Reinforcement Learning for Athletic Intelligence: Lessons from the 1st "AI Olympics with RealAIGym" Competition

Felix Wiebe[1] , **Niccolò Turcato**[2] , **Alberto Dalla Libera**[2] , **Chi Zhang**[3] , **Theo Vincent**[4] , **Shubham Vyas**[1] , **Giulio Giacomuzzo**[2] , **Ruggero Carli**[2] , **Diego Romeres**[5] , **Akhil Sathuluri**[3] , **Markus Zimmermann**[3] , **Boris Belousov**[4] , **Jan Peters**[4,6,9,10] , **Frank Kirchner**[1,7] and **Shivesh Kumar**[1,8]

[1]Robotics Innovation Center, German Research Center for Artificial Intelligence (DFKI), Germany [2]Department of Information Engineering, University of Padova, Italy [3]Technical University of Munich, Germany [4]Systems AI for Robot Learning, DFKI, Germany [5]Mitsubishi Electric Research Lab (MERL), USA [6]Technical University of Darmstadt, Germany [7]University of Bremen, Germany [8]Chalmers University of Technology, Sweden [9]Centre for Cognitive Science, Darmstadt [10]Hessian.AI
felix.wiebe@dfki.de

Abstract

As artificial intelligence gains new capabilities, it becomes important to evaluate it on real-world tasks. In particular, the fields of robotics and reinforcement learning (RL) are lacking in standardized benchmarking tasks on real hardware. To facilitate reproducibility and stimulate algorithmic advancements, we held an AI Olympics competition at IJCAI 2023 conference based on the double pendulum system in the RealAIGym project where the participants were asked to develop a controller for the swing up and stabilization task. This paper presents the methods and results from the top participating teams and provides insights into the real-world performance of RL algorithms with respect to a baseline time-varying LQR controller.

1 Introduction

Benchmarks and competitions have proven extremely successful in computer vision and machine learning for driving algorithmic innovation [Deng *et al.*, 2009]. In robotics, a number of simulated benchmarking environments gained popularity for evaluating learning algorithms [Brockman *et al.*, 2016; James *et al.*, 2020; Mittal *et al.*, 2023; Al-Hafez *et al.*, 2023]. However, only a handful of *real-robot* standardized environments are reliable, easy to simulate, and cheap to reproduce. Importantly, different environments have different focus, e.g., locomotion [Grimminger *et al.*, 2020; Feng *et al.*, 2023], finger-based manipulation [Funk *et al.*, 2021; Gürtler *et al.*, 2023], etc. Our RealAIGym project [Wiebe *et al.*, 2022b][1] offers a suite of canonical underactuated systems (simple pendulum [Wiebe *et al.*, 2022a], AcroMonk [Javadi *et al.*, 2023], hopper [Soni *et al.*, 2023]) for benchmarking learning and control algorithms for athletic intelligence on real hardware with a user-friendly Python API. By open-sourcing both the software and the hardware, we aim to establish a real-world equivalent to the well-known OpenAI Gym [Brockman *et al.*, 2016].

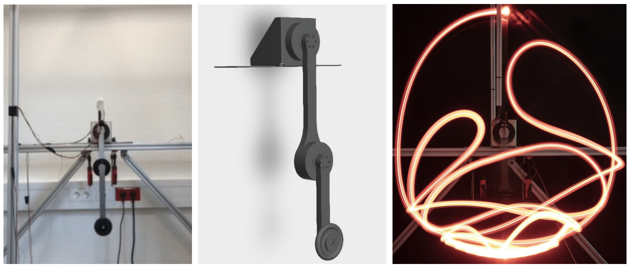

Figure 1: Double pendulum: real system, simulation, and a long-exposure photo of its free-fall showing its chaotic dynamics.

The "AI Olympics with RealAIGym" competition[2][3] is based on a Double Pendulum system (see Fig. 1) in the RealAIGym project. The Double Pendulum [Wiebe *et al.*, 2023] can operate in two modes: *Pendubot* when the actuator in the shoulder joint is active but the elbow is passive, and *Acrobot* with passive shoulder and active elbow. The challenge consists in performing a swing-up and stabilization from the free-hanging position to the upright position. For scoring, the Acrobot and Pendubot tasks are treated as two separate tracks.

