Cognitive Robotics Course - From Electronics to Behavior

Simon Untergasser and Manfred Hild

Berliner Hochschule für Technik, Berlin, Germany,
`simon.untergasser@bht-berlin.de`,
WWW home page: `http://neurorobotik.de`

Abstract. This paper presents an innovative undergraduate course framework in cognitive robotics that combines hands-on engineering experience with theoretical foundations in sensorimotor learning. The "Cognitive Robotics Project" course, offered in the third semester of a Humanoid Robotics program, enables students to design, construct, and program their own robots while understanding fundamental concepts of embodied cognition and sensorimotor exploration. The course consists of three parts: Hardware, Software, and Experiments. First, students solder custom-designed circuit boards and create 3D-printed robot designs. Then they implement sensorimotor behaviors using C programming. Finally, they conduct extensive experiments with their own hardware and software. The course's unique approach not only teaches practical skills but also develops comprehensive understanding of the complete robotics development cycle. The course structure, technical framework, and educational outcomes are presented, demonstrating insights from the development and initial implementation of this course, including student-created robot designs.

Keywords: Educational robotics · Sensorimotor learning · Project-based learning · Embodied cognition · Undergraduate education

1 Introduction

The field of robotics continues to gain traction across various societal applications, from healthcare to service or personal assistance. As this domain expands, the education of future robotics researchers and engineers becomes increasingly critical, particularly in understanding fundamental concepts such as sensorimotor coupling - the intricate relationship between sensory inputs and motor activities in both natural and artificial systems. However, teaching these complex relationships presents significant challenges in higher education, especially when dealing with sophisticated hardware and diverse student capabilities.

In robotics research, sensorimotor coupling represents a fundamental aspect where sensors provide environmental and proprioceptive information that must be processed and translated into meaningful behavior. This ongoing process, known as the sensorimotor loop, is crucial for robots to interact effectively in complex, unstructured environments. While implementations can range from simple Braitenberg-style wirings [Braitenberg, 1986] to sophisticated neural networks [Heinrich et al., 2020], the field of sensorimotor coupling research is still evolving ([Hild et al., 2021], [Sándor et al., 2017]), making it essential to incorporate these concepts into educational frameworks early in students' academic careers (see [Untergasser et al., 2022b], [Untergasser et al., 2022a]).

Traditional approaches to teaching humanoid robotics often rely on complex, expensive platforms such as Nao robots [Gouaillier et al., 2009] or iCub systems [Metta et al., 2008]. While these platforms offer comprehensive capabilities, their limited access to the underlying architecture, high maintenance requirements, and complexity can create significant barriers to effective education. Furthermore, future robotics researchers and engineers should not only be able to use commercial products, but to build them themselves. Additionally, the varying learning needs and performance levels of students present significant challenges in delivering complex technical content while maintaining engagement and preventing frustration across diverse skill levels.

To address these challenges, we present an innovative undergraduate course framework that combines hands-on engineering experience with theoretical foundations in cognitive robotics. The "Cognitive Robotics Project" course, implemented in the third semester of our Humanoid Robotics program, offers a unique approach where students engage with the material through direct experience - from soldering custom-designed circuit boards to programming sensorimotor behaviors. This approach allows students to explore sensorimotor principles interactively and gaining insights into current research topics.
Our framework addresses several key challenges in robotics education:

- The accessibility of hardware through custom-designed, cost-effective components
- The integration of theoretical concepts with practical experience
- The accommodation of diverse student skill levels through scalable project complexity
- The connection between undergraduate education and current research endeavors

This paper presents a comprehensive analysis of our educational framework, demonstrating how it bridges the gap between theoretical understanding and practical implementation in an educational setting. We show how this approach not only enhances student learning outcomes but also creates a sustainable model for integrating theoretical knowledge with hands-on experience in cognitive robotics.

The cognitive component of this framework is realized through the exploration of sensorimotor manifolds, where students analyze the relationship between motor actions and resulting body configurations. This embodied cognition approach allows students to understand how physical interactions shape perception and behavior in robotic systems. By creating their own robots and observing emergent behaviors, students gain firsthand experience with fundamental concepts of cognitive robotics: the integration of sensing, action, and environmental feedback in creating adaptive behavior.

The remainder of this paper is structured as follows: Section 2 describes the course framework and methodology, followed by technical implementation details in Section 3. Section 4 examines outcomes, presents and discusses the results. Finally, Section 5 concludes with an evaluation and future work.

