ARMoR

Arm-Based Robot Mobility on a Passive Platform via Reinforcement Learning of Contact-Driven Locomotion

Daniel Kadosh, Itay Kadosh, Keval Shah, Robert Teal

Instructor: Professor Yu Xiang
Course: Robotics - CS 6341

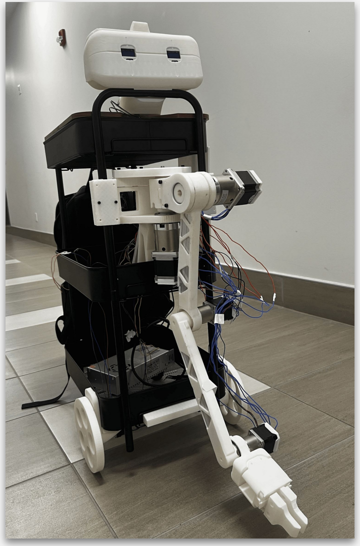
Mobile Manipulators are **costly**

While they allow you to take robotics research from the table top to the extended environment, that upgrade comes at a price.

$100,000

$25,000

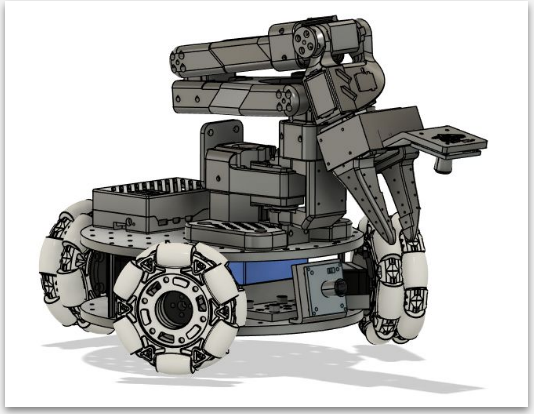
$1,000



LeKiwi is still **expensive**

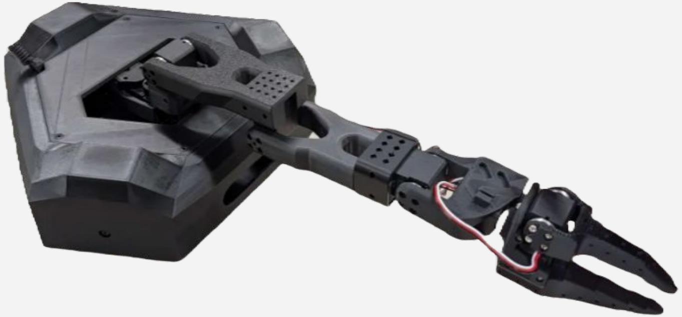
LeRobot's companion to the SO-101, LeKiwi, is a motorized mobile manipulation. It can be purchased for $500, still too expensive!

We looked to create a method that is cheaper by **hundreds of dollars!**



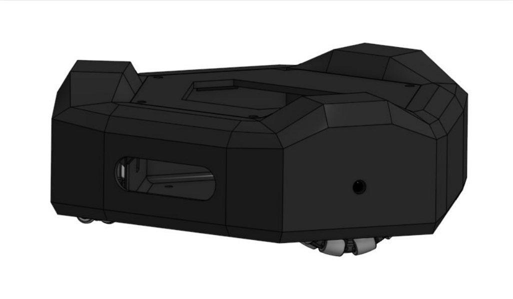
We propose, ARMoR

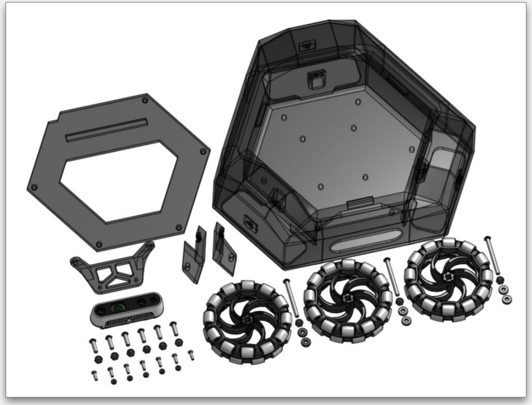
A cheaper, open-source alternative to mobile manipulators geared to smaller labs and independent researchers