The competition was held in two phases. In the simulation phase, the participants were provided model parameters, a simulation environment, and the scoring metric to rank the behaviors. The submitted controllers were evaluated with a performance score and a robustness score. The *performance score* evaluates how well the swing-up is performed and is given by a weighted sum of several terms, such as swing-up success, swing-up time, energy cost, torque smoothness, etc. The *robustness score* evaluates the sensitivity of the controllers to model inaccuracies, measurement noise, torque noise, torque response, and time delay.

The teams who achieved high simulation and robustness scores were advanced to the hardware phase, where they were given up to 20h remote-access to a real double pendulum system to train and test their controllers. The final controllers were evaluated on the same PC using the performance score over 10 consecutive executions to provide the final ranking.

[1] https://dfki-ric-underactuated-lab.github.io/real-ai-gym/

[2] https://ijcai-23.dfki-bremen.de/competitions/ai_olympics/

[3] https://youtu.be/eYDH1v1FqF8

2 Methods

In this section, three families of RL methods that participated in the competition are presented, together with a baseline optimal control approach for comparison.

2.1 Model-based RL: MCPILCO

MC-PILCO (Monte Carlo - Probabilistic Inference for Learning COntrol) [Amadio *et al.*, 2022] is a model-based policy gradient algorithm that relies on Gaussian Processes (GPs) to learn the system dynamics from data. Let $\boldsymbol{x}_t$ and $\boldsymbol{u}_t$ be, respectively, the state and input of the system at step t. A cost function $c(\boldsymbol{x}_t)$ encodes the task to be accomplished. A policy $\pi_{\boldsymbol{\theta}} : \boldsymbol{x} \to \boldsymbol{u}$ that depends on the parameters $\boldsymbol{\theta}$ selects the inputs applied to the system. The objective is to find policy parameters $\boldsymbol{\theta}^*$ that minimize the cumulative expected cost:

$$J(\boldsymbol{\theta}) = \sum_{t=0}^{T} \mathbb{E}[c(\boldsymbol{x}_t)], \quad x_0 \sim p(\boldsymbol{x}_0). \tag{1}$$

MC-PILCO performs several successive attempts to solve the desired task, also called trials. Each trial consists of three main phases: (i) model learning, (ii) policy update, and (iii) policy execution.

In the model learning step, previous experience is used to derive a one-step-ahead stochastic model of the system dynamics using Gaussian Process Regression. The policy update step aims at minimizing the cost in eq. (1) w.r.t. the policy parameters $\boldsymbol{\theta}$. The expectation in eq. (1) is approximated based on the GP dynamics derived in (i) and Monte Carlo simulation. Finally, in the last step, the current optimized policy is applied to the system and the collected samples are stored to update the model in the next trials. Examples of MC-PILCO applications have been reported in [Amadio *et al.*, 2023] and [Turcato *et al.*, 2023].

2.2 Model-free RL, Actor-Critic Methods: SAC

A potential solution in the field of model-free reinforcement learning involves combining the Soft Actor Critic (SAC) [Haarnoja *et al.*, 2018] algorithm with the Linear Quadratic Regulator (LQR). The SAC algorithm is utilized to train a reinforcement learning (RL) agent for swing-up tasks, while the LQR controller is responsible for stabilizing the system at its highest position.

The transition between the SAC and LQR is managed using a Region of Attraction (RoA) approach. The RoA for the LQR controller is numerically approximated [Maywald *et al.*, 2022], and the switch occurs when the system's state enters the RoA.

A three-stage reward function is designed to steer the agent into the LQR controller's RoA, facilitating the switch. The initial stage involves a quadratic function that penalizes state errors and torque usage. Upon the end-effector reaching a specific threshold line, a reward is introduced. A substantial reward is granted when the state is within the LQR's RoA.

Efforts to bridge the sim-to-real gap include domain randomization, noisy validation, and early termination. These approaches involve training and testing the agent in an environment that mimics real-world features and disturbances. Real-world constraints are integrated as termination conditions during training.