2 Course Framework and Methodology

In this section we introduce the course framework, its content and the motivation behind it. We start by laying out the learning goals and a brief theoretical background. Then we present the structure of the course with its three different phases. We end the section with a description of monitoring and supporting tools and a brief description of the submissions, the student teams have to hand in.

2.1 Goals of the course

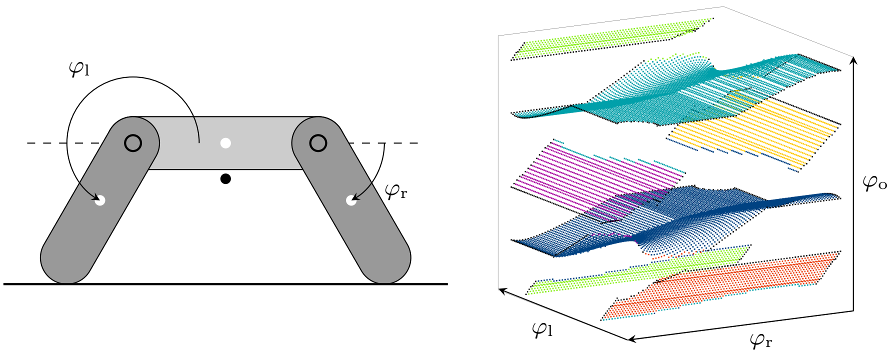
The topic of interest for this course are small robots with a body and one or two legs actuated with two servo motors. These robots should be programmed such that they explore their possible movements and body configurations. The reachable body configurations can be represented with a sensorimotor manifold by recording motor angles and body angles (in respect to the ground). A sensorimotor manifold represents the relationship between a robot's motor actions (input) and resulting body positions (output) - essentially a "map" of all physically attainable configurations. This concept is fundamental to embodied cognition, as it shows how physical form constrains and shapes possible behaviors.

A body configuration is a tuple $(\varphi_l, \varphi_r, \varphi_o)$, which are the angle of the left motor, the angle of the right motor and the body orientation respectively. Then sensorimotor manifold is the subset of all possible tuples, which are possible given a certain robot morphology. In 3D space, φ_o would be a 3 element vector defining the orientation of the robot. A suggestion (but not a requirement) for the students is, to place the two motors with parallel axes to each other, such that the robot moves in a 2D plane. Therefore φ_o reduces to one angle. Such a manifold can be easily visualized as can be seen in figure 1.

Here on the left side we have a simple toy robot consisting of a body (light gray) and two actuated legs (darker gray). The robot is fully symmetrical, meaning that one cannot differentiate top, bottom, left or right just by the morphology of the robot. The right side shows the sensorimotor manifold, where the colored points represent body configurations, the robot can reach.

The goals of the course can be formulated like this:

- Design and build a robot with an interesting morphology
- Program the robot such that it builds up and saves its sensorimotor manifold
- Conduct experiments by varying parameters like: robot morphology, environment, control laws



(a) (b)

Fig. 1: An example for a robot in (a) (the robot *Tablebot* from [Rist and Hild, 2022]) and its sensorimotor manifold in (b).

2.2 Structure of the course

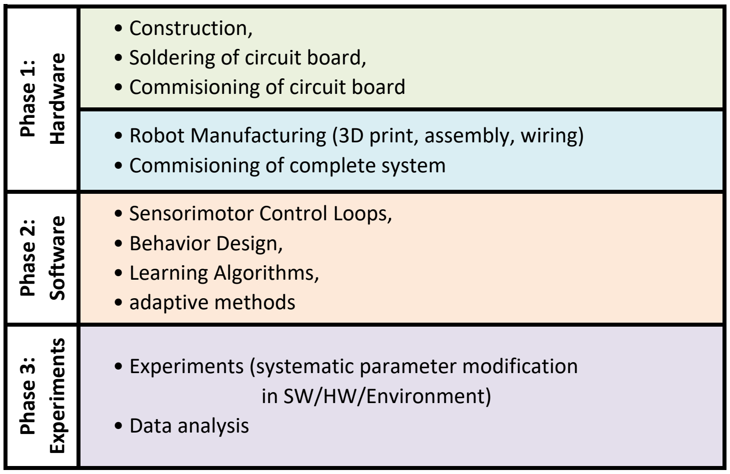
The course is divided into three phases that students work on in teams of two. Figure 2 shows an overview of the course structure and contents.

