

Robot Learning with `lerobot`: A Tutorial

Francesco Capuano  ... Adil Zouitine  Pepijn Kooijmans  Thomas Wolf  Michel Aractingi 

 École Normale Supérieure Paris-Saclay,  Hugging Face

Abstract



Hugging Face

Recent advancements in foundation models have enabled unprecedented progress in robotics, particularly in the development of increasingly more capable generalist robotics models. Leveraging modern learning-based approaches, the robotics community produced the first general models for tasks including locomotion and manipulation, and many said results have been openly released to the open-source community. The premise of generalist models is to learn control strategies across tasks and different robot embodiments, a long-standing challenge in robot learning. This work aims at being a tutorial on the most common techniques used for modern robot learning, covering both the conceptual underpinnings of the most prominent approaches, and pairing them with reproducible implementations using `lerobot` (<https://github.com/huggingface/lerobot>), the full-stack open-source robotics library developed by Hugging Face. This tutorial is designed to serve as a self-contained, structured reference for robotics practitioners working with `lerobot`, as well as a practical guide for researchers in the field of robot learning.

1 Classical Robotics

Know your enemy [...]

Sun Tzu

TL;DR

Exploring learning-based techniques versus purely dynamics-based approaches is motivated by the interest in (1) generalizing to tasks and embodiments (2) reducing dependency on human expertise (3) leveraging historical trends on the production of data.

1.1 Artificial motion

Robotics is concerned with producing artificial motion in the physical world in a way that is useful, reliable and safe. Thus robotics is at its very core an inherently multidisciplinary domain: producing motion requires interfacing various hardware and software components, each at various levels. Defining a concept of usefulness typically depends on a profound understanding of the applicative domain considered, whereas ensuring reliable and safe execution more so relies on estimating the inherent uncertainty in the robot's actions, guaranteeing adaptiveness of control while optimizing for performance, and more. Knowledge of mechanical, electrical, and software engineering, as well as rigid-body mechanics and control theory have therefore been quintessential in robotics since the field first developed. Lately, Machine Learning (ML) has proven useful in complementing said disciplines, making usage of the robotics data currently becoming more and more available. As a direct consequence of its multi-disciplinary nature, robotics developed as a rather wide array of methods, all concerned with the main purpose of producing artificial motion in the physical world.

Methods to produce robotics motion range from traditional *explicit* models—leveraging precise descriptions of the mechanics of robots' rigid bodies and their interactions with eventual obstacles in the robots environment—to fully *learning-based* systems, offloading modelling the robot mechanics and rather treating motion as a statistical pattern to learn given multiple sensorimotor readings (Bekris et al., 2024). Between these two extrema, a variety of methods

exists, with many borrowing traits from the opposite extremum to tackle specific challenges. On the one hand, as learning-based systems can benefit from information relative to the physics of particularly static tasks, Temporal Difference (TD) methods have been complemented with Model-Predictive Control (MPC) (Hansen et al., 2022). On the other hand, explicit models may be relying on assumptions proving overly simplistic—or even unrealistic—in practice, as in the case of assuming perfect observability of a robot state using sensorimotor readings. In this context, feedback loops at the control level help mitigate the effects of poor state estimation, complementing the motion planning process with discrepancy-from-target information, similarly to how loss functions are used in statistical learning. Figure ?? graphically illustrates the most relevant techniques developed for general-purpose applications. This categorization is far from being exhaustive, and we refer the interested reader to Bekris et al. (2024) for a much more comprehensive overview of both general and application-specific methods for motion generation. In this section, we aim at introducing the inherent benefit of learning-based approaches to robotics—the core focus on this tutorial—in the context of increasing availability of robot-data

