


EduBal: An open balancing robot platform for teaching control and system theory

Working Paper**Author(s):**

Framing, Christian-Eike; [Hedinger, Raffael](#)  Iglesias, Emmanuel Santiago; Heeler, Frank-Josef; Abel, Dirk

Publication date:

2020-05-19

Permanent link:

<https://doi.org/10.3929/ethz-b-000462382>

Rights / license:

[In Copyright - Non-Commercial Use Permitted](#)

Originally published in:

arXiv

EduBal: An open balancing robot platform for teaching control and system theory

Christian-Eike Framing* Raffael Hedinger**
Emmanuel Santiago Iglesias*** Frank-Josef Heeler*
Dirk Abel*

* Institute of Automatic Control, RWTH Aachen University, DE-52074 Aachen, Germany (e-mail: {c.framing}{f.hesseler}{d.abel}@irt.rwth-aachen.de).

** Institute for Dynamic Systems and Control, ETH Zurich, 8092 Zurich, Switzerland (e-mail: hrffael@idsc.mavt.ethz.ch)

*** RWTH Aachen University, DE-52074 Aachen, Germany (e-mail: emmanuel.iglesias@rwth-aachen.de)

Abstract: In this work we present EduBal, an educational open-source hardware and software platform for a balancing robot. The robot is designed to be low-cost, safe and easy to use by students for control education. Along with the robot we present example tasks from system identification as well as SISO and MIMO control. Using Simulink, students can quickly implement their control algorithms on the robot. Individual control parameters can be tuned online while analyzing the resulting behavior in live signal plots. At RWTH Aachen University and ETH Zurich 28 units have so far been built and used in control classes. In first laboratory experiences students show high intrinsic motivation and creativity to apply the studied concepts of control theory to the real system.

Keywords: Control education, Laboratory education, System analysis, SISO control, MIMO control

1. INTRODUCTION

Control theory, while being essential for many of today's innovations, can seem an abstract and unfamiliar topic for students. Employing interactive learning as part of the curriculum, especially on a real system, can build a bridge between theory and practice. As part of it, students may develop an intuitive understanding of individual control characteristics. In the past, low-cost educational systems such as a mobile robot (Mondada et al., 2009) and a robotic arm (Soriano et al., 2014) have been proposed. A balancing robot, like the inverted pendulum, is an unstable, nonlinear and multiple-input-multiple-output (MIMO) system allowing a wide range of control-theoretic applications. Balancing robot designs for education have appeared in the literature (Howard and Bushnell, 2015; Gonzalez et al., 2017) and are commercially available¹. Most of these designs are built around the inexpensive Arduino microcontroller². This microcontroller is well supported by MATLAB/Simulink but has only limited processing power for useful features such as live communication or advanced control algorithms like Model Predictive Control. Some of the hardware designs may also be too fragile for direct student use.

In this work we present the *Educational Balancing Robot EduBal*. It is a low-cost and robust design enabling the

¹ e.g. the Balboa Balancing Robot by Pololu and the GoPiGo3 BalanceBot by Dexter Industries

² with the exception of the Raspberry Pi in the GoPiGo3

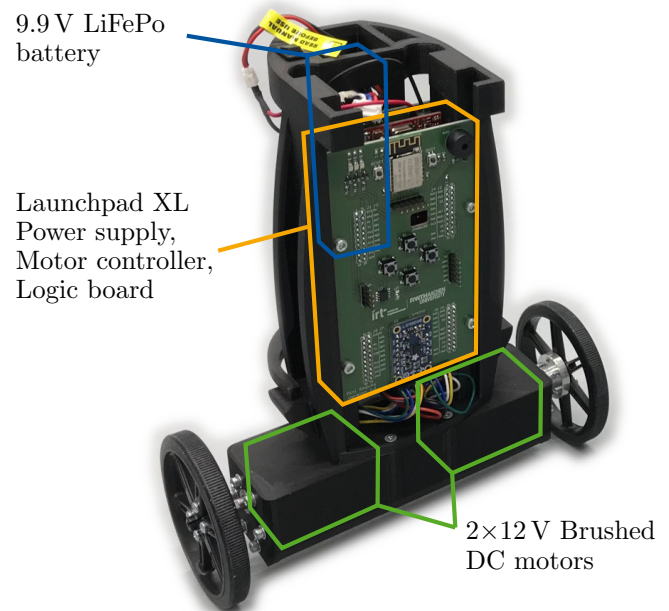


Fig. 1. Front view of EduBal with its individual parts highlighted

safe experimentation with a wide range of control-theoretic aspects from system analysis, control design and system identification. The selected microcontroller and actuators offer the potential for demanding control algorithms. For a wider audience, its hardware and software design is