The course follows a dual-track approach combining practical project work with synchronized lectures. While students progress through the three project phases, the accompanying lectures provide theoretical foundations and technical knowledge exactly when needed for each phase, ensuring concepts can be immediately applied to their robots.

The first phase "Hardware" consists of 6 weeks, where students develop, manufacture and assemble their own robots. From day one, students start with iterating on ideas for their robot morphology and work on the construction. Teams begin by sketching ideas on paper and thinking about possible behaviors of their robot design. When they settle on a design idea, they start constructing. We built up a process of early feedback, so that by week 4, all students have received at least two rounds of feedback on their constructions and are ready to print and assemble their robots. In parallel we created a pipeline for SMD (surface mounted device) soldering. All groups soldered their own circuit boards (see section 3 for details). After phase 1 all teams have a working robot.

The second phase "Software" is 5 weeks long. Here the programming happens, using the C programming language. The teams get a template where the basic structure of the program is already prepared and the firmware is provided (for more details, see next section). Now the students learn how to implement sensorimotor control (the basics of which they learn in the second semester) and behavior design. The goal of phase 2 is an algorithm, which allows the robot to build up its complete sensorimotor manifold when switched on.

The last phase "Experiments" is 5 weeks long as well. We specifically aimed for a long experimental phase. First it can serve as a buffer for the other phases.



Phase 1: Hardware	• Construction, • Soldering of circuit board, • Commisioning of circuit board
	• Robot Manufacturing (3D print, assembly, wiring) • Commisioning of complete system
Phase 2: Software	• Sensorimotor Control Loops, • Behavior Design, • Learning Algorithms, • adaptive methods
Phase 3: Experiments	• Experiments (systematic parameter modification in SW/HW/Environment) • Data analysis

Fig. 2: Overview of the course structure

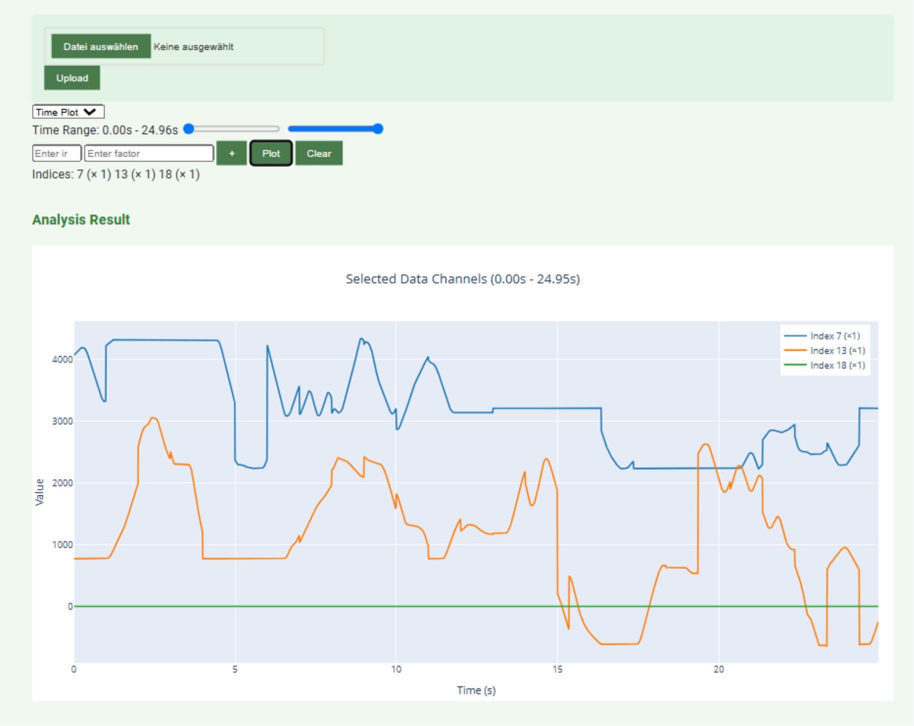
Second and mainly, we wanted to reduce time pressure for the experiments, since this phase is often overlooked and too short. At the beginning of phase 1, the teams are asked to think about possible experiments for phase 3, so that they have a long lasting and clear plan to follow.

