and locomotion. *ArXiv*, abs/2210.10044, 2022. URL https://api.semanticscholar.org/CorpusID:252968218.

[10] G. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal. Rapid locomotion via reinforcement learning. *The International Journal of Robotics Research*, 43:572 – 587, 2022. URL https://api.semanticscholar.org/CorpusID:248524958.

[11] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, L. Lee, T.-W. E. Lee, S. Levine, Y. Lu, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. Ryoo, G. Salazar, P. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut, H. Tran, V. Vanhoucke, Q. Vuong, A. Wahid, S. Welker, P. Wohlhart, J. Wu, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023. URL https://arxiv.org/abs/2307.15818.

[12] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[13] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[14] S. Haldar, Z. Peng, and L. Pinto. Baku: An efficient transformer for multi-task policy learning, 2024. URL https://arxiv.org/abs/2406.07539.

[15] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware, 2023. URL https://arxiv.org/abs/2304.13705.

[16] T. Lin, Y. Zhang, Q. Li, H. Qi, B. Yi, S. Levine, and J. Malik. Learning visuotactile skills with two multifingered hands. *arXiv preprint arXiv:2404.16823*, 2024.

[17] Z. Fu, T. Z. Zhao, and C. Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. In *Conference on Robot Learning (CoRL)*, 2024.

[18] C. C. Kemp, A. Edsinger, H. M. Clever, and B. Matulevich. The design of stretch: A compact, lightweight mobile manipulator for indoor human environments. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 3150–3157. IEEE, 2022.

[19] Rb-y1 mobile manipulator, 2025. URL https://rainbowrobotics.github.io/rby1-dev/.

[20] FIRST robotics competition, 2024. URL https://www.firstinspires.org/robotics/frc.

[21] J. Wu, W. Chong, R. Holmberg, A. Prasad, Y. Gao, O. Khatib, S. Song, S. Rusinkiewicz, and J. Bohg. Tidybot++: An open-source holonomic mobile manipulator for robot learning. In *Conference on Robot Learning*, 2024.

[22] Wago — electrical interconnection and automation technology, 2025. URL https://www.wago.com.

[23] H. Etukuru, N. Naka, Z. Hu, S. Lee, J. Mehu, A. Edsinger, C. Paxton, S. Chintala, L. Pinto, and N. M. M. Shafiullah. Robot utility models: General policies for zero-shot deployment in new environments. *arXiv preprint arXiv:2409.05865*, 2024.

[24] N. M. M. Shafiullah, A. Rai, H. Etukuru, Y. Liu, I. Misra, S. Chintala, and L. Pinto. On bringing robots home. *arXiv preprint arXiv:2311.16098*, 2023.

[25] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots, 2024. URL https://arxiv.org/abs/2402.10329.

[26] A. Iyer, Z. Peng, Y. Dai, I. Guzey, S. Haldar, S. Chintala, and L. Pinto. Open teach: A versatile teleoperation system for robotic manipulation. *arXiv preprint arXiv:2403.07870*, 2024.

[27] S. Lee, Y. Wang, H. Etukuru, H. J. Kim, N. M. M. Shafiullah, and L. Pinto. Behavior generation with latent actions. *arXiv preprint arXiv:2403.03181*, 2024.