Open-Source Robotics in Practice: Lessons from Upkie Wheeled Bipeds

Stéphane Caron, Etienne Arlaud, Valentin Tordjman--Levavasseur

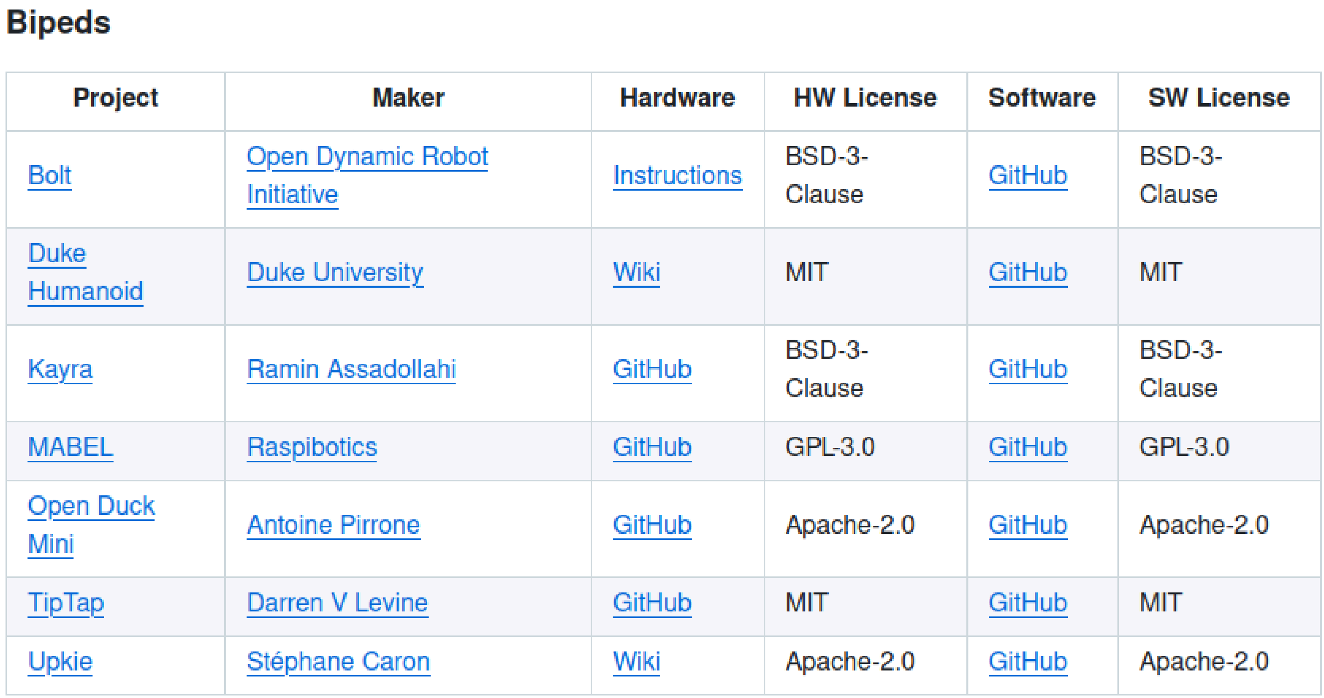
FOSDEM 2026

31 January 2026

Open-source robots

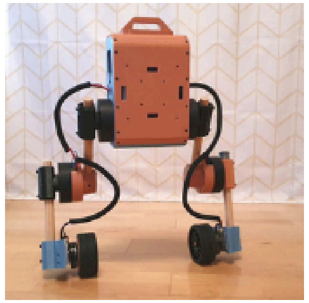
Awesome Open-Source Robots

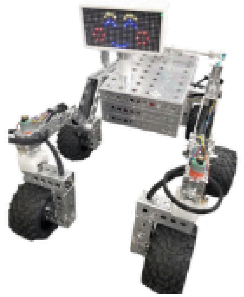
Curated collaborative list, for instance:

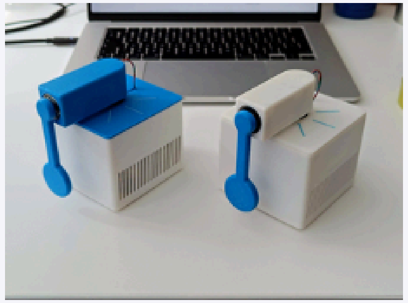


Bipeds

Project	Maker	Hardware	HW License	Software	SW License
Bolt	Open Dynamic Robot Initiative	Instructions	BSD-3-Clause	GitHub	BSD-3-Clause
Duke Humanoid	Duke University	Wiki	MIT	GitHub	MIT
Kayra	Ramin Assadollahi	GitHub	BSD-3-Clause	GitHub	BSD-3-Clause
MABEL	Raspibotics	GitHub	GPL-3.0	GitHub	GPL-3.0
Open Duck Mini	Antoine Pirrone	GitHub	Apache-2.0	GitHub	Apache-2.0
TipTap	Darren V Levine	GitHub	MIT	GitHub	MIT
Upkie	Stéphane Caron	Wiki	Apache-2.0	GitHub	Apache-2.0

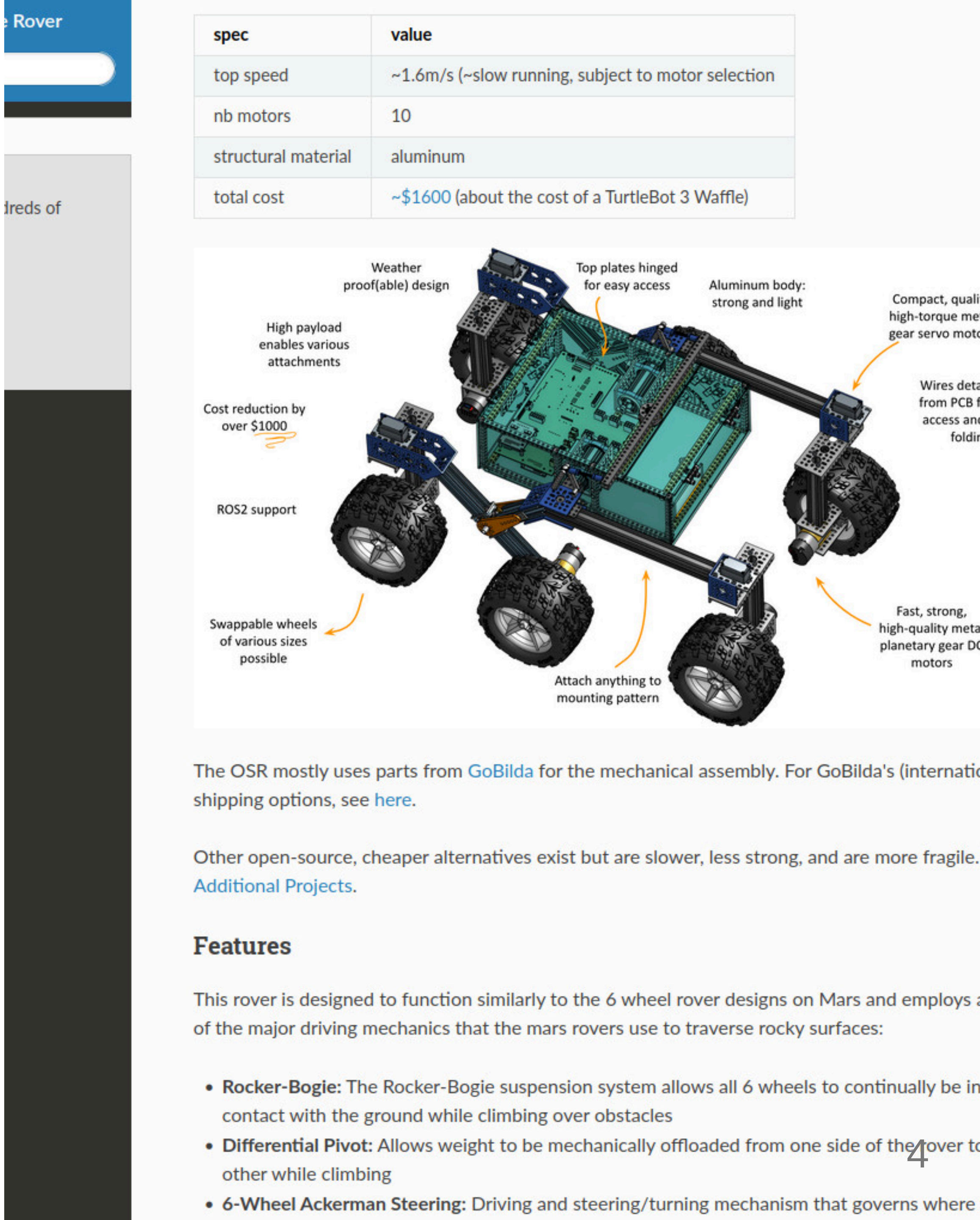






JPL Open Source Rover

- Maker: Jet Propulsion Laboratory
- Released: 2018
- License: Apache-2.0
- **Documentation: yes**
- Reproduced: ?
- Active: yes



e Rover

dreds of

spec	value
top speed	~1.6m/s (~slow running, subject to motor selection
nb motors	10
structural material	aluminum
total cost	~$1600 (about the cost of a TurtleBot 3 Waffle)

Weather proof(able) design
Top plates hinged for easy access
Aluminum body: strong and light
Compact, quali
high-torque me
gear servo moto
High payload enables various attachments
Wires deta
from PCB f
access an
foldi
Cost reduction by over $1000
ROS2 support
Swappable wheels of various sizes possible
Attach anything to mounting pattern
Fast, strong, high-quality meta
planetary gear DC
motors

The OSR mostly uses parts from GoBilda for the mechanical assembly. For GoBilda's (internatio
shipping options, see here.

Other open-source, cheaper alternatives exist but are slower, less strong, and are more fragile.
Additional Projects.