2.3 Progress tracking and evaluation

To help the students to track, review and evaluate their own progress and work, we created a website. The website has two parts, the projects page and the data analysis pages. On the projects page, all teams have a section, where they describe their work in diary form. They can upload images and other relevant files and then update the html of their section. All teams have access to all sections, such that they can get inspiration from other teams, help each other and provide feedback, but they can only alter their section. The second part is for data visualization and inspection. There are several modes, which can be used here. One is, to plot all necessary data, like IMU (Inertial Measurement Unit) data and all data coming from the motors (e.g. Angle, Velocity, Current). In the second mode, one can choose which data should be plotted. That is for inspecting specific data more closely or data which the students defined in their C code (e.g. body angle). Figure 3 shows a plot of some data on the data analysis website.

The evaluation for our course consists of three distinct results, the student teams have to hand in. These are:

- A pdf with concept description, project plan and experiment design
- Program code, initial data recordings and plots of their sensorimotor manifolds



Time Range: 0.00s - 24.96s

Indices: 7 (× 1) 13 (× 1) 18 (× 1)

Analysis Result

Fig. 3: The figure shows the data analysis website.

- A video with a live demo of their robot showing an example behavior. They should also show an explanation of their system using slides or a whiteboard

3 Technical Implementation

This section describes the technical details of the hardware and software we prepared and used in the course. We start with the "brain" of the robots, the circuit board, which the students soldered in the first phase. We then continue with the design constraints for the robots including some example robots. The section ends with the software framework.

3.1 Custom circuit board design and components

In preparation of the presented course, we designed a custom circuit board called "AccelBoard6D". The development took approximately 6 months, from the concept to the final circuit board, shown in figure 4 left and right, respectively. The board features a powerful STM32 processor, an IMU that measures acceleration in three dimensions and rotation in three dimensions, a SD card slot, a connector for Dynamixel XL330 actuators and various other components. The board shown in the figure is a minimal version without microphone and speaker, since for the course presented here, these components are not necessary.

The board features a STM32 Cortex-M33 microcontroller which provides sufficient processing power for the sensorimotor control algorithms. The integrated IMU (ST LSM6DSL) measures acceleration and rotation in three dimensions with a precision in the range of 0.001 g ($0.01^\circ\, s^{-1}$ respectively), allowing accurate determination of the robot's orientation. The servo motors (Dynamixel XL330) offer 0.52 Nm torque and 0.088° degree precision. These specifications were selected to balance cost, performance, and accessibility for an undergraduate educational setting. The board's firmware was developed in an outstanding bachelor thesis (see [Kramp, 2024]), including the IMU, SD-Card, Actuator, Microphone and Speaker integration. The electronic design and circuit board layout was done by the second author. After extensive tests we finalized the design and ordered electronic parts and circuit boards and prepared documentation for the soldering part of the course.

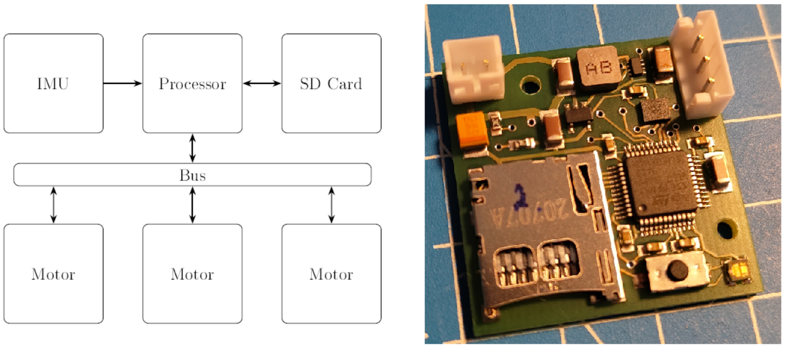


Fig. 4: The concept of the Accelboard6D on the left side and a completely soldered and tested board on the right.

3.2 Robot design guidelines and constraints

As mentioned in section 2.1, the robots can have a body with two actuated joints, with e.g. two legs. We suggested to have the axes of the motors parallel to each other, such that the robot moves in a 2D plane. This allows for easier data inspection, since the orientation of the body can be represented by one angle. The rest of the design we left open, for the teams to be creative. Of course there have been constraints regarding the construction, due to material properties, maximum torque of the motors and the like. We created a guideline to help the teams evaluate their constructions before printing. Some example points are listed below:

- Do not design too tightly: Allow for a tolerance of 0.1-0.3 mm
- Plan for cable routing between the circuit board and motors
- Do not forget any fixing points
- Avoid too small or fine structures
- Note and calculate all required forces → construct appropriately