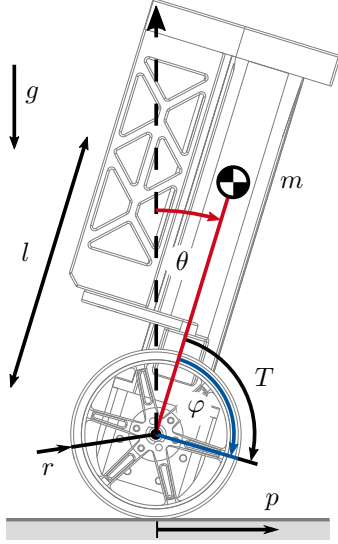


Fig. 2. Lateral plane view of EduBal

Table 1. Model parameters as measured in experiment

Parameter	Symbol	Value
Body mass	m	$9.33 \times 10^{-1} \text{ kg}$
Wheel radius	r	0.04 m
COM above wheel axle	l	$8.57 \times 10^{-2} \text{ m}$

made openly available in the project repository³ under the permissive MIT license.

In the remainder of the paper, we first derive a physical model of the robot in Section 2. The model is used to demonstrate example student tasks from SISO and MIMO control design in Section 3. Section 4 gives details about the specific hardware, software and safety implementation of the robot. First experiences from educational use are described in Section 5 before a conclusion is made and future work is suggested in Section 6.

2. ROBOT MODEL

Depicted in Fig. 2 is a lateral plane view of an EduBal model. The robot has two wheels of radius r which are assumed weightless. The wheel rotation angle φ is measured relative to the body axis and leads to the longitudinal position p . The upper body has mass m which is assumed to be located at a distance l from the wheel axle. Due to gravity g acting on the body, it may pitch with an angle θ . The electric motors, which are fixed to the upper body, produce a combined torque T acting on both the wheel and body. The experimentally determined model parameters are given in Table 1.

2.1 Mechanics

The mechanical dynamics of the robot can be derived using the Lagrange formalism (Frankovský et al., 2017):

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = Q_i - \frac{\partial D}{\partial \dot{q}_i} \quad (1)$$

where the Lagrangian is defined as the difference between the kinetic and potential energy of body and wheels:

$$L = E_k - E_p = E_{kB} + 2E_{kW} - E_{pB} - 2E_{pW} \quad (2)$$

The generalized forces acting on the system are the motor torques $Q_i = T$ and the generalized coordinates $q_i = (p, \theta)^T$. We neglect friction on both the wheel axle and ground and assume weightless wheels. Applying the Lagrangian formalism yields the two differential equations:

$$rlm\ddot{\theta} \cos(\theta) + rm\ddot{p} - rlm \sin(\theta)\dot{\theta}^2 = T, \quad (3)$$

$$2l^2m\ddot{\theta} + lm\ddot{p} \cos(\theta) - glm \sin(\theta) = -T. \quad (4)$$

For further analysis we linearize the system about the upper equilibrium point using the approximations:

$$\cos(\theta) \approx 1, \quad \sin(\theta) \approx \theta, \quad \dot{\theta}^2 \approx 0 \quad (5)$$

Substituting the terms (5) in (3) and (4) as well as equating both equations gives the linearization:

$$rlm\ddot{\theta} + rm\ddot{p} = -2l^2m\ddot{\theta} - lm\ddot{p} + glm\theta \quad (6)$$

2.2 Electric motor

The torque T is generated by the electric motors. For stabilization control design, we initially neglected their internal dynamics and modeled them as simple proportional element. During controller validation however, the motor gearbox showed significant backlash. This nonlinearity could not directly be incorporated into the linear model and complicated stabilization control. In order to mitigate this effect, we designed an underlying feedback proportional-integral controller (PI) for the wheel velocity $\dot{\varphi}$.