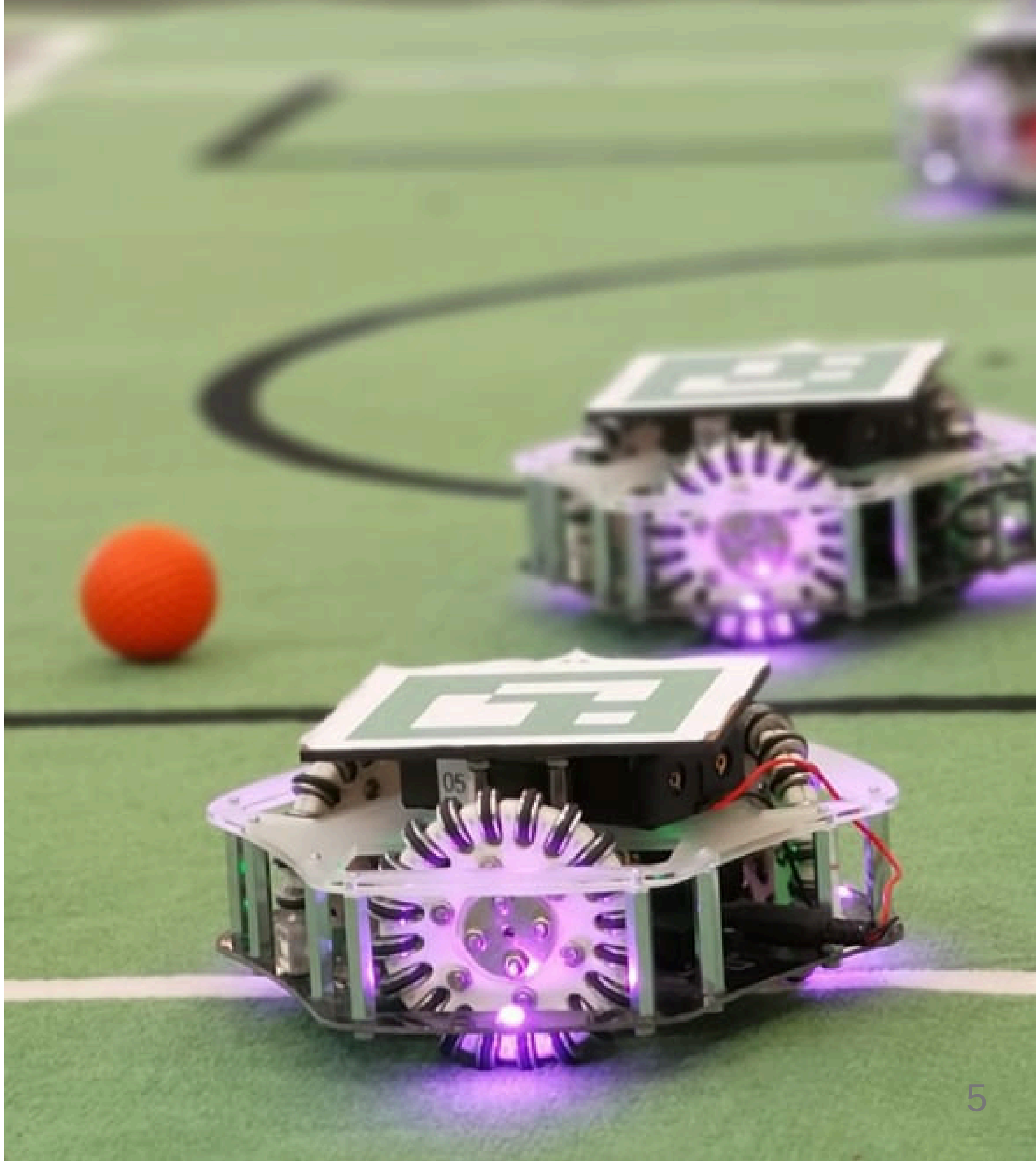
Features

This rover is designed to function similarly to the 6 wheel rover designs on Mars and employs a
of the major driving mechanics that the mars rovers use to traverse rocky surfaces:

- **Rocker-Bogie:** The Rocker-Bogie suspension system allows all 6 wheels to continually be in
contact with the ground while climbing over obstacles
- **Differential Pivot:** Allows weight to be mechanically offloaded from one side of the rover to
other while climbing
- **6-Wheel Ackerman Steering:** Driving and steering/turning mechanism that governs where

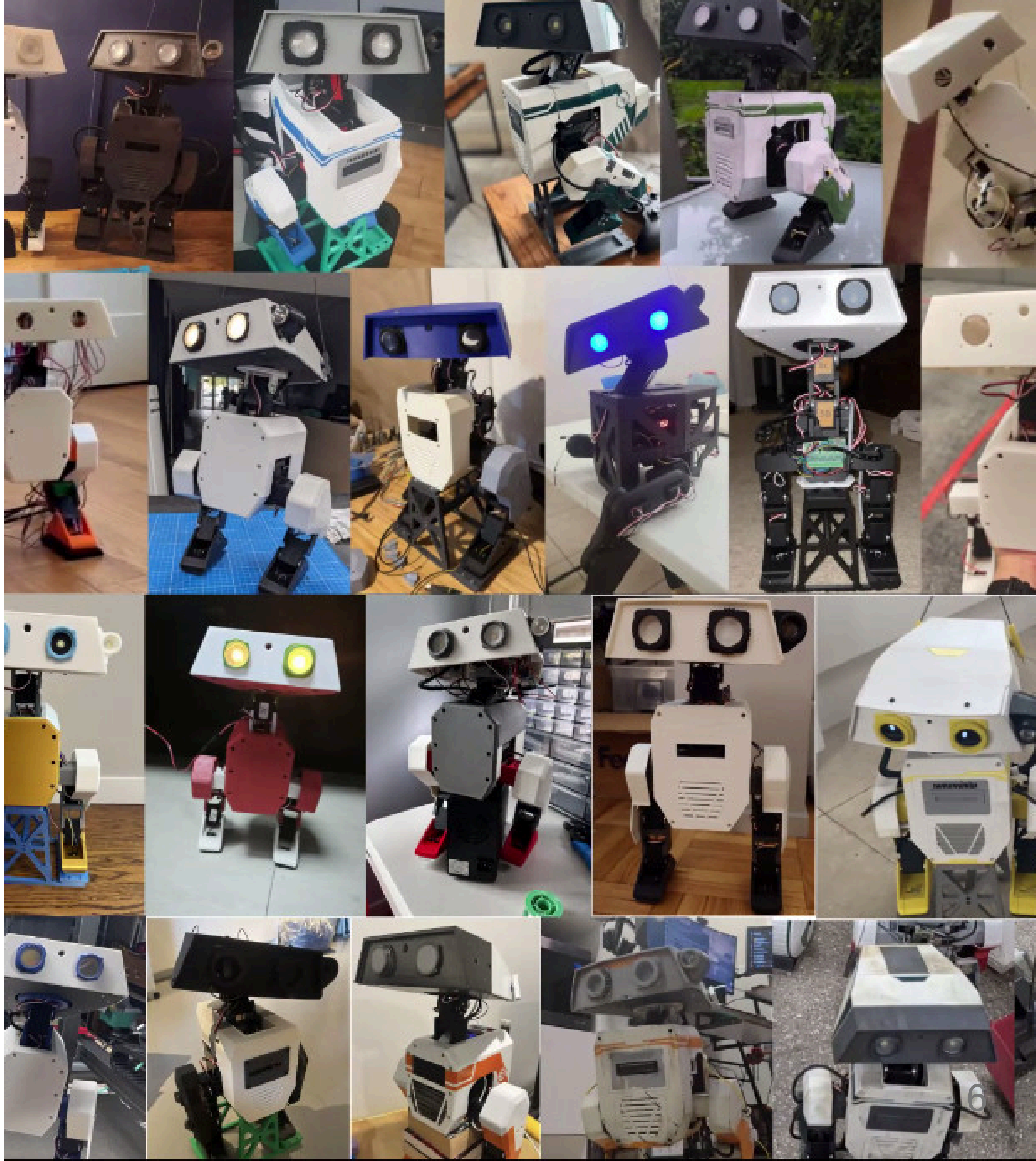
Robot Soccer Kit

- Maker: Robot Soccer Kit
- Released: 2021
- License: CC-BY-NC ❌
- Documentation: yes
- Reproduced: yes
- **Active: yes**



Open Duck Mini

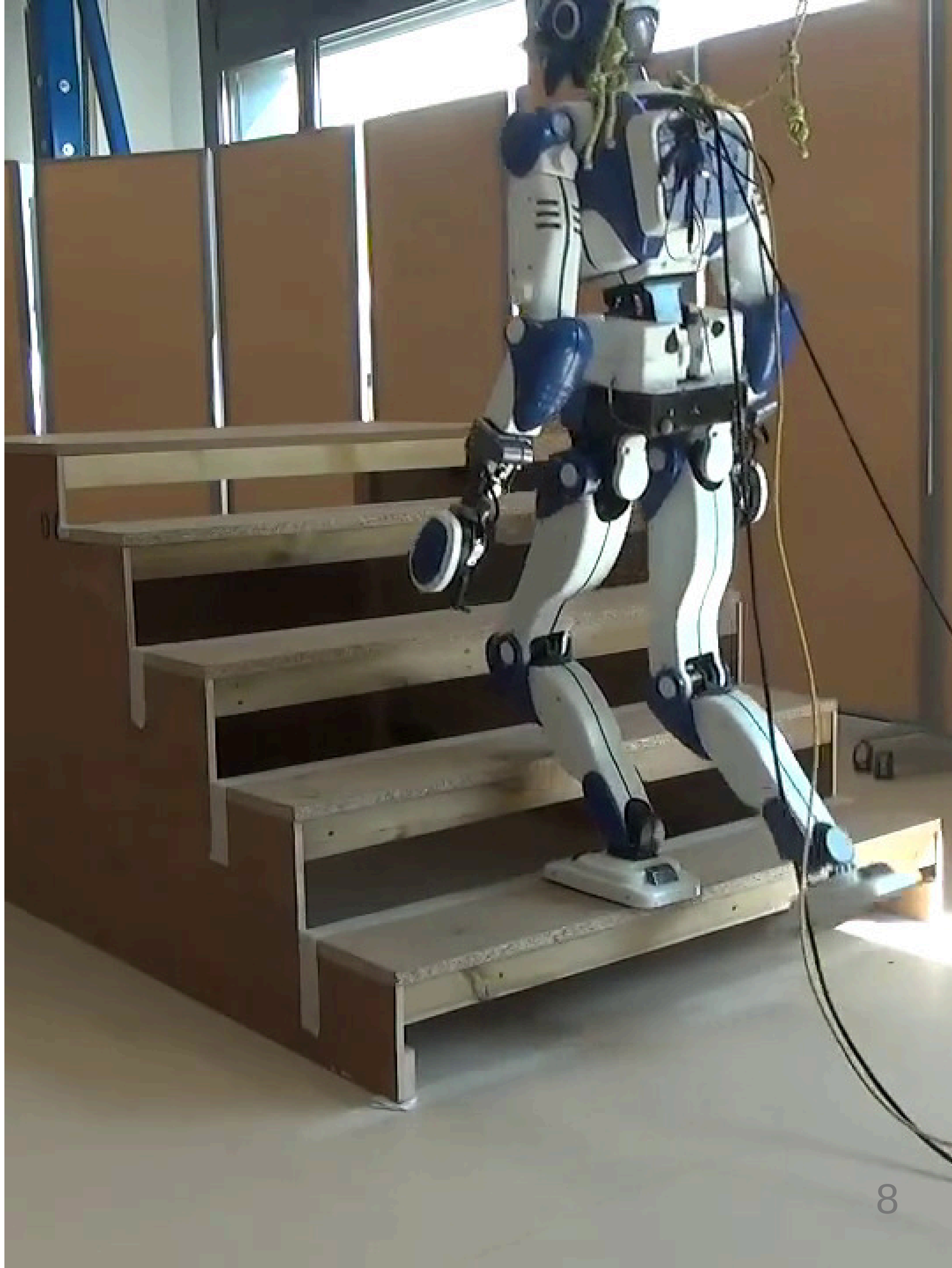
- Maker: Antoine Pirrone
- Released: 2024
- License: Apache-2.0
- Documentation: ?
- **Reproduced: yes**
- Active: yes



How we got there?