1.2 Different kinds of motion

At its core, robotics deals with producing motion via actuating joints connecting nearly entirely-rigid links. A key distinction between focus areas in robotics is based on whether the generated motion modifies (1) the relative state of the robot with respect to its environment, (2) the absolute state of the environment or (3) both relative and absolute state (Figure ??). For instance, (1) may consist in changes in the robot’s physical location within its environment. Generally, modifications to a robot’s location within its environment may be considered instances of the general *locomotion* problem, further specified as *wheeled* or *legged* locomotion based on whenever a robot makes use of wheels or leg(s) to move in the environment. Further, (2) are typically achieved *through* the robot, i.e. generating motion to cause a perform an action inducing a desirable modification, effectively manipulating (*manipulation*) the environment. Manipulation is a well studied problem class in robotics, nowadays solved in practice for static and repetitive manipulation tasks, typically performed in controlled and precisely studied scenarios for instance via industrial robots employed at various stages of manufacturing processes. Lastly, an increased level of dynamism in the robot-environment interactions can be obtained combining (1) and (2), thus designing systems capable to move within *and* interact with their environment, falling under the category of *whole-body control*, and is characterized by a typically much larger set of control variables compared to either locomotion or manipulation alone.

The traditional body of work developed since the very inception of robotics is increasingly more complemented by learning-based approaches. ML has indeed proven particularly transformative across the entire robotics stack, first empowering planning-based techniques with improved state estimation used for traditional planning (Tang et al., 2023) and then end-to-end replacing controllers, from perception to action (Kober et al.). Work in producing robots capable of navigating a diverse set of terrains demonstrated the premise of both dynamics and learning-based approaches for locomotion (Griffin et al., 2017; Ji et al., 2023; Lee et al., 2020; Margolis et al., 2022), and recent works on whole-body control indicated the premise of learning-based approaches to generate rich motion in complex robot including humanoids (Zhang et al., 2024; NVIDIA et al., 2025). Manipulation has also been widely studied, particularly considering its aforementioned relevance for many impactful applications ranging from high-risk applications for humans (Fujita et al., 2020; Alizadeh and Zhu, 2024; Fujita et al., 2020) to manufacturing (Sanneman et al., 2020). While explicit models relying on precise descriptions of the dynamics and knowledge of the robot-environment system have been the key milestones in the development of modern robotics, recent works leveraging learning-based techniques for manipulation proved particularly promising in surpassing scalability and practical applicability challenges (Kober et al.). Consequently, this work is primarily focused on the application of ML to robotics (*robot learning*), and is particularly *focused on manipulation*.

1.3 A Toy Example: (Planar) Manipulation

Robot manipulators typically consist of a series of links and joints, all connected to an *end-effector*. In this example, we wish to provide a concrete example of a pick-and-place task, a relatively simple manipulation task whereby a robot reaches, grasps, moves and releases an object in space through a gripper end-effector. Links and joints are considered responsible for generating motion, while the end effector is instead used to perform specific actions at the target location (e.g., grasping/releasing objects via closing/opening a gripper, using a specialized tool like a screwdriver or any measurement device).

Recently, the development of low-cost manipulators like the Aloha (Zhao et al., 2023) Aloha-2 (Aldaco et al.) and SO-100/SO-101 (Knight et al.) platforms significantly lowered the barrier to entry to robot learning research, considering their accessibility compared to more traditional platforms like the Franka or Panda robot ?. As a

consequence, works on robot learning increasingly tackle the problem of learning to perform highly dexterous tasks using said low-end platforms: a long-standing challenge in manipulation.

Deriving an intuition as per why learning-based approaches are gaining in popularity requires briefly analyzing traditional approaches for manipulation, leveraging forward/inverse kinematics and potentially control theory to account for disturbances and misestimations. Traditionally, manipulation combines insights from fields including spatial algebra, optimization theory, control theory, and more. Providing a detailed overview of these fields fall (well) beyond the scope of this tutorial, hence we refer the reader to works including Siciliano and Khatib (2016); Lynch and Park (2017); Tedrake (a,b) for a more comprehensive description of these methods. Here, we mostly wish to highlight the main reasons why the field of robotics is increasingly moving *away* from these legacy methods.