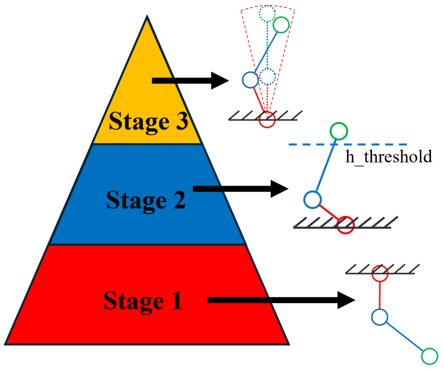

Figure 2: The 3-stage reward function: a funneling approach to driving the agent learning behavior. At each subsequent stage, the designated area of the state space shrinks, providing additional guidance to the learner. Such shaped reward function was found to significantly improve the learning speed of SAC and DQN. Both methods switch to LQR for stabilization to improve the smoothness of actions around the unstable equilibrium.

2.3 Model-free RL, Value-based Methods: DQN

Deep Q-Network (DQN) [Mnih *et al.*, 2015] is the seminal algorithm in the field of Deep RL, which has demonstrated that end-to-end controllers with high-dimensional observation spaces can be learned by trial and error. DQN is a value-based method that learns the optimal action-value function $Q(x, u)$ such that the optimal action can be obtained by optimization over u. To learn Q, DQN iteratively applies the Bellman operator, which is a contraction mapping and hence its successive application leads to a fixed point. The theory guarantees that the fixed point of the Bellman operator is the optimal action-value function corresponding to the optimal policy, yielding the maximal reward / minimal cost.

DQN is known to work well in discrete action spaces. Since the action space of Pendubot is 1-dimensional, we consider discretizing it into 9 bins. A logarithmic discretization centered around zero yields better performance in practice, because the agent requires more actions around zero to control the pendulum close to the upright position. The same rewards function as for the SAC algorithm is used for training DQN (Fig. 2), and the LQR controller provided by RealAIGym is employed to stabilize the Pendubot when it enters the region of attraction.

2.4 Model-based Optimal Control: TVLQR

An optimal control (OC) controller serves as a baseline for the learning-based controllers in the competition. The controller utilizes three versions of the linear quadratic regulator (LQR). The basis is a trajectory $(\mathbf{u}_t^d, \mathbf{x}_t^d)$ computed with the trajectory optimization technique iterative LQR (iLQR) [Li. and Todorov., 2004] and a time-varying LQR (TVLQR) [Tedrake, 2022] is used to stabilize the system towards that trajectory during the execution. This results in the control law at time t:

$$\mathbf{u}_t(\mathbf{x}_t) = \mathbf{u}_t^d - \mathbf{K}_t(\mathbf{x}_t - \mathbf{x}_t^d) \tag{2}$$

with the linear feedback matrix $\mathbf{K}_t$. The controller switches to a stabilizing LQR control once its region of attraction around the unstable fixpoint is entered.