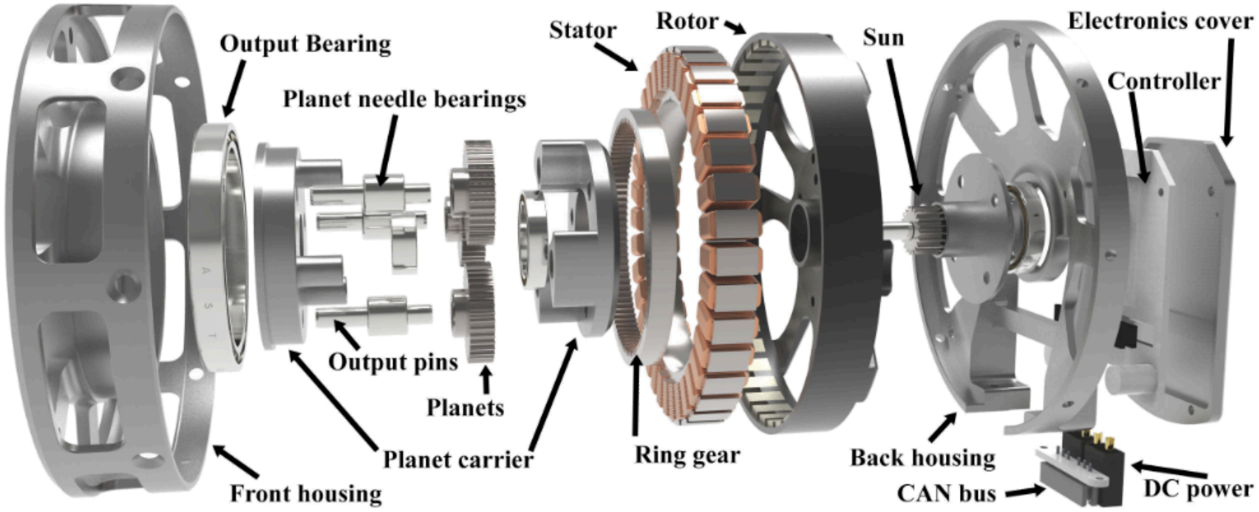
Ten years ago

- Expensive robots, labs only
- Not breaking them was high priority
- Software: open-source frameworks
- Hardware: **not** open-source



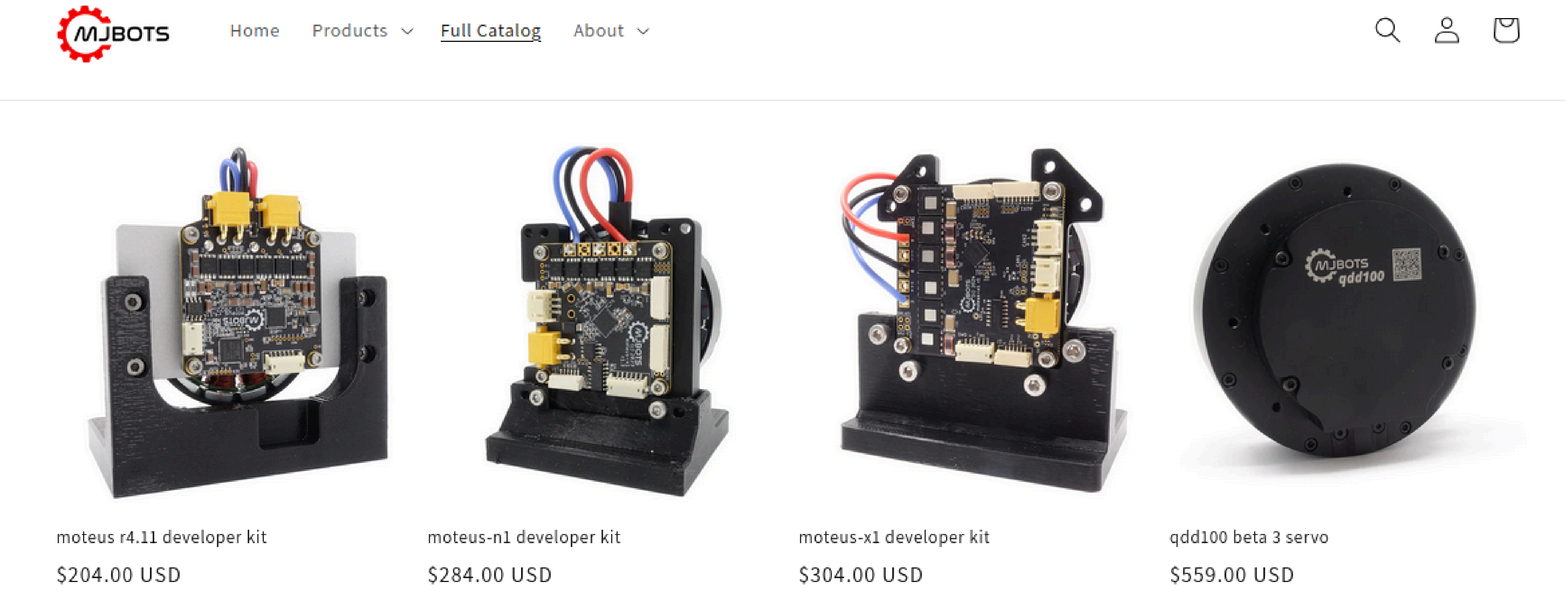
And then: QDD actuators

Ben Katz's MSc thesis: A Low Cost Modular Actuator for Dynamic Robots (2018)



See also: Ben Katz's blog, the QDD actuators of the Berkeley humanoids.

Open-source actuators



moteus r4.11 developer kit

$204.00 USD

moteus-n1 developer kit

$284.00 USD

moteus-x1 developer kit

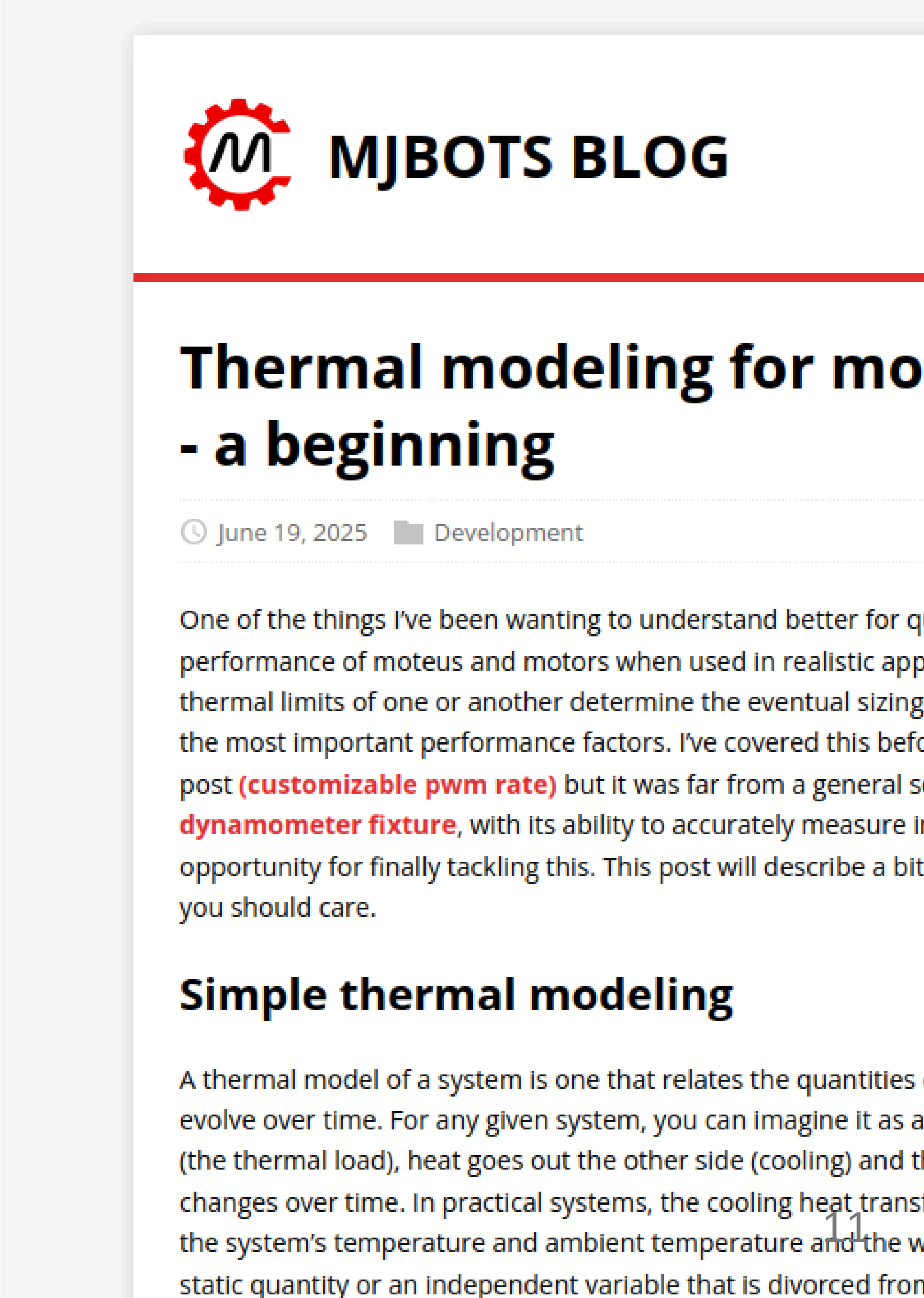
$304.00 USD

qdd100 beta 3 servo

$559.00 USD

mjbots

- **Open-source** QDD actuators, brushless motor controllers, Raspberry Pi hat, ...
- Hardware: mech. and electrical designs
- Firmware: of the brushless controllers
- Software: C++/Python libraries
- Discord channel



MJBOTS BLOG

Thermal modeling for mo - a beginning

June 19, 2025 Development

One of the things I've been wanting to understand better for q performance of moteus and motors when used in realistic app thermal limits of one or another determine the eventual sizing the most important performance factors. I've covered this befo post **(customizable pwm rate)** but it was far from a general s **dynamometer fixture**, with its ability to accurately measure i opportunity for finally tackling this. This post will describe a bit you should care.