Let us consider the (simple) case where the SO-100 is restrained from actuating (1) the shoulder pane and (2) the wrist flex and roll motors. This effectively reduces the degrees of freedom of the SO-100 from the original 5+1 (5 actuators + end effector) to 2+1 (shoulder lift, elbow flex + gripper), yielding the planar manipulator robot presented in Figure ??, where spheres represent actuators yielding and lines links from the base of the stilified SO-100 to its end-effector (*ee*, gripper). Further, let us make the simplifying assumptions actuators can yield rotations in the links up to 2π radians with respect to a given frame of reference. In practice, this is seldom the case due to movement obstructions caused by the robot body. For instance, the shoulder lift cannot produce counter-clockwise movement due to the presence of the robot's base used to secure the SO-100 to its support and host the robot bus. These simplifying assumptions leave us with the planar manipulator of Figure ??, free of moving its end-effector by controlling the angles θ_1 and θ_2 , jointly referred to as *robot's configuration* and indicated with $q = [\theta_1, \theta_2] \in [0, 2\pi]^2$. In this tutorial, we do not cover spatial algebra and frames of reference, and we instead refer the interested reader to (?, Chapter 2) and (Tedrake, a, Chapter 3) for excellent explanations of mechanics and the theoretical foundations of producing motion.

Considering the (toy) example presented in Figure ??, then we can analytically write the end-effector's position $p \in \mathbb{R}^2$ as a function of the control applied, $p = p(q)$, $p : \mathcal{Q} \mapsto \mathbb{R}^2$. In particular, we have:

$$p(q) = \begin{pmatrix} p_x(\theta_1, \theta_2) \\ p_y(\theta_1, \theta_2) \end{pmatrix} = \begin{pmatrix} l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \\ l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \end{pmatrix}$$

Deriving the end-effector's *pose* $p \in \mathcal{P} \subset \mathbb{R}^{\frac{n(n+1)}{2}}$ (position and orientation) starting from the configuration $q \in \mathcal{Q} \subset \mathcal{R}^n$ of a n -joints robot is a problem referred to as *forward kinematics* (FK), whereas identifying the configuration that corresponds to a given target pose is termed *inverse kinematics* (IK). In that, FK is used to map a robot configuration into end-effector position, whereas IK is used to derive what configuration(s) results in a given end-effector pose. In the simplified case here considered, one can solve the problem of controlling the end-effector's location to reach a goal position p^* by solving analytically for $q : p(q) = f_{\text{FK}}(q) = p^*$. However, in the general case, one might not be able to solve this problem analytically, and can typically resort to iterative optimization methods comparing candidate solutions based using a loss function (in the simplest case, $\|p(q) - p^*\|_2^2$ is a natural candidate), yielding:

$$\min_{q \in \mathcal{Q}} \|p(q) - p^*\|_2^2 \quad (1)$$

Analytical solutions for FK are even less appealing when one considers the presence of obstacles in the robot's workspace, resulting in constraints on the possible values of $q \in \mathcal{Q} \subset [-\pi, +\pi]^n \subset \mathbb{R}^n$ in the general case of n -links robots. For instance, the robot in Figure ?? is (very naturally) obstructed by the presence of the surface upon which it rests: θ_1 can now exclusively vary within $[-\frac{\pi}{2}, \frac{\pi}{2}]$, while possible variations in θ_2 depend on θ_1 (when $\theta_1 \rightarrow \pm\frac{\pi}{2}$, further downwards movements are restricted). Even for a simplified kinematic model such as the one considered here, solving 1 is in general non-trivial, particularly considering an arbitrary \mathcal{Q} itself may change across problems. Figure ?? provides another such example of how analytical solutions might fail to prove effective, even in our extremely simplified settings—if an obstacle was to be positioned in the robot environment, a new set of constraints based on the position of this new obstacle would emerge, potentially resulting in unfeasibility of the problem.

Further, IK—selecting the q solving 1—only proves useful in determining information regarding to the robot's configuration in the goal configuration, and crucially does not provide information on the *trajectory* to follow over time to reach a target pose. *Differential* inverse kinematics (diff-IK) complements IK by enabling trajectory tracking. Let $J(q)$ denote the Jacobian matrix of (partial) derivatives of the FK-function $f_{\text{FK}} : \mathcal{Q} \mapsto \mathcal{P}$, $J(q) = \frac{\partial f_{\text{FK}}(q)}{\partial q}$, then one can

apply the chain rule to any $p(q) = f_{\text{FK}}(q)$ deriving $\dot{p} = J(q)\dot{q}$, finally relating variations in the robot configurations to variations in positions, thus providing a platform to for control.