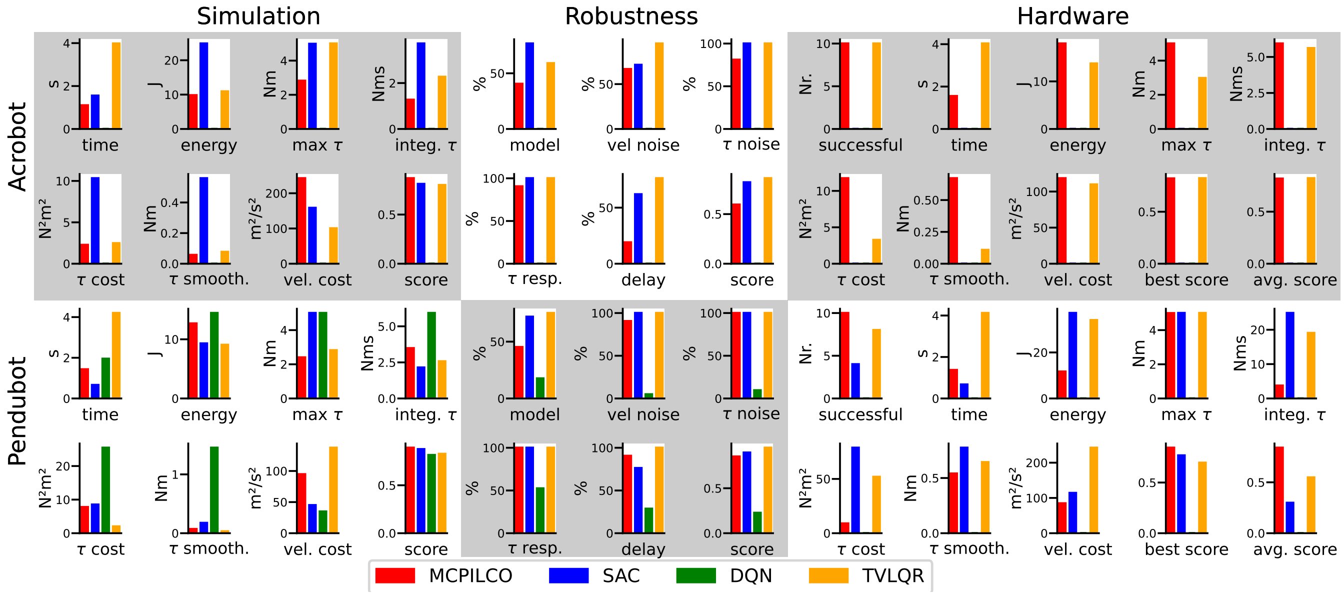

Figure 3: Results for all controllers on acrobot and pendubot in the three competition categories. Note that in the 'Simulation' and 'Hardware' categories smaller values are better (except for successful attempts and the scores) while in the 'Robustness' category larger values are better. Bars for controllers which did not compete in a category are left blank.

3 Results

The results of the competition are listed in Table 1 and visualized in Fig. 3. All data and figures can be found online[4]. The results are explained and contextualized first for the acrobot and then for the pendubot in the following subsections.

3.1 Acrobot

Two submitted controllers (MC-PILCO and SAC) passed the simulation stage for the acrobot swingup problem. MC-PILCO achieved a high performance score (0.869), beating the optimal control baseline (0.8) and a mediocre robustness score (0.595). SAC's performance score is a little lower (0.811) while the robustness score is significantly higher (0.82) and closer to the OC baseline (0.861). Despite the high performance and robustness scores, the SAC team did not succeed in transferring the controller to the real hardware. A successful swing up was not achieved. The MC-PILCO team, on the other hand, retrained their policy on the real hardware and demonstrated a 100% success rate in the 10 evaluation trials resulting in a hardware score of 0.817 which is only slightly lower than the OC baseline (0.821). MC-PILCO's mediocre robustness score indicates that a policy trained with MC-PILCO has limits when exposed to dynamics which have not been experienced during training. However, the notable performance of MC-PILCO on the real hardware shows its applicability even for real dynamics when directly trained under these conditions.

3.2 Pendubot

On the pendubot, MC-PILCO, SAC and DQN passed the simulation stage of the competition. MC-PILCO scored the highest in the simulation (MC-PILCO: 0.891, SAC: 0.876, DQN: 0.815, TVLQR: 0.827) while SAC showed the highest robustness score only secondary to TVLQR (MC-PILCO: 0.852, SAC: 0.986, DQN: 0.226, TVLQR: 0.950). MC-PILCO again was retrained on the real hardware and achieved a 100% success rate with a hardware score of 0.839. SAC achieved a 40% success rate and a hardware score of 0.298. Note that the SAC policy was not retrained on the hardware but only in a simulation with added noise. DQN could not be successfully transferred to the real system. The OC baseline has a 80% success rate and a 0.547 score.

		MC-PILCO	SAC	DQN	TVLQR
Acrobot	Simulation	**0.869**	0.811	-	0.8
	Robustness	0.595	0.820	-	**0.861**
	Hardware	0.817	-	-	**0.821**
Pendubot	Simulation	**0.891**	0.876	0.815	0.827
	Robustness	0.852	0.896	0.226	**0.950**
	Hardware	**0.839**	0.298	-	0.547

Table 1: Final Scores of the submitted controllers and the OC baseline for both systems.