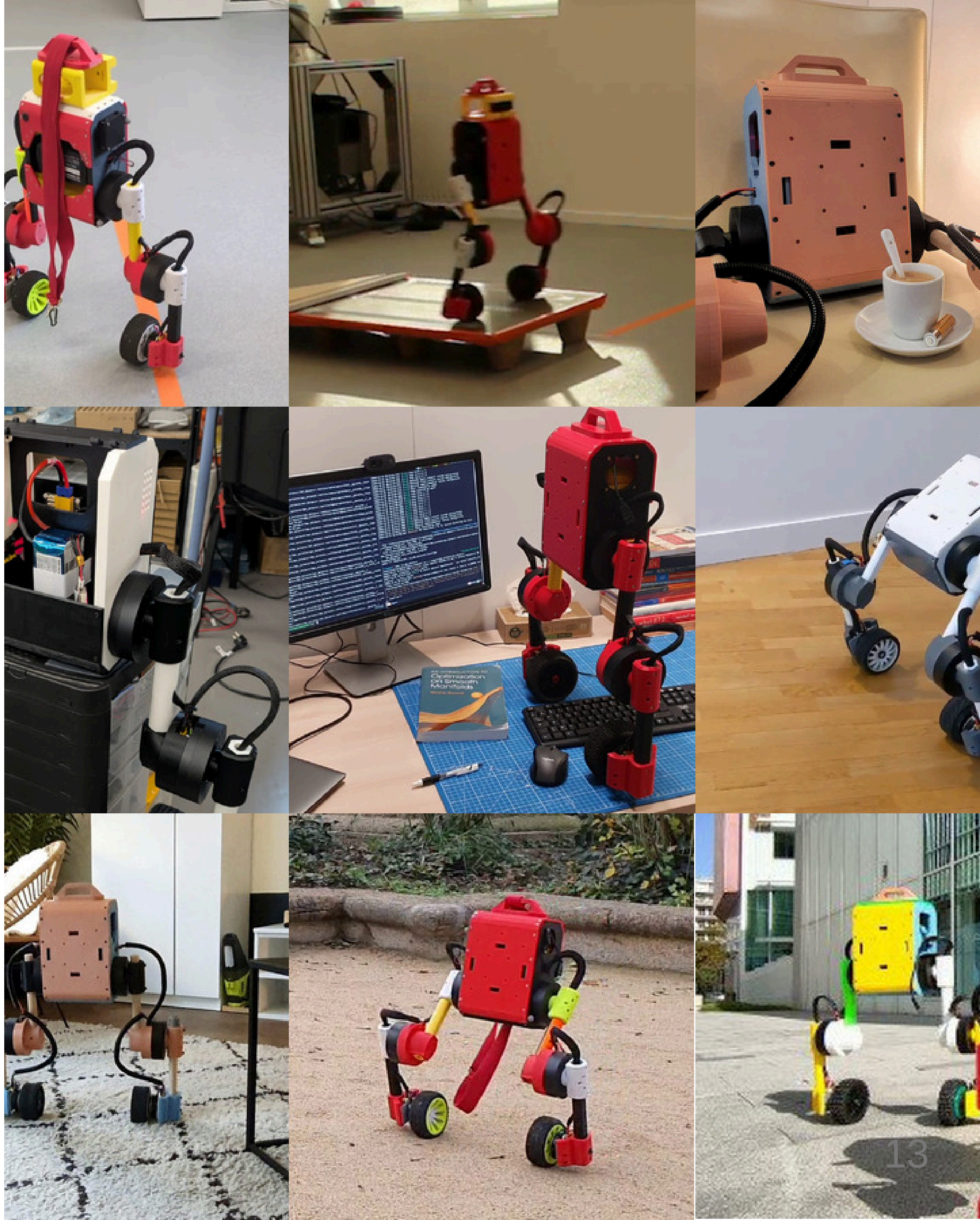
Simple thermal modeling

A thermal model of a system is one that relates the quantities evolve over time. For any given system, you can imagine it as a (the thermal load), heat goes out the other side (cooling) and t changes over time. In practical systems, the cooling heat trans the system's temperature and ambient temperature and the w static quantity or an independent variable that is divorced fron

Upkie wheeled bipeds

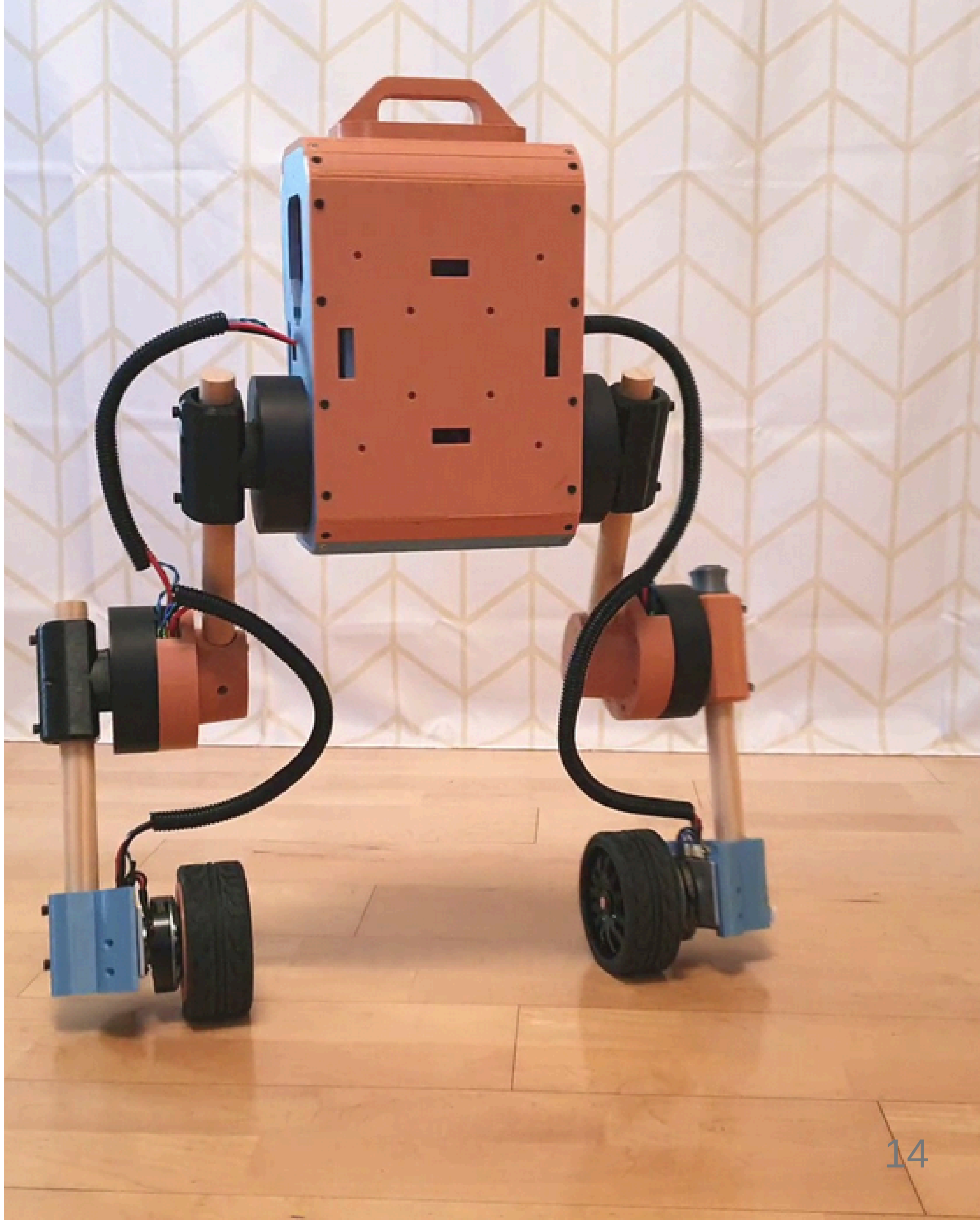
Upkie wheeled bipeds

- Released: 2022
- License: Apache-2.0
- Documentation: yes
- Reproduced: 6+
- Learning: yes
- Active: yes



Goals

- Open-source hardware and software
- Hybrid locomotion capabilities
- Reproducibility and accessibility
- Education and training
    - Model predictive control
    - Reinforcement learning
- Research platform