Thus, given a desired end-effector velocity trajectory \dot{p}^* indicating (1) which points in space to traverse and (2) how much time to spend in each region, diff-IK finds $\dot{q}(t)$ by solving a λ -regularized version of 1, solving for *velocities* instead of *positions*,

$$\dot{q}(t) = \arg \min_{\nu} \|J(q(t))\nu - \dot{p}^*(t)\|_2^2 \quad (2)$$

Unlike control in position, \dot{q} is much less dependent on the environment (typically, variations in velocity are constrained by physical limits on the actuators represented with box-like constraints in 2. Conveniently, 2 admits the closed-form solution, $\dot{q} = J(q)^+ \dot{p}^*$, where $J^+(q)$ denotes the Moore-Penrose pseudo-inverse of $J(q)$. Finally, discrete-time trajectories can then obtained using forward integration,

$$q_{t+1} = q_t + \Delta t \dot{q}_t$$

Following trajectories with diff-IK is a valid option in well-controlled and static environments like manipulators in industrial applications, and relies on the ability to define a set of target velocities to track $[p_0^*, p_1^*, \dots, p_k^*]$ —a task largely carried out by human experts and prone to errors. Furthermore, diff-IK relies on the ability to (1) access $J(q) \forall q \in \mathcal{Q}$ and (2) compute its pseudo-inverse at every iteration of a given control cycle—an assumption complicated to guarantee in highly dynamical settings or for complex instances of FK (nonlinear dynamics).

1.3.1 Adding feedback loops

While effective when a goal has been well specified, the performance of diff-IK can degrade significantly in the presence of modeling/tracking errors or in the presence of non-modeled dynamics in the environment. One such case is presented in Figure ??, where another rigid body is moving in the environment with velocity \dot{x}_B . Accounting analytically for the presence of this disturbance—for instance, to prevent l_1 from ever colliding with the object—requires access to \dot{x}_B and of a reference position $x_{B,0}$ for the object B , to derive the equation characterizing its motion. For less predictable disturbances (e.g., $\dot{x}_B \leftarrow x_B + \varepsilon, \varepsilon \sim N(0, 1)$) one could attain similar results by adding a condition on the distance between the midpoint of l_1 and x_B , enforced through a feedback loop on the position of the robot and object at each control cycle.

To mitigate the effect of modeling errors, sensing noise and other disturbances, classical pipelines augment diff-IK with feedback control on the measured error between the target position and the one measured, $\Delta p = p^* - p(q)$, hereby modifying the control applied to $\dot{q} = J(q)^+ (\dot{p}^* + \lambda \Delta p)$, with λ defined as the gain. More advanced techniques for control consisting in feedback linearization, PID control, or LQR/MPC can also be employed to stabilize tracking and reject moderate perturbations ((Siciliano and Khatib, 2016, Chapter 8), or (Tedrake, a, Chapter 8) for a simplified example in the case of a point-mass system). Nonetheless, tuning gains remains laborious and system-specific: In manipulation, intermittent contacts induce hybrid dynamics (mode switches) and discontinuities in the effective Jacobian, challenging stability guarantees and often necessitating conservative gains and substantial hand-tuning.

We point the interested reader to (Siciliano and Khatib, 2016, Chapter 2,7,8), (Lynch and Park, 2017, Chapter 6,11), and (Tedrake, a, Chapter 3,8) for extended coverage of FK, IK and diff-IK and control for (diff-)IK.

1.4 Limitations

Despite the last 60+ years of robotics research, autonomous robots are still largely incapable of performing tasks at human-level performance in the physical world across (1) robot embodiments (different manipulators, different locomotion platforms, etc.) and (2) tasks (tying shoe-laces, manipulating a diverse set of objects). While essential in the early development of robotics, the aforementioned methods require significant human expertise to be used in practice, and are typically specific to a particular applicative problem. We argue this follows from various reasons.