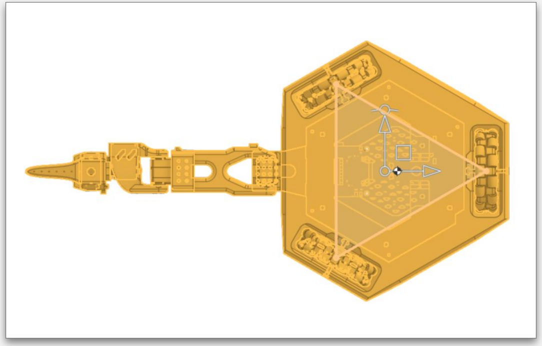


Design Considerations

- Fully 3D printable
- Lightweight
- Balanced
- Holonomic



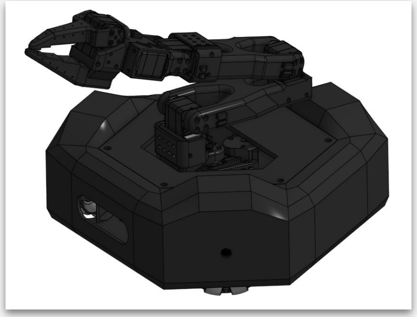


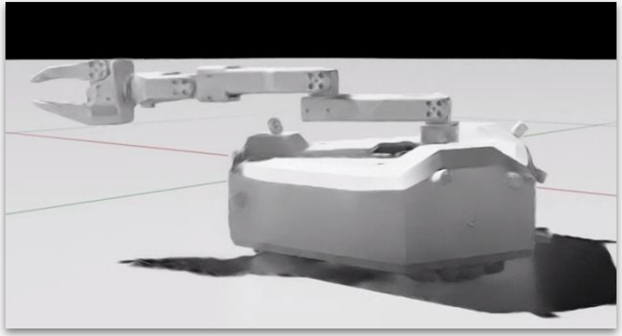


Simulation

- CAD → URDF
- We used IsaacSim and IsaacLab as our physics simulators for the RL policy.

- This part took special consideration because it is crucial that the simulation is as close as possible to reality.

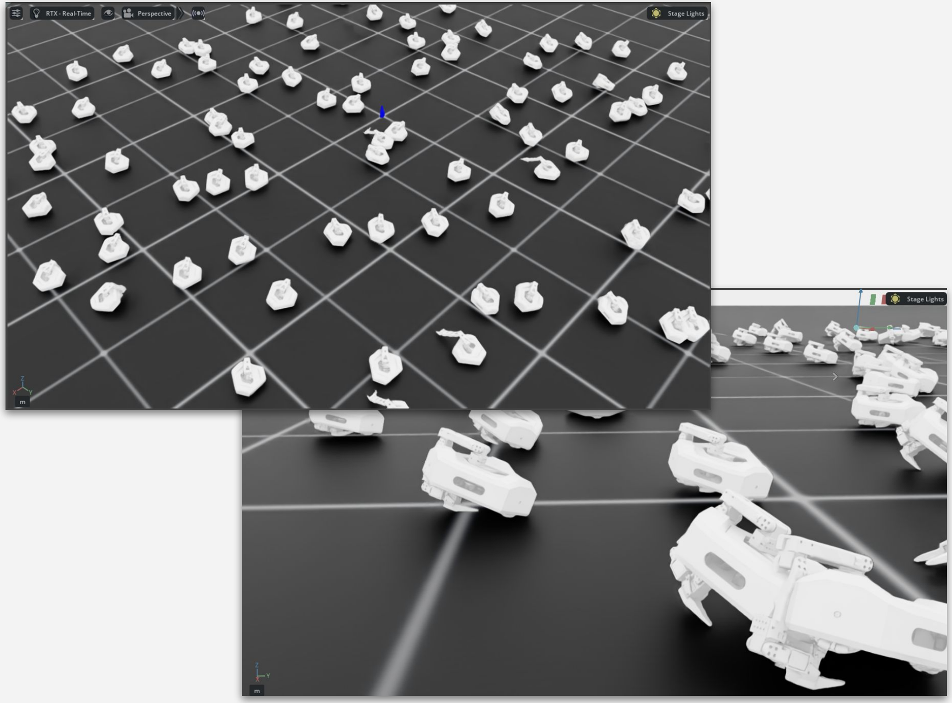




Reward and goal

- We want to learn a policy that is able to move the robot in a specified direction.

- In Isaac Lab, we simulate an environment with dozens of robots.



Reward and goal

- We carefully design a reward function to evaluate the effectiveness of our policy.

- We use this to choose which version of the robots to reinforce and which to discourage.

$$R = \underbrace{\alpha_{\text{alive}}(1-z)}_{\textbf{Alive}} + \underbrace{\alpha_f [v_x]_+ + \beta_f [v_x]_+^2}_{\textbf{Forward}} - \underbrace{\alpha_a \|\Delta a\|_2^2}_{\textbf{Smoothness}}$$
$$+ \underbrace{\alpha_c\, d_{\text{clamp}}(1+\gamma[v_x]_+)}_{\textbf{Contact+push}} - \underbrace{\alpha_w(\|\omega\|_2 + \lambda\, \text{lift}_{\text{mean}})}_{\textbf{Wheel ground}}.$$

Notation

z	Termination flag (1 if terminated, else 0).
v_x	Forward base velocity.
Δa	Action delta (per-step control change).
d_{clamp}	Clamped gripper–ground contact depth.
ω	Wheel joint velocities.
$\textbf{lift}_{\text{mean}}$	Mean positive lift of wheel heights above the ground.
x_+	$\max(x, 0)$ (ReLU).