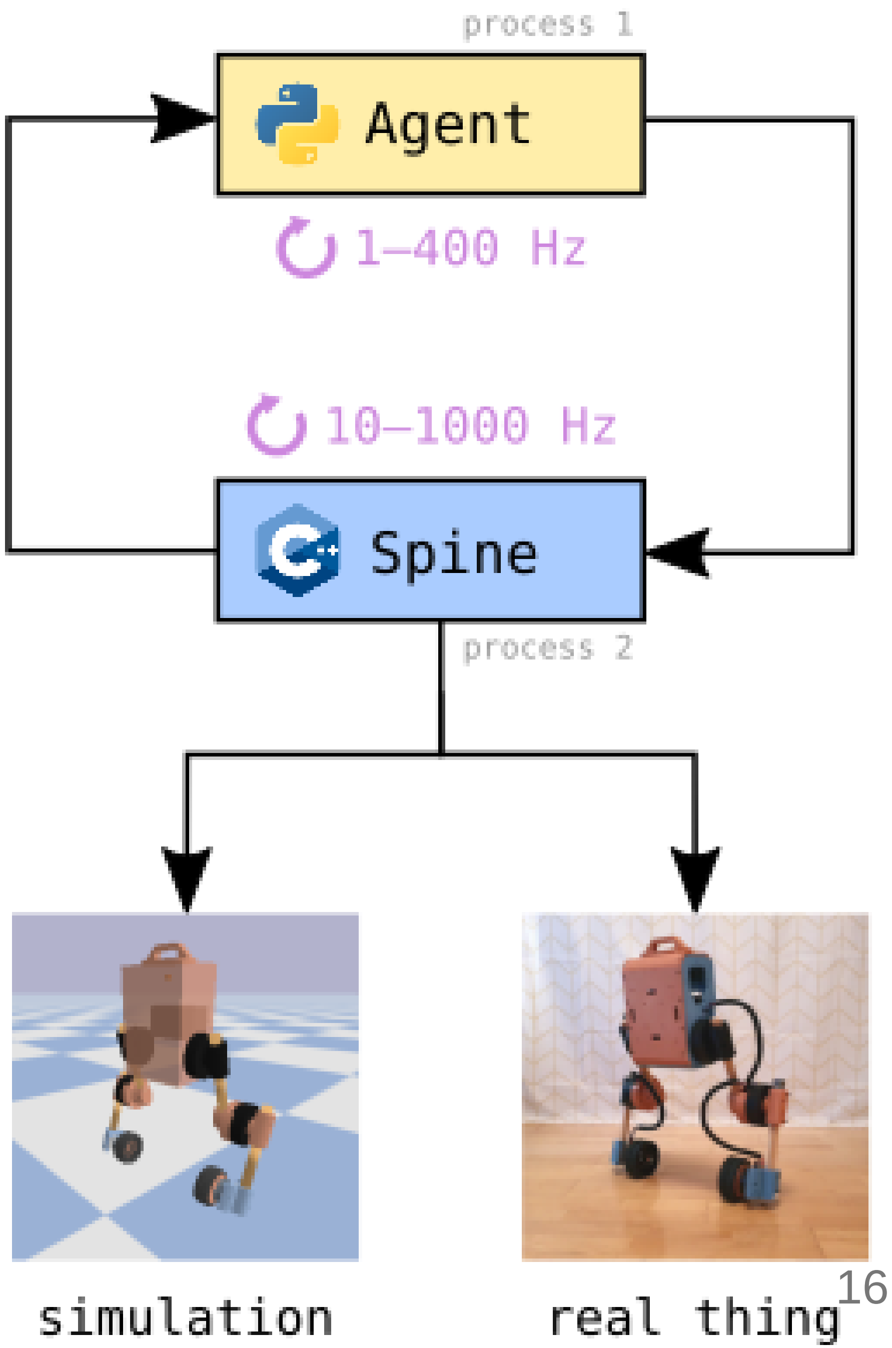


Upkies' software

Software

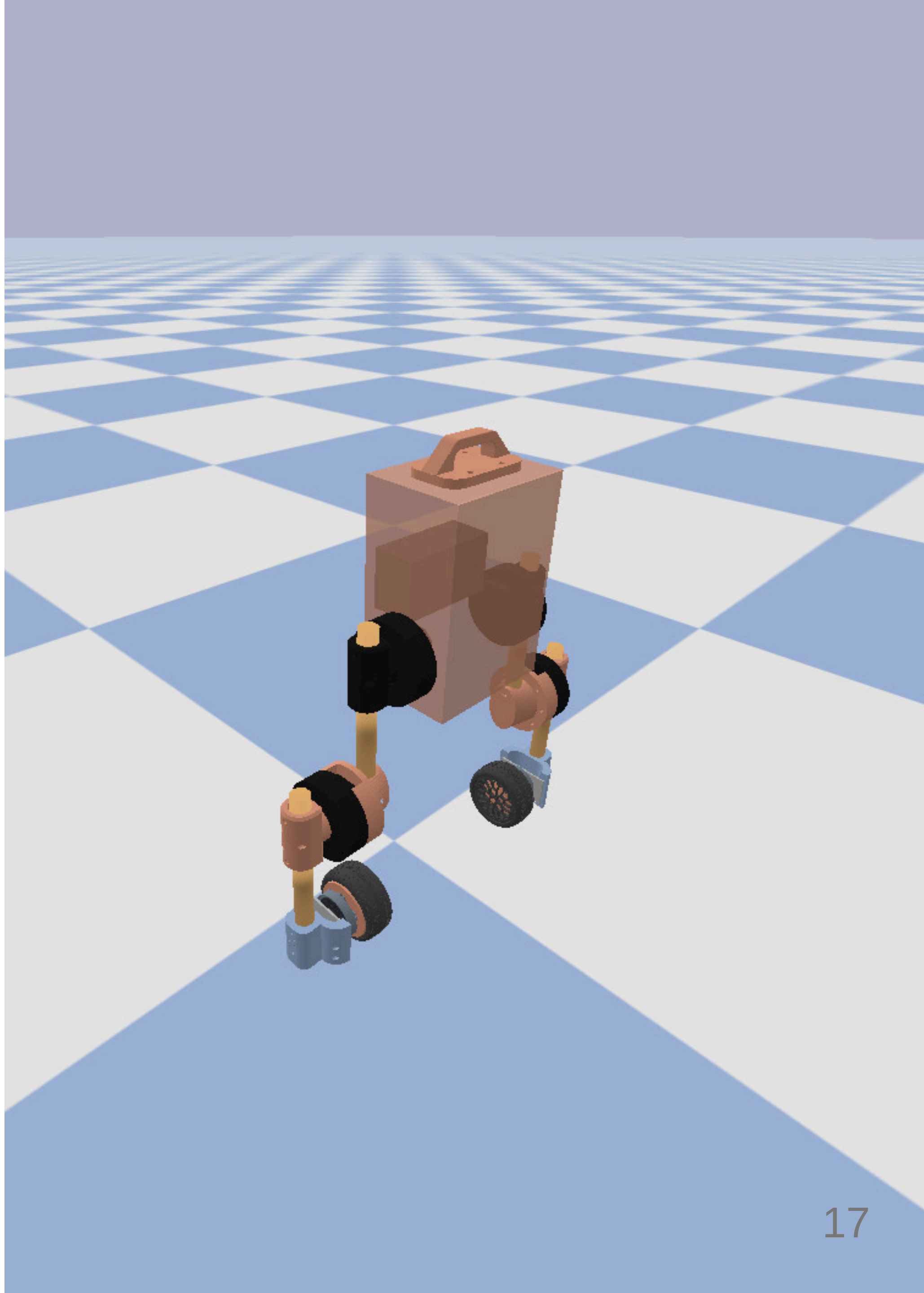
```
pip install upkie
```

- **Agents** in Python for new behaviors
- **Spines** in C++ for low-level motor control
- Logging and inter-process communication
- Gymnasium API for reinforcement learning
- Interface to different physics simulators
- Basic state estimation and control

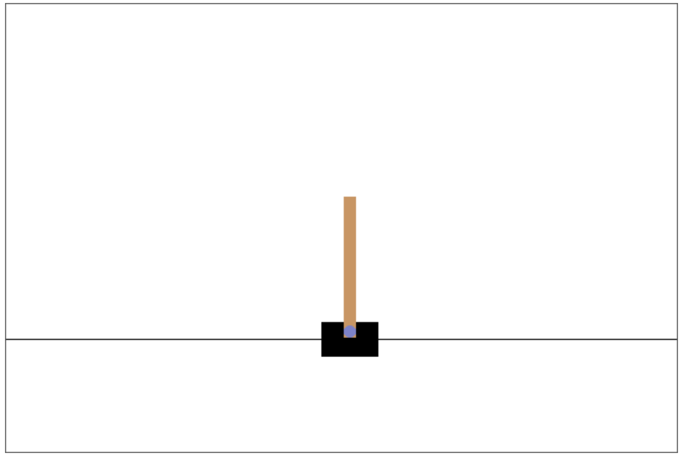


Try it out

```
$ git clone https://github.com/upkie/upkie
$ cd upkie
$ uv run examples/mpc_balancing.py
```



Gymnasium API



```python
with gym.make("CartPole-v1", render_mode="human") as env:
    observation, _ = env.reset()
    action = env.action_space.sample()
    for step in range(1_000_000):
        observation, reward, _, _, _ = env.step(action)
        position = observation[0]
        action = 0 if position > 0.0 else 1
```

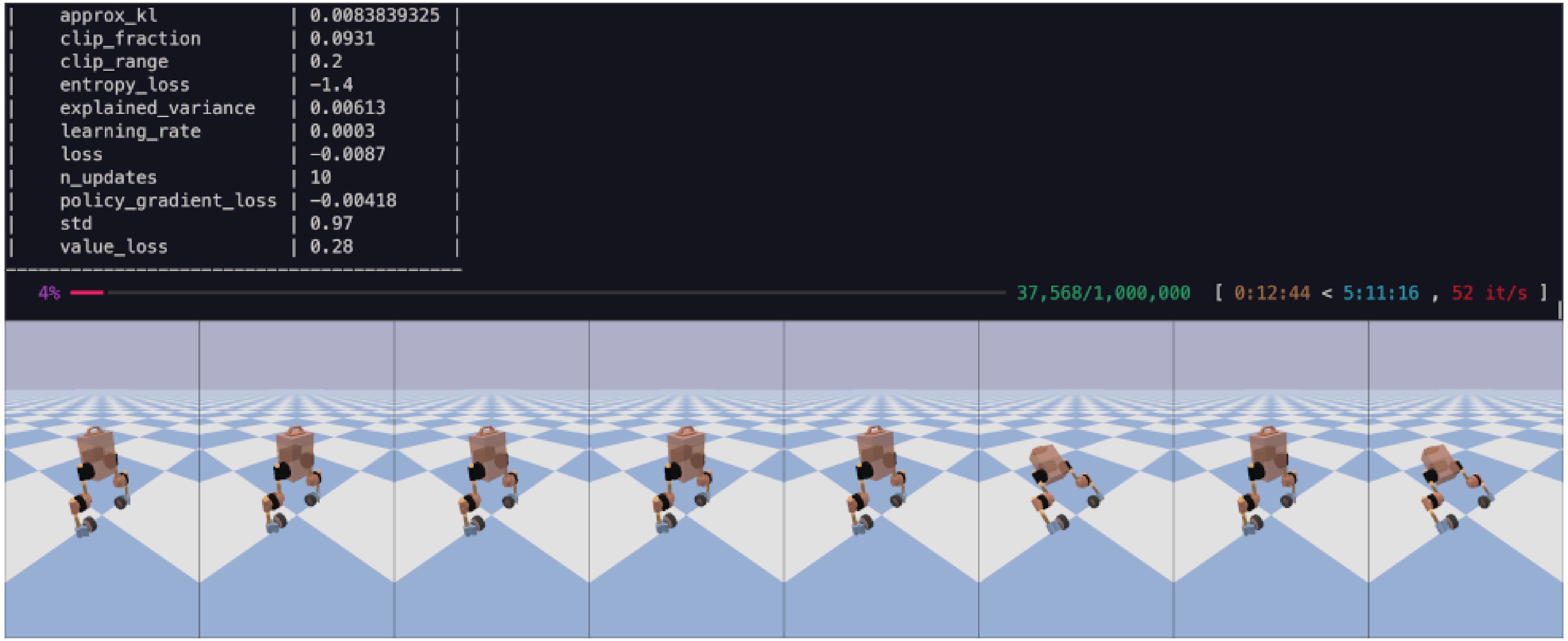
Gymnasium API

```
with gym.make("Upkie-PyBullet-Pendulum", frequency=200.0) as env:
    observation, _ = env.reset()
    action = env.action_space.sample()
    for step in range(1_000_000):
        observation, reward, _, _, _ = env.step(action)
        pitch = observation[0]
        action[0] = 10.0 * pitch  # action is [ground_velocity]
```

Stable-Baselines3

- Stable-Baselines3: reliable implementations of reinforcement-learning algorithms
- RL Baselines3 Zoo: training framework for Stable-Baselines3