Robotics pipelines have been **developed sequentially, hand-engineering the different block architectures** for specific purposes. Sensing, state estimation, mapping, planning, (diff-)IK, and low-level control have been traditionally developed as distinct modules with fixed interfaces. Pipelining these highly specific modules is a error-prone, and brittleness emerges whenever shifts with respect to a considered pipeline incur (e.g., changes in lighting for sensing, occlusion/failure of sensors). Adapting such a stack to new tasks or embodiments often entails re-specifying objectives, constraints, and heuristics at multiple stages, incurring significant engineering overhead.

Classical planners operate on compact, assumed-sufficient state representations; extending them to reason directly over raw, heterogeneous and noisy data streams is non-trivial. This results in a **limited scalability to multimodal data and multitask settings**, as incorporating high-dimensional perceptual inputs (RGB, depth, tactile, audio) traditionally required extensive engineering efforts to extract meaningful efforts for control. Moreover, the large number of tasks, coupled with the adoption of *per-task* planners, goal parameterizations, and safety constraints, results in an explosion in design and validation effort, with little opportunity to share experience across tasks.

Further, even setting aside integration and scalability challenges, developing accurate modeling of contact, friction, and compliance for complicated remains difficult. Rigid-body approximations are often insufficient in the presence of deformable objects, and **relying on approximated models hinders real-world applicability**. In the case of complex and/or non-linear dynamics, even moderate mismatches in contact parameters, unmodeled flexibilities, or grasp-induced couplings can qualitatively affect the observed dynamics, degrading the reliability of the model obtained. Even a perfect model of reality would need to relax the stationarity assumptions that are typically made in control. Explicitly modeling the effects of payload changes, tool substitutions, temperature-dependent friction, and component-aging on the resulting dynamics is, in most cases, simply unfeasible.

Lastly, dynamics-based methods (naturally) overlook the rather recent phenomenon of the **explosion of openly-available robotics data**, following a similar pattern to the explosion of image and text data. The curation of academic datasets by large centralized groups of human experts in robotics (Ope; DRO; AgiBot-World-Contributors et al., 2025) is now increasingly complemented by a **growing number of robotics datasets contributed in a decentralized fashion** by individuals with varying expertises (Figure ??). If not tangentially, dynamics-based approaches are not posed to maximally benefit from this new trend, holding the premise of allowing generalization in the space of tasks and embodiments, like data was the cornerstone for advancements in vision (Alayrac et al., 2022) and natural-language understanding (OpenAI et al., 2024).

Taken together, these limitations motivate the exploration of learning-based approaches that can (1) integrate perception and control more tightly, (2) adapt across tasks and embodiments with reduced expert intervention and (3) scale gracefully in performance as more robotics data becomes available.

References

- DROID: A Large-Scale In-the-Wild Robot Manipulation Dataset. <https://droid-dataset.github.io/>.
- Open X-Embodiment: Robotic Learning Datasets and RT-X Models. <https://robotics-transformer-x.github.io/>.
- AgiBot-World-Contributors, Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, Shu Jiang, Yuxin Jiang, Cheng Jing, Hongyang Li, Jialu Li, Chiming Liu, Yi Liu, Yuxiang Lu, Jianlan Luo, Ping Luo, Yao Mu, Yuehan Niu, Yixuan Pan, Jiangmiao Pang, Yu Qiao, Guanghui Ren, Cheng Ruan, Jiaqi Shan, Yongjian Shen, Chengshi Shi, Mingkan Shi, Modi Shi, Chonghao Sima, Jianheng Song, Huijie Wang, Wenhao Wang, Dafeng Wei, Chengen Xie, Guo Xu, Junchi Yan, Cunbiao Yang, Lei Yang, Shukai Yang, Maoqing Yao, Jia Zeng, Chi Zhang, Qinglin Zhang, Bin Zhao, Chengyue Zhao, Jiaqi Zhao, and Jianchao Zhu. AgiBot World Colosseo: A Large-scale Manipulation Platform for Scalable and Intelligent Embodied Systems, August 2025.
- Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: A Visual Language Model for Few-Shot Learning, November 2022.
- Jorge Aldaco, Travis Armstrong, Robert Baruch, Jeff Bingham, Sanky Chan, Debidatta Dwibedi, Chelsea Finn, Pete Florence, Spencer Goodrich, Wayne Gramlich, Alexander Herzog, Jonathan Hoech, Thinh Nguyen, Ian Storz, Baruch Tabanpour, Jonathan Tompson, Ayzaan Wahid, Ted Wahrenburg, Sichun Xu, Sergey Yaroshenko, and Tony Z Zhao. ALOHA 2: An Enhanced Low-Cost Hardware for Bimanual Teleoperation.
- Mohammad Alizadeh and Zheng H. Zhu. A comprehensive survey of space robotic manipulators for on-orbit servicing. *Frontiers in Robotics and AI*, 11, October 2024. ISSN 2296-9144. doi: 10.3389/frobt.2024.1470950.
- Kostas E. Bekris, Joe Doerr, Patrick Meng, and Sumanth Tangirala. The State of Robot Motion Generation, October 2024.
- Jun Fujita, Daisuke Soda, Chotaro Murata, and Hiroyuki Tsuchi. Development of Robots for Nuclear Power Plants and Their Application to New Fields. 57(4), 2020.
- Robert J. Griffin, Georg Wiedebach, Sylvain Bertrand, Alexander Leonessa, and Jerry Pratt. Walking Stabilization Using