3.3 Programming framework for sensorimotor exploration

We use the STM32Cube IDE [1] as software development framework, since the students already know it from another course. We provided a project template, with the necessary files and functions. The main routine, shown in listing 1.1, consists of an initialization and an infinite loop, as is common for robot applications. As can be seen, the code is rich in comments, such that the students can understand all details of it.

```c
int main(void) {
    // Set the LED to red
    set_LED(250,0,0);

    // Setting the IDs of both motors active (=1),
    // so that the firmware updates them
    SM.ID[1] = 1;
    SM.ID[2] = 1;

    // Activating saving to SD card and starting an empty file
    init_sd_saving();

    // Infinite loop of sensorimotor processing
    while (1) {
        // sync for 100Hz loop
        sync();

        // Update all sensor values and calculate control functions
        sensorimotor_update();

        // Update Behavior
        behavior_update();

        // save current batch of buffer data
        sd_update();
    }
```

Listing 1.1: The main routine with infinite update loop

The main parts, the students are working on, are the functions `sensorimotor_update` and `behavior_update`, which control the low level control functions and the high level behavior respectively. An example for a baseline `sensorimotor_update` function can be found in listing 1.2.

---

[1] https://www.st.com/en/development-tools/stm32cubeide.html

```
1  void sensorimotor_update(void) {
2      // This function reads sensor values and writes motor values
3      // Sensorimotor loops (control functions) are calculated
4  
5      // collect sensordata
6      get_SM();
7  
8      if (loops_active) {
9          // Motor 1 enable and constant velocity control function
10         set_constant_velocity(motor_1);
11         // Motor 2 enable and constant velocity control function
12         set_constant_velocity(motor_2);
13         set_LED(0,250,0);
14     }
15     else {
16         // Disable both motors
17         disable_motors();
18         set_LED(250,0,0);
19     }
20     debug_buffer[0] = imu_to_bodyangle();
21     debug_buffer[1] = current_energy();
22 
23     // write motor values
24     set_SM();
```

Listing 1.2: The sensorimotor_update function with constant velocity control

As can be seen here, there is a variable `loops_active`, which can be set by pressing the button on the circuit board. It enables the motors and starts the calculation of the control functions (here: constant velocity). Additionally, in lines 19 and 20, two functions developed together with the students in the accompanying lecture are used. In the first one `imu_to_bodyangle`, the IMU acceleration data is used to determine the angle of the robots body relative to the ground, using simple geometry. The second function `current_energy` calculates the electrical energy, the motors currently use. Both values are stored in a buffer, which is logged to the sd card, such that the students can inspect data using the data analysis tools provided on the website.

4 Outcomes, Results and Discussion

In this section we report preliminary results from our student teams. We have a total of 21 teams, all of which constructed and manufactured their own robot and soldered their circuit board successfully. It is the first project in the humanoid robotics program where the robot design is in most parts free for the creativity of the students. Therefore there are groups which handle that with ease, creating creative creations, while other teams struggled with a lack of ideas. We have provided an example robot that is helpful in some respects, but has also, as expected, paved the way for ideas in precisely this direction.

Fig. 5: Six different robots created in the course. The upper right shows the example robot from the authors, the other five are from student teams.

4.1 Some example projects

Figure 5 shows 6 robots created for this course. The robot in the upper right was created by our team as an example for the course and for preliminary tests. It has a very simple structure, but still offers a lot of flexibility with many possibilities to attach body shape altering parts. As can be seen, conceptually all robots are similar, since they have a body and two actuated legs on each side of the body. Still, there is enough diversity for interesting results.

4.2 Learning Outcomes and Challenges

The course yielded diverse outcomes across our student cohort. Of the 21 initial teams, 13 successfully completed all requirements with functional robots and satisfactory demonstrations. One team dropped out during the early phase, two teams failed to complete the final exercise, and five teams submitted videos that did not meet the evaluation criteria.

Analysis of student performance revealed specific technical and organizational challenges. The C programming component—particularly working with multiple C files having different purposes—proved significantly more complex than what students had previously encountered. Despite discussing various sensorimotor control concepts in lectures, students rarely implemented the more advanced approaches, as these would have required structured experimental validation beyond most teams' capabilities. Additionally, many students struggled with project organization and time management, particularly during the software phase.