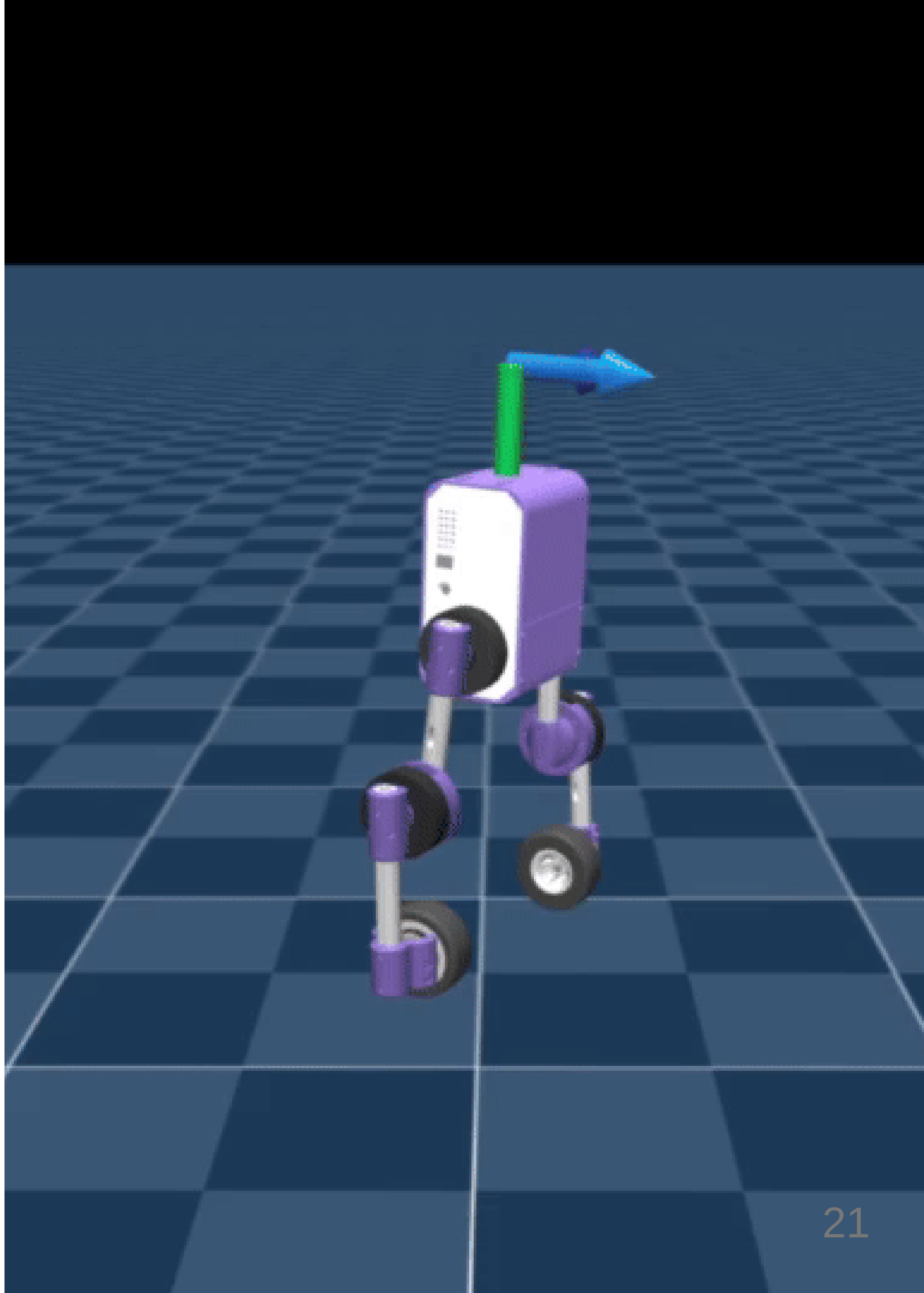
Thanks to the Gymnasium API, we can use these libraries to train agent policies:



```
|    approx_kl            | 0.0083839325 |
|    clip_fraction        | 0.0931       |
|    clip_range           | 0.2          |
|    entropy_loss         | -1.4         |
|    explained_variance   | 0.00613      |
|    learning_rate        | 0.0003       |
|    loss                 | -0.0087      |
|    n_updates            | 10           |
|    policy_gradient_loss | -0.00418     |
|    std                  | 0.97         |
|    value_loss           | 0.28         |
-------------------------------------------
   4% ━━                                                                37,568/1,000,000  [ 0:12:44 < 5:11:16 , 52 it/s ]
```

MjLab Upkie

- RL training environment by Marc Duclusaud
- Repository: mjlab_upkie
- GPU-optimized training based on mjlab:
    - Pro: train on e.g. 2048 environments
    - Con: requires an Nvidia GPU
- Larger FPS than current CPU-based sims
- Can train richer policies in less wall time



Lessons learned

Worked:

- Open source *everything*
- Use conda-forge then pixi
- Model predictive control works on various robots with only two parameters

Didn't work:

- Vulp: too much initial modularity
- Custom C++ spines: no usage so far
- Simulation spines for both testing and reinforcement learning

Applications and future works

Applications of Upkies:

- Articulated head doing marker tracking
- Contact estimation using machine learning from real-robot data
- Obstacle avoidance trained with Gaussian splatting and reinforcement learning

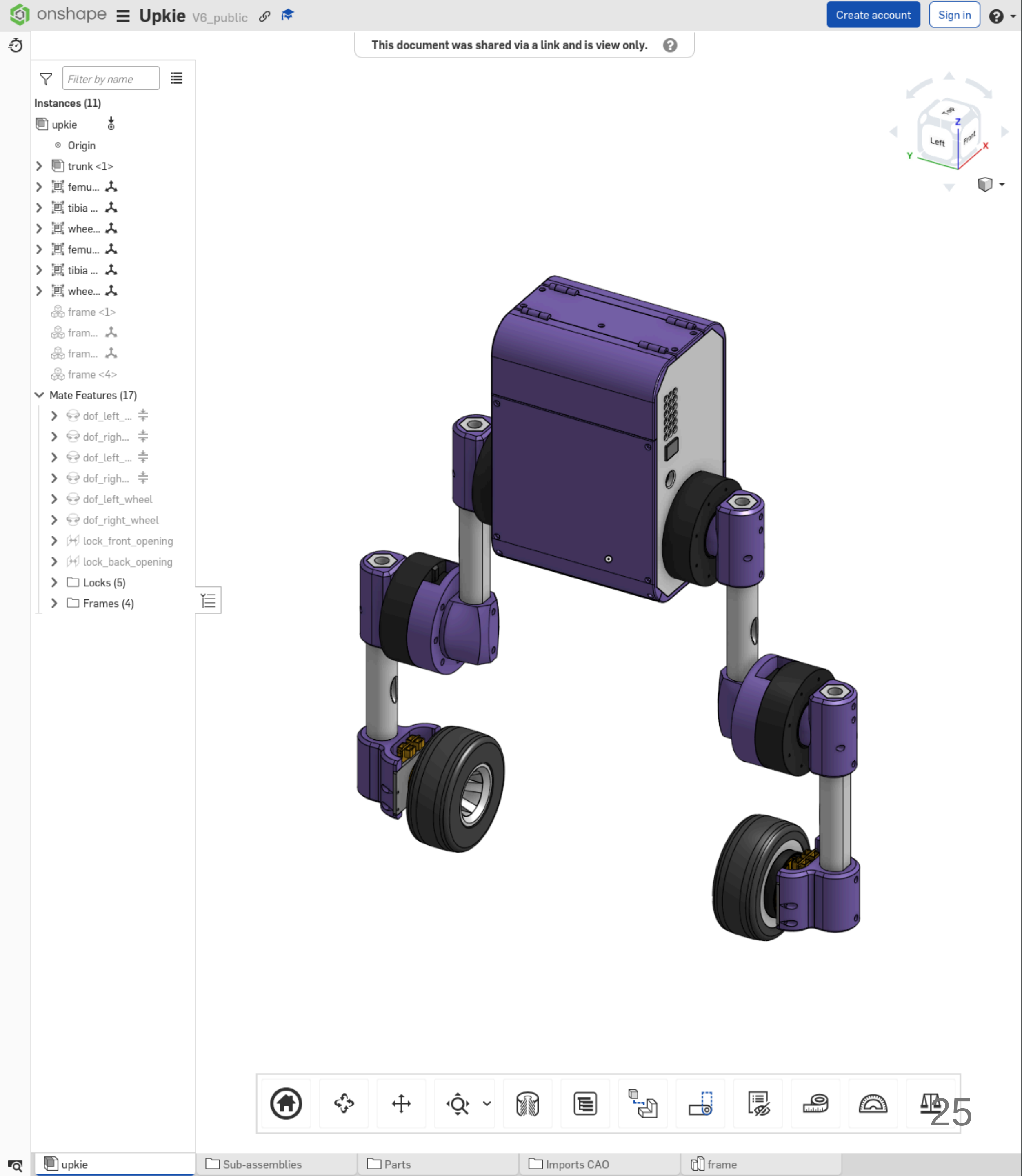
In the pipeline:

- Execute dynamic motions, like jumping
- Vision-based locomotion with RGB-D cameras