- Step Timing and Location Adjustment on the Humanoid Robot, Atlas. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 667–673, September 2017. doi: 10.1109/IROS.2017.8202223.
- Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal Difference Learning for Model Predictive Control, July 2022.
- Yandong Ji, Gabriel B. Margolis, and Pulkit Agrawal. DribbleBot: Dynamic Legged Manipulation in the Wild, April 2023.
- Rob Knight, Pepijn Kooijmans, Thomas Wolf, Simon Alibert, Michel Aractingi, Dana Aubakirova, Adil Zouitine, Russi Martino, Steven Palma, Caroline Pascal, and Remi Cadene. Standard Open SO-100 & SO-101 Arms.
- Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement Learning in Robotics: A Survey.
- Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning Quadrupedal Locomotion over Challenging Terrain. *Science Robotics*, 5(47):eabc5986, October 2020. ISSN 2470-9476. doi: 10.1126/scirobotics.abc5986.
- Kevin M. Lynch and Frank C. Park. *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, 1 edition, May 2017. ISBN 978-1-316-66123-9 978-1-107-15630-2 978-1-316-60984-2. doi: 10.1017/9781316661239.
- Gabriel B. Margolis, Ge Yang, Kartik Paigwar, Tao Chen, and Pulkit Agrawal. Rapid Locomotion via Reinforcement Learning, May 2022.
- NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An Open Foundation Model for Generalist Humanoid Robots, March 2025.
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altmenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kopic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O’Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, C. J. Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. GPT-4 Technical Report, March 2024.
- Lindsay Sanneman, Christopher Fourie, and Julie A. Shah. The State of Industrial Robotics: Emerging Technologies, Challenges, and Key Research Directions, October 2020.

- Bruno Siciliano and Oussama Khatib, editors. *Springer Handbook of Robotics*. Springer Handbooks. Springer International Publishing, Cham, 2016. ISBN 978-3-319-32550-7 978-3-319-32552-1. doi: 10.1007/978-3-319-32552-1.
- Yang Tang, Chaoqiang Zhao, Jianrui Wang, Chongzhen Zhang, Qiyu Sun, Weixing Zheng, Wenli Du, Feng Qian, and Juergen Kurths. Perception and Navigation in Autonomous Systems in the Era of Learning: A Survey. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12):9604–9624, December 2023. ISSN 2162-237X, 2162-2388. doi: 10.1109/TNNLS.2022.3167688.
- Russ Tedrake. Robotic Manipulation. Perception, Planning and Control., a.
- Russ Tedrake. Underactuated Robotics. Algorithms for Walking, Running, Swimming, Flying, and Manipulation, b.
- Chong Zhang, Wenli Xiao, Tairan He, and Guanya Shi. WoCoCo: Learning Whole-Body Humanoid Control with Sequential Contacts, November 2024.
- Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware, April 2023.