4 Conclusion

We introduced a competition to evaluate the performance and robustness of learning-based vs. optimal control approaches. From the 1st run, we conclude that learning-based approaches provide a strong alternative to model-based optimal control, being able to learn from a few samples on the real system. The model-based MC-PILCO outperformed the model-free methods SAC and DQN, showing a more reliable performance on the real Pendubot system than the baseline TVLQR method. We hope this challenge inspires further research into control algorithms for athletic intelligence; in particular, tackling generalization to different initial conditions, robustness to perturbations, handling of unmodelled dynamic effects.

[4] https://dfki-ric-underactuated-lab.github.io/real_ai_gym_leaderboard/

Ethical Statement

There are no ethical issues.

Contribution Statement

NT, ADL, GG, RC and DR constitute the winning team and contributed to the MC-PILCO controller. CZ, AS and MZ formed the runner-up team which implemented the SAC+LQR controller. TV and BB provided the DQN implementation. FW, SV and SK implemented the baseline controller and benchmarking software. SK and BB conceptualized and executed the competition at the IJCAI venue. All authors contributed to the preparation of this manuscript.

Acknowledgments

The competition organizers at DFKI-RIC (FW, SV, FK, SK) acknowledge the support of M-RoCK project funded by the German Aerospace Center (DLR) with federal funds (Grant Number: FKZ 01IW21002) from the Federal Ministry of Education and Research (BMBF) and is additionally supported with project funds from the federal state of Bremen for setting up the Underactuated Robotics Lab (Grant Number: 201-342-04-2/2021-4-1). TV, BB, and JP acknowledge the grant "Einrichtung eines Labors des Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI) an der Technischen Universität Darmstadt" of the Hessisches Ministerium für Wissenschaft und Kunst. This research was supported by Research Clusters "The Adaptive Mind" and "Third Wave of AI", funded by the Excellence Program of the Hessian Ministry of Higher Education, Science, Research and the Arts. Parts of the calculations for this research were conducted on the Lichtenberg high-performance computer of the TU Darmstadt. TV and BB also thank Daniel Palenicek and Tim Schneider for their support while performing the experiments on the IAS group's computing cluster at TU Darmstadt.

References

[Al-Hafez *et al.*, 2023] F. Al-Hafez, G. Zhao, J. Peters, and D. Tateo. Locomujoco: A comprehensive imitation learning benchmark for locomotion. In *Robot Learning Workshop, Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

[Amadio *et al.*, 2022] Fabio Amadio, Alberto Dalla Libera, Riccardo Antonello, Daniel Nikovski, Ruggero Carli, and Diego Romeres. Model-based policy search using monte carlo gradient estimation with real systems application. *IEEE Transactions on Robotics*, 38(6):3879–3898, 2022.

[Amadio *et al.*, 2023] Fabio Amadio, Alberto Dalla Libera, Daniel Nikovski, Ruggero Carli, and Diego Romeres. Learning control from raw position measurements. In *2023 American Control Conference (ACC)*, pages 2171–2178. IEEE, 2023.

[Brockman *et al.*, 2016] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[Feng *et al.*, 2023] Gilbert Feng, Hongbo Zhang, Zhongyu Li, Xue Bin Peng, Bhuvan Basireddy, Linzhu Yue, Zhitao Song, Lizhi Yang, Yunhui Liu, Koushil Sreenath, et al. Genloco: Generalized locomotion controllers for quadrupedal robots. In *Conference on Robot Learning*, pages 1893–1903. PMLR, 2023.

[Funk *et al.*, 2021] Niklas Funk, Charles Schaff, Rishabh Madan, Takuma Yoneda, Julen Urain De Jesus, Joe Watson, Ethan K Gordon, Felix Widmaier, Stefan Bauer, Siddhartha S Srinivasa, et al. Benchmarking structured policies and policy optimization for real-world dexterous object manipulation. *IEEE Robotics and Automation Letters*, 7(1):478–485, 2021.