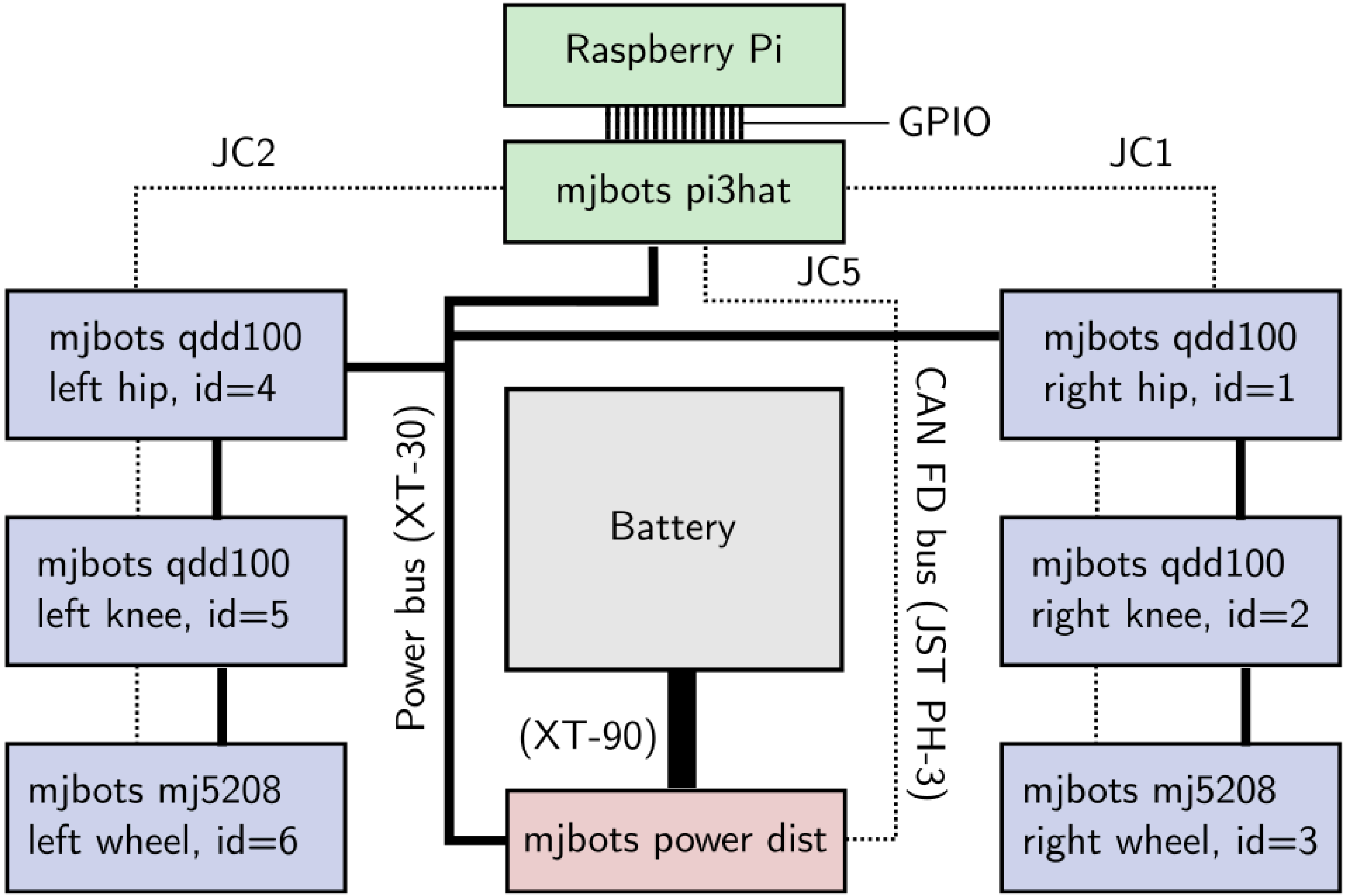
Upkies' hardware

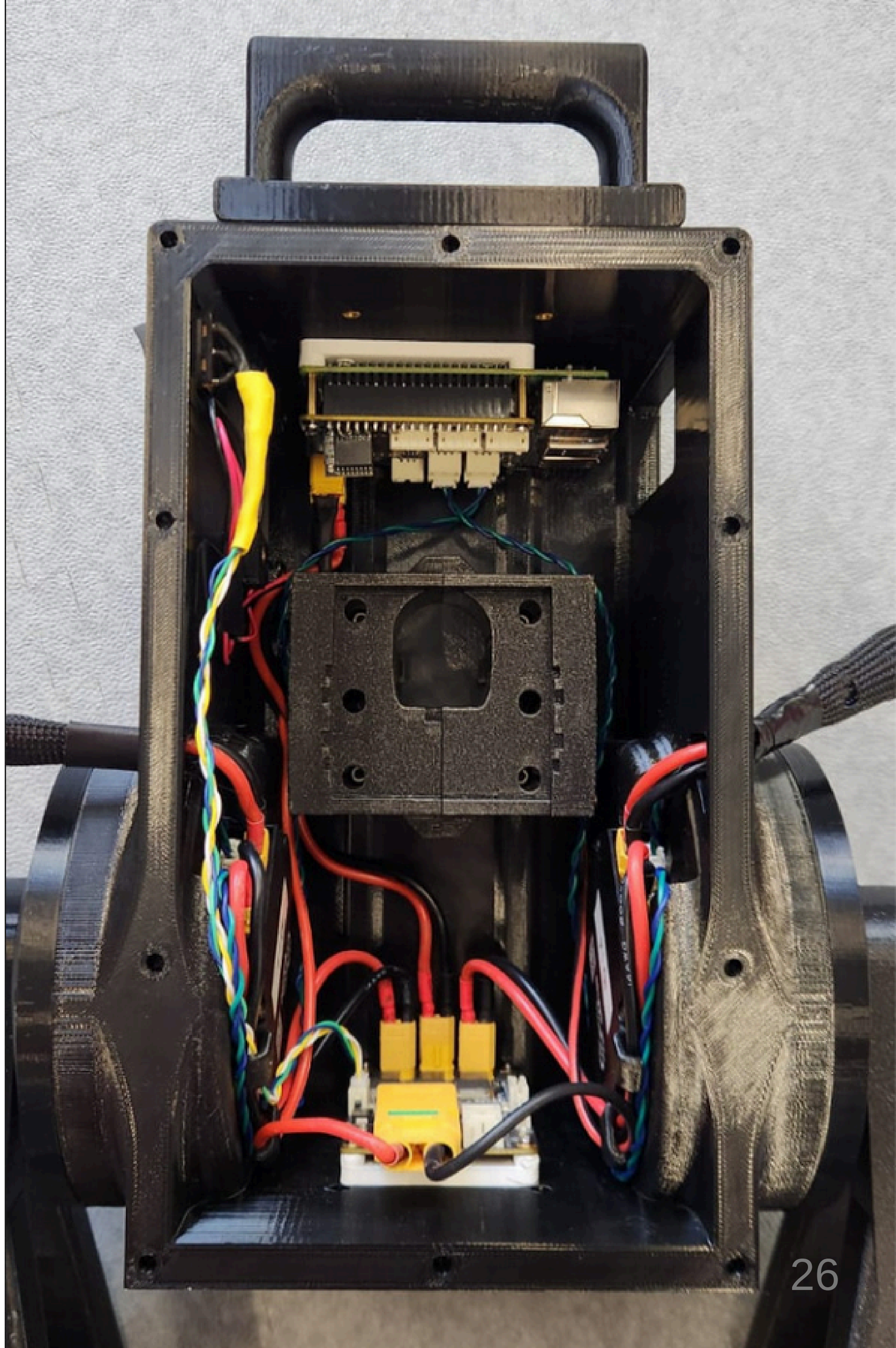
Hardware

- Build manual with pictures
- Bill of materials
- 3D printed parts repository
- Add-ons using four-screw patterns
- Variants and redesigns:
    - Version 1
    - Version 2
    - Michael Mathieu's Upkie-T
    - Marc Duclusaud's redesign


This document was shared via a link and is view only.