Intent

Alive	Keep episode feasible & safe.
Forward	Encourage progress (linear + quadratic shaping).
Smoothness	Discourage jittery control via $\|\Delta a\|_2^2$.
Contact+push	Reward controlled contact, amplified when forward.
Wheel ground	Penalize wheel spin and lifted wheels.

Real Life + SimToReal

- We 3D printed all the parts and assembled the base.

- We wrote a script that creates a feedback loop between observations (joint positions, velocities) and actions (joint torques)

DEMO!

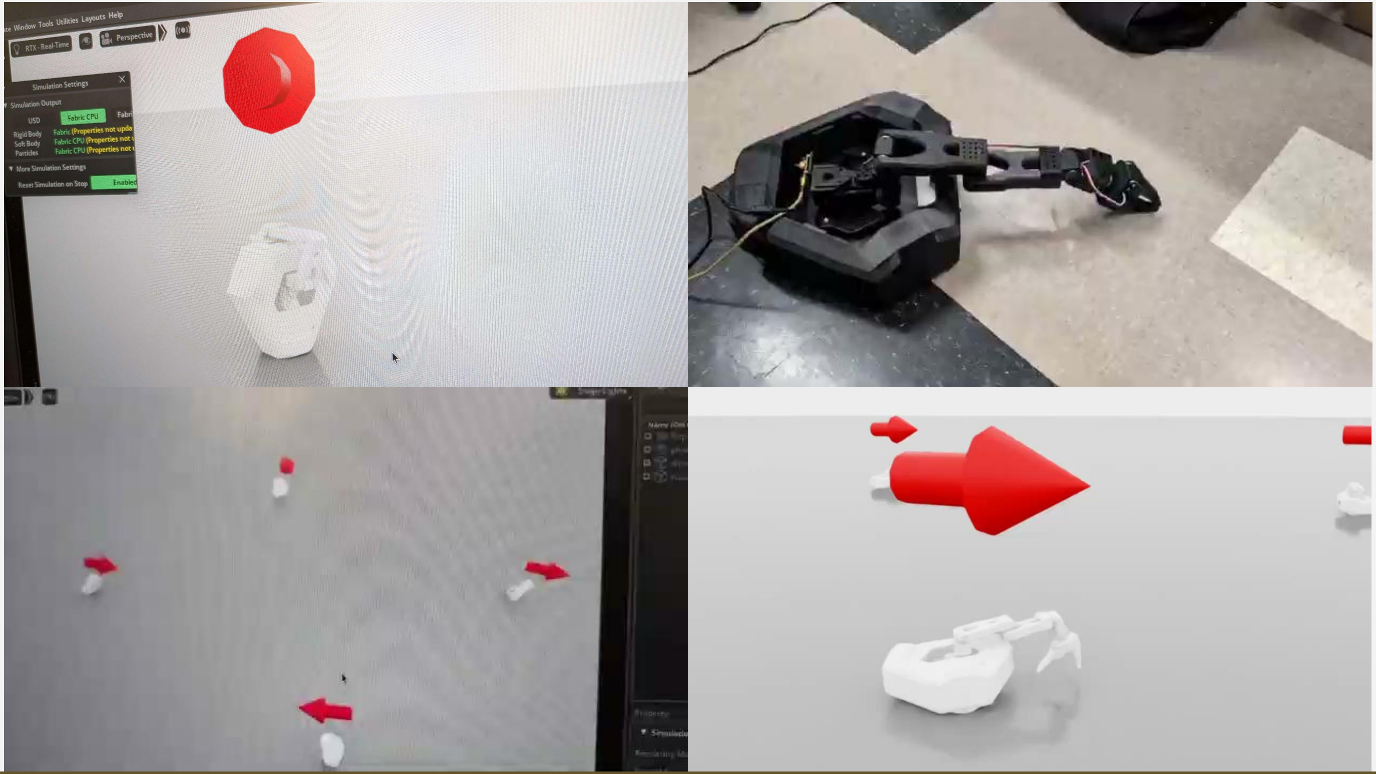
Future Plans

To turn this into a fully fledged mobile manipulator, we plan to:

1. Make Base untethered by adding microcontroller (RaspberryPi) and Battery (LiPo 12v)
2. Implement the RGB-D camera on the base.
3. Work on a more robust policy.

Although this will make the design more expensive, it will allow for robust downstream tasks like SLAM and longer horizon mobile manipulation.

Bloopers!


Window Tools Utilities Layouts Help
RTX - Real-Time
Perspective
Simulation Settings
Simulation Output
USD
Fabric CPU
Rigid Body
Soft Body
Particles
More Simulation Settings
Reset Simulation on Stop
Enabled