The PI controller was designed using pole-placement. An often used representation of the electric motor is a linear second-order model describing the electrical and mechanical dynamics. The corresponding motor parameters may be estimated using a manual armature resistance measurement and least-squares identification on measurement data (Saab and Kaed-Bey, 2001). With the goal of stabilization control, it is however sufficient to identify only the slower mechanical dynamics as a first-order model. In an experiment a voltage step sequence U was applied to the electric motor and the resulting wheel velocity $\omega = \dot{\varphi}$ was measured. The following dynamic model could be identified:

$$G_{EM} = \frac{\omega(s)}{U(s)} = \frac{2.6}{0.038s + 1} \quad (7)$$

The transfer function of the closed-loop system has two poles and one zero. The poles were placed so that a relatively long settling time of $t_s = 0.5 \text{ s}$ for the dominant pole, and a natural frequency of at most $f = 30 \text{ s}^{-1}$ were achieved. The goal was to reduce the actuator bandwidth and in consequence decrease excitation of the critical high frequencies.

For system analysis and control design, this underlying control loop needs to be included in the model. We approximate the closed-loop dynamics using a first order lag element

$$\ddot{\varphi} = -\frac{1}{t_{EM}}\dot{\varphi} + \frac{K}{t_{EM}}\dot{\varphi}_{\text{ref}} \quad (8)$$

³ <https://gitlab.com/rwth-irt-public/edubalembded>

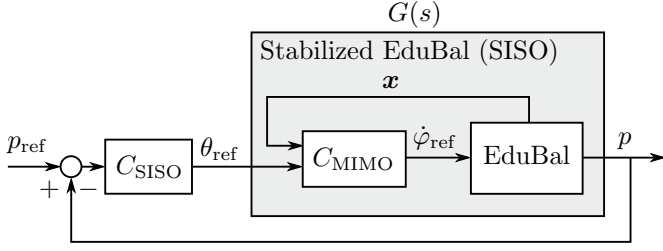


Fig. 3. Position control of robot as SISO problem

where $\dot{\varphi}_{\text{ref}}$ is the desired motor speed, K is the gain and t_{EM} the time constant. The given parametrization leads to $K = 1$ and $t_{\text{EM}} = 0.0994$ s.

2.3 Combined system model

By solving (6) for $\ddot{\theta}$ and using (8) with the relation $p = r(\varphi + \theta)$ we formulate the overall robot model as a linear state-space system

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}u \\ \mathbf{y} &= \mathbf{C}\mathbf{x} \end{aligned} \quad (9)$$

with state vector $\mathbf{x} = (\varphi, \dot{\varphi}, \theta, \dot{\theta})^T$, input $u = \dot{\varphi}_{\text{ref}}$ and output $\mathbf{y} = (p, \theta)^T$. The system matrices result as

$$\begin{aligned} \mathbf{A} &= \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{1}{t_{\text{EM}}} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{lm+rm}{t_{\text{EM}}\eta} & \frac{glm}{r\eta} & 0 \end{pmatrix} & \mathbf{B} &= \begin{pmatrix} 0 \\ \frac{K}{t_{\text{EM}}} \\ 0 \\ -\frac{lm+rm}{t_{\text{EM}}\eta} \end{pmatrix} \\ \mathbf{C} &= \begin{pmatrix} r & 0 & r & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{aligned} \quad (10)$$

where

$$\eta = 2ml + mr + \frac{2ml^2}{r}. \quad (11)$$

The matrix \mathbf{A} has eigenvalues with positive real part $\text{Re}(s_i) > 0$, hence the system is unstable for the upper equilibrium point. As the controllability matrix

$$\mathbf{C}o = (\mathbf{B} \ \mathbf{A}\mathbf{B} \ \mathbf{A}^2\mathbf{B} \ \mathbf{A}^3\mathbf{B}) \quad (12)$$

has full rank, the system is controllable and a stabilizing feedback controller can be designed.

3. SYSTEM ANALYSIS AND CONTROL DESIGN

3.1 SISO Position control

One motivation for designing EduBal was the use of it as a demonstrative example in basic control theory. Most of the relevant methods such as pole-zero analysis and root-locus control design apply for SISO systems. The task of stabilizing the full robot around its equilibrium point however is a MIMO control problem which is treated in more advanced control theory classes. For SISO stability analysis and control design the problem of controlling the robot's position can be used (see Fig. 3).