Electronics

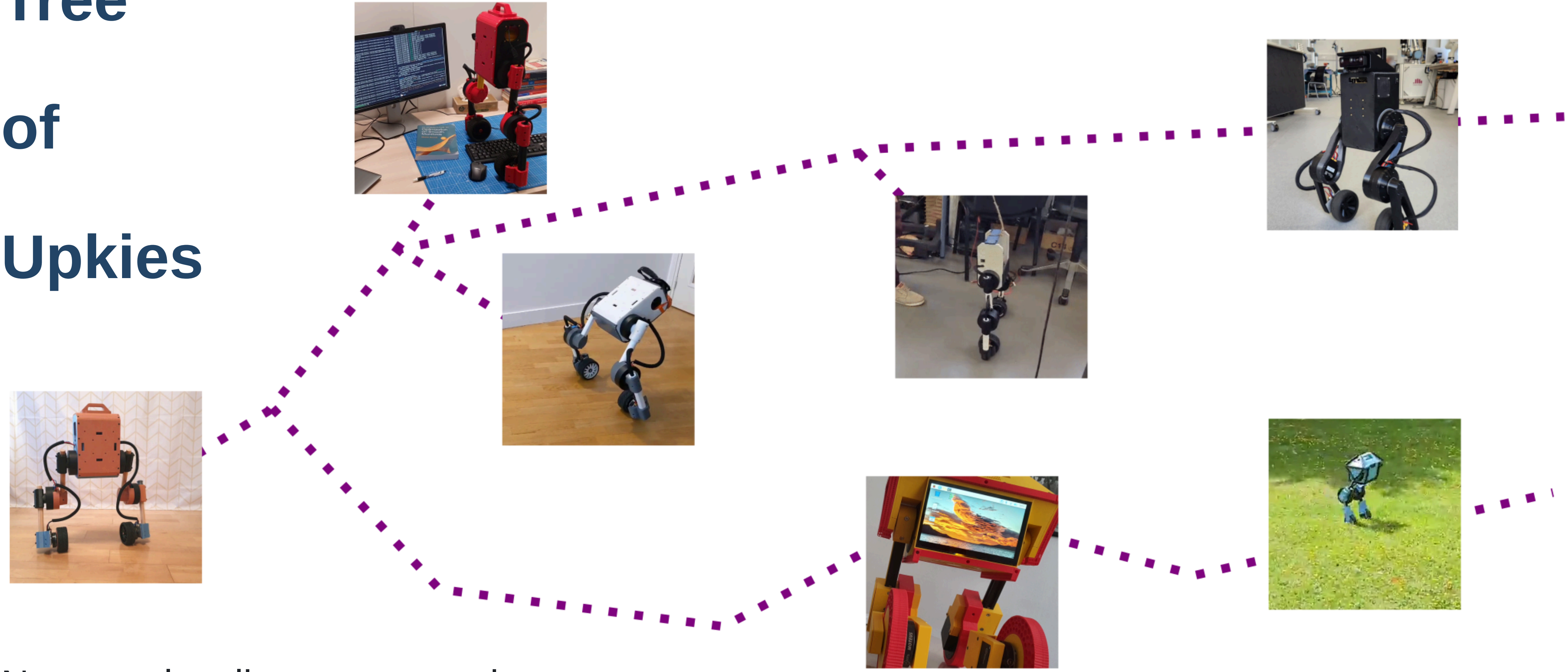




Tree

of

Upkies



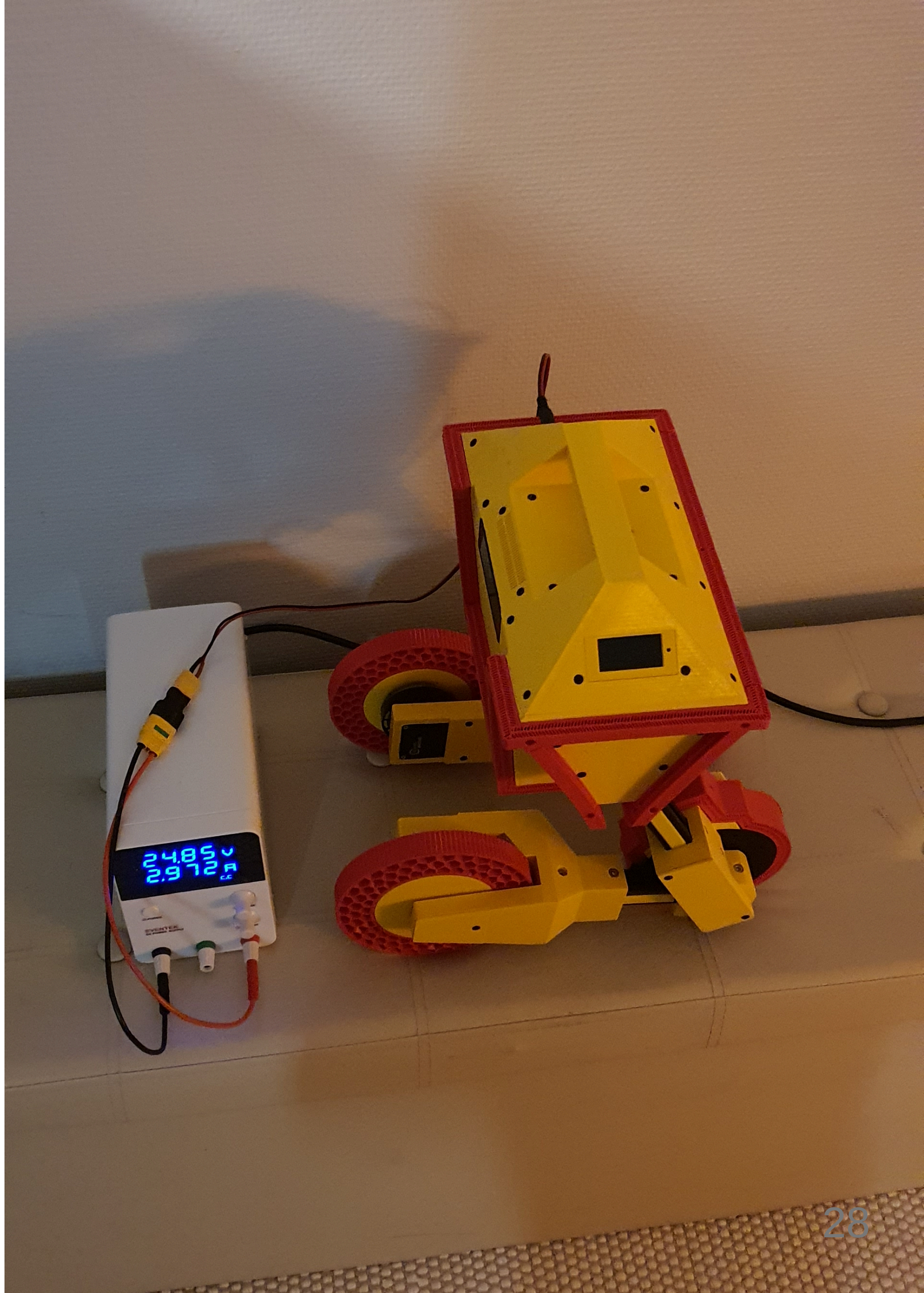
Not exactly a linear progression...

Upkie-T

Michael Mathieu's Upkie-T design:

- All cables are internal
- Battery cells and BMS are internal
- Bumpers for fall protection
- Charging while the robot is on
- Custom 3D-printed wheels
- Screen on the robot's back

Direction: more features, harder to reproduce

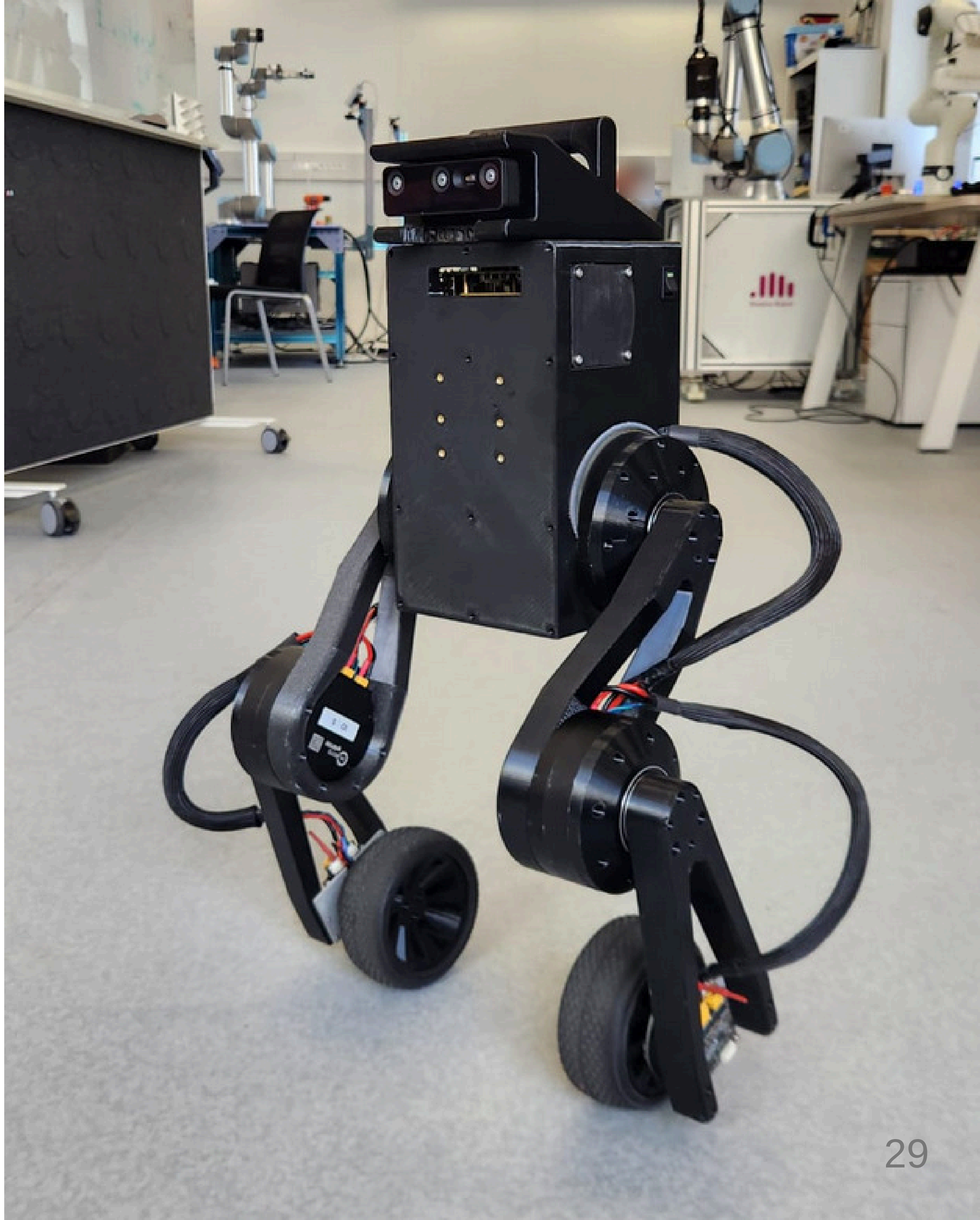


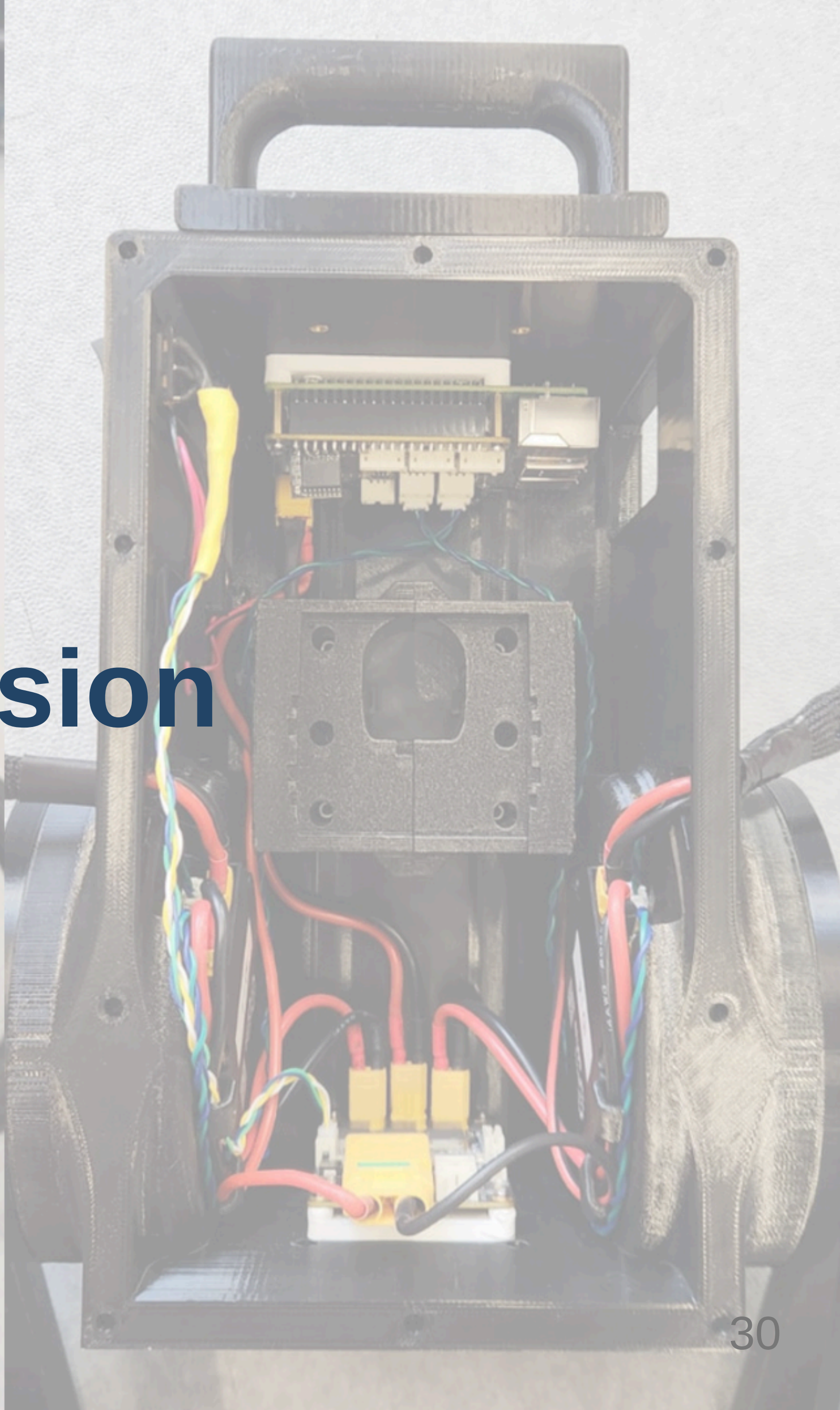
Upkie v2

Etienne Arlaud's and Valentin Tordjman--Levavasseur's revision:

- Redesign parts with FreeCAD
- Reduce number of 3D printed parts
- Ease electronics/cable management
- Reduce play in hip and knee joints
- Improve durability of the robot
- Add mounting patterns for extensions
- Improve legs range of motion

Direction: streamlining, easier to reproduce





Live demo & Discussion

Extra slides

Definition of open source for robots?

Defining “open source” for robots:

1. **Software:** open-source license (copyleft or permissive)
2. **Mechanics:** Creative Commons? CERN Open Hardware License?
3. **Electronics:** Creative Commons? TARP Open Hardware License?
4. **Datasets:** Create Commons? Open Database License?

Copyleft and permissive licenses were written *for software*.

Open-source hardware

There is an OSHW Definition:

- "Hardware": anything physical with public source files
- Definition applies to electronics and mechanical designs
- Requires sharing the files to build *and* modify the hardware

See also: OSHW certification program.