[Grimminger *et al.*, 2020] Felix Grimminger, Avadesh Meduri, Majid Khadiv, Julian Viereck, Manuel Wüthrich, Maximilien Naveau, Vincent Berenz, Steve Heim, Felix Widmaier, Thomas Flayols, et al. An open torque-controlled modular robot architecture for legged locomotion research. *IEEE Robotics and Automation Letters*, 5(2):3650–3657, 2020.

[Gürtler *et al.*, 2023] Nico Gürtler, Sebastian Blaes, Pavel Kolev, Felix Widmaier, Manuel Wuthrich, Stefan Bauer, Bernhard Schölkopf, and Georg Martius. Benchmarking offline reinforcement learning on real-robot hardware. In *The Eleventh International Conference on Learning Representations*, 2023.

[Haarnoja *et al.*, 2018] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[James *et al.*, 2020] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

[Javadi *et al.*, 2023] Mahdi Javadi, Daniel Harnack, Paula Stocco, Shivesh Kumar, Shubham Vyas, Daniel Pizzutilo, and Frank Kirchner. Acromonk: A minimalist underactuated brachiating robot. *IEEE Robotics and Automation Letters*, 8(6):3637–3644, 2023.

[Li. and Todorov., 2004] Weiwei Li. and Emanuel Todorov. Iterative linear quadratic regulator design for nonlinear biological movement systems. In *Proceedings of the First International Conference on Informatics in Control, Automation and Robotics - Volume 1: ICINCO*, pages 222–229. INSTICC, SciTePress, 2004.

[Maywald *et al.*, 2022] Lasse Jenning Maywald, Felix Wiebe, Shivesh Kumar, Mahdi Javadi, and Frank Kirchner. Co-optimization of acrobot design and controller for increased certifiable stability. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2636–2641. IEEE, 2022.

[Mittal *et al.*, 2023] Mayank Mittal, Calvin Yu, Qinxi Yu, Jingzhou Liu, Nikita Rudin, David Hoeller, Jia Lin Yuan, Ritvik Singh, Yunrong Guo, Hammad Mazhar, et al. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 2023.

[Mnih *et al.*, 2015] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[Soni *et al.*, 2023] Raghav Soni, Daniel Harnack, Hauke Isermann, Sotaro Fushimi, Shivesh Kumar, and Frank Kirchner. End-to-end reinforcement learning for torque based variable height hopping. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7531–7538, 2023.

[Tedrake, 2022] Russ Tedrake. *Underactuated Robotics*. 2022.

[Turcato *et al.*, 2023] Niccolo' Turcato, Alberto Dalla Libera, Giulio Giacomuzzo, and Ruggero Carli. Teaching a robot to toss arbitrary objects with model-based reinforcement learning. In *2023 9th International Conference on Control, Decision and Information Technologies (CoDIT)*, pages 1126–1131, 2023.

[Wiebe *et al.*, 2022a] Felix Wiebe, Jonathan Babel, Shivesh Kumar, Shubham Vyas, Daniel Harnack, Melya Boukheddimi, Mihaela Popescu, and Frank Kirchner. Torque-limited simple pendulum: A toolkit for getting familiar with control algorithms in underactuated robotics. *Journal of Open Source Software*, 7(74):3884, 2022.

[Wiebe *et al.*, 2022b] Felix Wiebe, Shubham Vyas, Lasse Jenning Maywald, Shivesh Kumar, and Frank Kirchner. RealAIGym: Education and Research Platform for Studying Athletic Intelligence. In *Proceedings of Robotics Science and Systems Workshop Mind the Gap: Opportunities and Challenges in the Transition Between Research and Industry*, New York, July 2022.

[Wiebe *et al.*, 2023] Felix Wiebe, Shivesh Kumar, Lasse J. Shala, Shubham Vyas, Mahdi Javadi, and Frank Kirchner. Open source dual-purpose acrobot and pendubot platform: Benchmarking control algorithms for underactuated robotics. *IEEE Robotics Automation Magazine*, pages 2–13, 2023.