A controller C_{MIMO} that stabilizes the body angle needs to be designed beforehand. We use a three-state linear quadratic regulator (LQR) that controls the state $\tilde{\mathbf{x}}$:

$$\tilde{\mathbf{x}} = \begin{pmatrix} \dot{\varphi} \\ \theta - \theta_{\text{ref}} \\ \dot{\theta} \end{pmatrix} \quad (13)$$

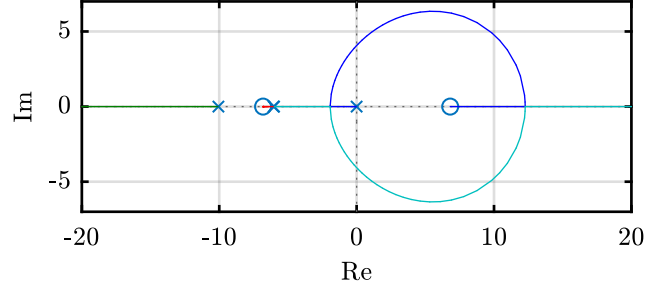


Fig. 4. Root locus diagram of $G(s)$ in Fig. 3

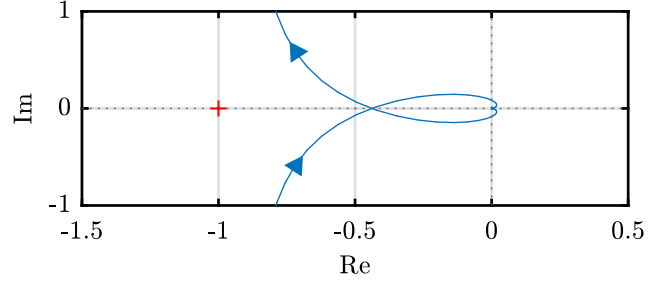


Fig. 5. Nyquist diagram of $G(s)$ in Fig. 3, here for a stable parametrization $K_p = 0.58$.

Using θ_{ref} , a target pitch angle $\theta_{\text{ref}} \neq 0$ can be achieved. The controller gain $\mathbf{K}_{\text{LQR}, 3 \times 3}$ is determined by minimizing the quadratic cost function

$$J(u) = \int_0^\infty (\mathbf{x}^T \mathbf{Q} \mathbf{x} + u R u) dt \quad (14)$$

where $\mathbf{Q} = \text{diag}(0, 10^0, 0)$ and $R = 10^2$ penalize control error in body angle and actuation energy respectively. Numerically solving the resulting Riccati equation gives the optimal controller parameters in continuous time:

$$\mathbf{K}_{\text{LQR}, 3 \times 3} = (-2, -88.75, -14.73) \quad (15)$$

The resulting closed-loop system with C_{MIMO} is SISO and has the input θ_{ref} and the output p . It is provided to the students in the form of a transfer function:

$$G(s) = \frac{P(s)}{\theta_{\text{ref}}(s)} = \frac{-27.96s^2 + 1297}{s^4 + 22.11s^3 + 157.6s^2 + 365.3s} \quad (16)$$

The characteristics of the stabilized robot can be analyzed using the pole-zero diagram (refer to the root-locus diagram in Fig. 4). Most notably it includes an integrator and a zero which does not have minimum phase. Within the laboratory, students are asked to link these characteristics to the observed behavior of the robot. As C_{MIMO} does not control the wheel angle φ , a body angle reference $\theta_{\text{ref}} \neq 0$ will lead to a position drift p over time which corresponds to integrating behavior. The non-phase-minimum zero characteristic may be observed as an undershoot in the step response of the system. Intuitively, in order to lean forward ($\theta > 0$), the robot briefly needs to drive in reverse ($\dot{p} < 0$) before changing direction ($\dot{p} > 0$) to keep body angle constant.

Students are then tasked to design a controller C_{SISO} which stabilizes the position. From the root-locus diagram, it can be determined that a proportional controller (P) is sufficient to shift all closed-loop poles to stability. Using the diagram, the resulting closed-loop behavior, i.e. oscil-

