Approaches to generating arm movements in humanoid robot NICO[*]

Igor Farkaš, Andrej Lúčny, Michal Vavrečka

Faculty of Mathematics, Physics and Informatics, Comenius University Bratislava
Mlynská dolina F1, 84248 Bratislava
{lucny,farkas}@fmph.uniba.sk, michal.vavrecka@cvut.cz

Abstract

The generation of arm movements is a fundamental executive capability associated with robots interacting with an environment. Despite easiness of motor behavior in humans, generating a desired movement in humanoid robots is difficult depending on a number of factors, including the hardware and the task specification. Here we briefly describe various approaches to movement generation with their pros and cons. Generation of smooth and legible movement is often a requirement, especially in case of humanoid robots interacting with humans, which in the future are expected to play an important role in modern society.

1 Introduction

Motor movement is the basic functionality of robots, enabling them to act in the environment. There are two basic categories of motor movement: (1) navigation of a mobile robot in space and (2) object manipulation using robotic arms, affecting the state of the environment.

In our research, we are working with the semi-humanoid robot NICO (Kerzel et al., 2017), which is required to perform movements with the right arm according to the assigned tasks in the context of human–robot interaction (HRI, see Fig.1). The goal of the task is to ensure that the robot performs *legible* movements, i.e. those that better reveal robot *intention*. The legibility of motion is a concept developed in the literature (Stulp et al., 2015). It can be considered as one of the prerequisites for *trustworthy* HRI, which is a relatively new field of research (Kok and Soh, 2020). In this context, humanoid robots represent the most suitable option for successful HRI, mostly because of human tendencies to antropomorphise the robots (Vernon and Sandini, 2024).

2 Approaches to generating arm movements

There exist various approaches toward generation of robotic arm movement, ranging from very rigid engineering ones to the most flexible, machine learning based methods. Typically generating a precise arm movement requires the knowledge of joint values for all degrees of freedom, such that the *forward control* could be applied (i.e. setting the joints to the required angle values). In principle, one can either try to solve the problem directly with the physical NICO[1], or take an advantage of a robotic simulator, optionally combined with *sim-to-real transfer* (if one needs to deploy the functionality to the physical robot). Here we briefly summarize the available approaches.

Fig. 1: Semihumanoid robot NICO (left), in the human–robot interaction setup (right).

2.1 Robot programming by demonstration

In this ecologically invalid approach (children do not learn that way), the robotic arm is held by a person who is trying to execute a concrete desired trajectory while the intermediate values of all joints are being stored. The arm must be in a compliant mode to enable easy manipulation. Due to gravity, however, the trajectories recorded during demonstration, will not be the same as those executed in a self-execution mode. The degree of inaccuracy is also a matter of hardware, which in case of NICO is not very robust, as we discovered.

2.2 Movement based on human motion tracking

In this approach, we first record the human arm movement using an appropriate device, such as an RGB camera or a motion tracker (e.g. Kinect). Extracted arm skeleton is then converted to the robot's frame and executed in an offline mode. One can also use motion track-

[*]supported by projects VEGA 1/0373/23 a APVV-21-0105

[1]https://github.com/andylucny/nico

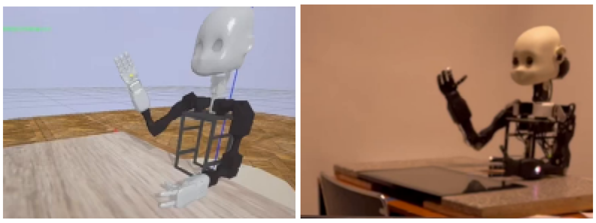

Fig. 2: Simulated NICO in myGym environment (left), and its physical counterpart (right).

ing for online teleoperation of the robotic arm. The observed inaccuracies may be due to differences in the geometry between the demonstrator's arm and the robotic arm.

2.3 Using the robotic simulator

Using a robotic simulator offers a number of advantages compared to the physical robot (Choi et al., 2021). In the simulator one needs to use a very accurate robotic model in order to enable sim-to-real transfer (Fig. 2). We developed an accurate NICO model (in URDF format) in myGym environment[2], an easy-to-use toolkit suitable for fast prototyping of neural networks for robotic manipulation (Vavrečka et al., 2021).

Using the simulator, we first generated arm movement to the desired position of the index finger in Cartesian 3D space using *inverse kinematics* (IK) module (being part of Bullet physics engine) that finds the best corresponding joint angles. There are several options how to use the IK module in the sense how many times to call that module (a single call with a large step vs a number of calls with small steps). This approach is very safe and flexible, yet requires parameter fine-tuning due to differences in motion execution between the simulator and the physical servomotors.

2.4 Reinforcement learning

Machine learning has proven a successful approach to solving many computational tasks, including the robotics field. Reinforcement learning (Sutton and Barto, 2018) is a type of machine learning enabling the agent acting in the environment to learn an optimal policy (i.e. knowledge which optimal action to take in each step). This is guided by the reward signals, designed by the experimenter, to be maximized in the long term. Due to long training times, policy learning requires the use of robotic simulators. Reward shaping is a very powerful mechanism enabling to shape the learned behavior according to task requirements, but it has some limitations. Learned policy can be run in the simulator, while being almost simultaneously executed in the real robot (sim-to-real transfer) (Zhao et al., 2020).

[2]https://github.com/incognite-lab/myGym/tree/nico-sim2real

3 Importance of implementation level for robotic movement

On an algorithmic level, the motor learning can be understood as changing the joint values of all degrees of freedom in discrete time steps. However, the concrete execution of motor commands depends on the used robotic simulator, its physics engine, but also on the properties of the servomotors controlling the robot movement. In sim-to-real transfer it is necessary to accurately transfer the commands and make sure they are executed properly. What matters here is the communication between the software and the motors. The goal is often to make the robot movements smooth, legible and safe. Fine tuning is hard to automate, so it often requires trial and error. In summary, generating robotic arm movements with desired properties also requires a solid knowledge of the robot at the implementational level.

References

Choi, H. et al. (2021). On the use of simulation in robotics: Opportunities, challenges, and suggestions for moving forward. In *Proceedings of National Academy of Sciences of the USA*, volume 118.

Kerzel, M., Strahl, E., Magg, S., Navarro-Guerrero, N., Heinrich, S., and Wermter, S. (2017). NICO - Neuro-Inspired COmpanion: A developmental humanoid robot platform for multimodal interaction. In *Proceedings of the IEEE Int. Symposium on Robot and Human Interactive Communication*, pages 113–120.

Kok, B. and Soh, H. (2020). Trust in robots: Challenges and opportunities. *Current Robotics Reports*, 1:297–309.

Stulp, F., Grizou, J., Busch, B., and Lopes, M. (2015). Facilitating intention prediction for humans by optimizing robot motions. In *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1249–1255.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT Press, 2 edition.

Vavrečka, M., Sokovnin, N., Mejdrechová, M., and Šejnová, G. (2021). MyGym: Modular toolkit for visuomotor robotic tasks. In *IEEE Int. Conf. on Tools with Artificial Intelligence (ICTAI)*, pages 279–283.

Vernon, D. and Sandini, G. (2024). The importance of being humanoid. *International Journal of Humanoid Robotics*, 21(1).

Zhao, W., Queralta, J. P., and Westerlund, T. (2020). Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *IEEE Symposium Series on Computational Intelligence*, pages 737–744.