The most successful project demonstrated the potential of our approach. One team created a fish-shaped robot and conducted extensive experiments with different weight distributions in the robot body. They tested their design on surfaces with varying friction coefficients, carefully analyzing the resulting sensorimotor manifolds. Their documentation included comprehensive and nicely crafted videos of their experiments, demonstrating how thoughtful experimentation yields interesting results.

The learning outcomes included:

- Applying sensorimotor loop concepts from second semester and using IMU-based body angle calculations for sensorimotor manifold analysis
- Development of technical skills across disciplines: electronics (SMD soldering), mechanical design (3D printing), and programming (embedded C)
- Acquisition of experimental methodology skills, including systematic parameter variation and data analysis

Based on these observations, future iterations of the course will incorporate more structured programming support through incremental milestones and enhanced project management guidance, particularly for the software phase.

5 Conclusion and Future Work

This paper has presented an educational framework that integrates three critical components of robotics education—hardware development, software implementation, and experimental validation—into a cohesive undergraduate course. Unlike traditional approaches that often focus predominantly on software or rely on commercial platforms, our methodology emphasizes the importance of students experiencing the complete development cycle of robotic systems. This holistic approach better prepares future robotics engineers for careers where they may need to develop rather than merely utilize existing systems.

The educational value of this framework extends beyond specific technical skills. By working through hardware constraints, software implementation challenges, and experimental validation processes, students develop cross-disciplinary thinking and systems-level problem-solving capabilities. The robot designs, though relatively simple, serve as effective vessels for teaching fundamental concepts in embodied cognition and sensorimotor exploration that can be applied to more complex systems later in students' careers.

While our initial implementation has shown promising results, we've identified key enhancements for future iterations:

- Allocating at least three full lab sessions (instead of two) for the SMD soldering phase, as most teams required 1-2 hours to complete their boards (initally planned were a maximum of 1 hour per group)
- Breaking down the C programming tasks into three milestone submissions to identify teams struggling with c programming concepts earlier
- Making weekly diary updates mandatory after three teams fell significantly behind without communicating their challenges

The integrative approach demonstrated in this course offers a valuable model for robotics education that balances theoretical foundations with practical implementation experience. Such an approach is increasingly important as robotics continues to expand into new application domains requiring comprehensive understanding of the interconnections between hardware, software, and real-world interactions.

References

[Braitenberg, 1986] Braitenberg, V. (1986). *Vehicles: Experiments in synthetic psychology.* MIT press.

[Gouaillier et al., 2009] Gouaillier, D., Hugel, V., Blazevic, P., Kilner, C., Monceaux, J., Lafourcade, P., Marnier, B., Serre, J., and Maisonnier, B. (2009). Mechatronic design of nao humanoid. In *2009 IEEE International Conference on Robotics and Automation*, pages 769–774. IEEE.

[Heinrich et al., 2020] Heinrich, S., Yao, Y., Hinz, T., Liu, Z., Hummel, T., Kerzel, M., Weber, C., and Wermter, S. (2020). Crossmodal language grounding in an embodied neurocognitive model. *Frontiers in Neurorobotics*, 14:52.

[Hild et al., 2021] Hild, M., Panreck, B., Untergasser, S., and Tolksdorf, M. (2021). An adaptive balancing robot with exceptional touch sensitivity driven by an analog sensorimotor loop.

[Kramp, 2024] Kramp, L. (2024). Development of a Processor-Platform for Humanoid Robots.

[Metta et al., 2008] Metta, G., Sandini, G., Vernon, D., Natale, L., and Nori, F. (2008). The icub humanoid robot: an open platform for research in embodied cognition. In *Proceedings of the 8th workshop on performance metrics for intelligent systems*, pages 50–56.

[Rist and Hild, 2022] Rist, V. and Hild, M. (2022). How to design morphologies. a design process for autonomous robots. In *International Conference on Simulation of Adaptive Behavior*, pages 3–14. Springer.

[Sándor et al., 2017] Sándor, B., Martin, L., and Gros, C. (2017). The role of the sensorimotor loop for cognition. *Cognitive Robot Architectures*, page 40.

[Untergasser et al., 2022a] Untergasser, S., Hild, M., and Panreck, B. (2022a). An educational framework for complex robotics projects. In *Robotics in Education: RiE 2021 12*, pages 94–104. Springer.

[Untergasser et al., 2022b] Untergasser, S., Hild, M., and Panreck, B. (2022b). Hands-on exploration of sensorimotor loops. In *Robotics in Education: RiE 2021 12*, pages 221–228. Springer.