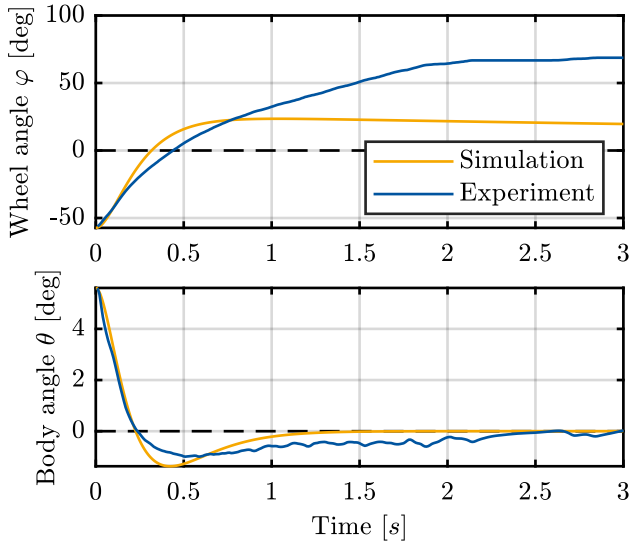


Fig. 6. Example trajectory of stabilized robot with full state-feedback control in simulation and experiment

lation for an underdamped system, can be tuned visually. Plotting the Nyquist diagram (Fig. 5), students can identify the stability margins and the critical controller gain. Finally the students implement the designed controller in a target model. The simulated characteristics are then validated experimentally on the actual robot.

3.2 MIMO Full state-feedback control

As a task for students from advanced control theory, EduBal can directly be stabilized using a MIMO controller. The goal is to stabilize all four states of the system in eq. (9) around the operating point $\mathbf{x}_0 = (0, 0, 0, 0)^T$. This controller concept may also serve as a performance baseline for other controllers. After deriving the linear model of the robot and implementing a wheel velocity controller, students are provided with the model parameters. The design of a full-state feedback controller then follows the same procedure as shown in Section 3.1. With $\mathbf{Q} = \text{diag}(10^0, 10^{-2}, 10^0, 10^{-2})$ and $R = 10^2$ the optimization of equation (14) gives the optimal controller parameters in continuous time:

$$\mathbf{K}_{\text{LQR}, 4 \times 4} = (-0.1, -2.04, -90.23, -14.97) \quad (17)$$

As the closed-loop system matrix $\mathbf{A}_K = (\mathbf{A} - \mathbf{B}\mathbf{K})$ only possesses eigenvalues with negative real part $\text{Re}(s_i) < 0$ the controlled dynamics are stable.

Like in Section 3.1, students are now able to validate the model quality and evaluate the controller performance on the actual robot. Fig. 6 shows the closed-loop behavior for an initial state $\mathbf{x}_0 = (-55^\circ, 0, 5^\circ, 0)$ from simulation and experiment. Observing the behavior for φ it can be noted that the robot successfully recovers from the initial disturbance $\varphi_0 = -55^\circ$ but, especially in experiment, shows a large overshoot. An explanation may be found in the unconsidered friction effects or in a slightly uneven ground during the experiment. The trajectory of θ shows similarities between simulation and experiment. The slight

oscillations can be attributed to the gearbox backlash which is not included in the simulation model.

4. IMPLEMENTATION

4.1 Hardware design

The EduBal robot is displayed in Fig. 1. It is assembled from a body, a stack of circuit boards, a battery and two electric motors. The complete robot weighs 0.99 kg and has the outer dimensions of $206 \times 101 \times 235 \text{ mm}$ ($W \times L \times H$). It is small enough to be placed and operated on a table. Bumpers are installed on the body to protect the circuit boards from damage in case of a fall. The body parts are made from PLA and are fully 3D-printable on a hobbyist machine. This allows for quick modification and replacement of individual parts. The 9.9 V 2100 mA h 40 C LiFePo battery is purposely placed high above the ground in a cage in order to raise the center of mass. As can be seen in Fig. 2 this asymmetric design in the lateral plane leads to an equilibrium body tilt angle of about 10.5° which is taken as controller reference.

Inside of the body is contained the circuit board stack with the Texas Instruments *Launchpad XL* evaluation board. The F28069M microcontroller is based on the C2000-architecture and runs at 90 MHz. Using the C2000 Embedded Coder Support Package, it is directly programmable via a USB cable. Contained in the support package is a blockset to access the low-level hardware interfaces. The board is extended with three custom circuit boards that are stacked on top and bottom of the microcontroller headers. The *Power Supply Board* provides a 3.3 V (2 A maximum) and 12 V (5 A continuous) line from the nominal 9.9 V battery. Given a PWM reference, the *Motor Control Board* can directionally drive two 12 V brushed DC electric motors using two H-bridges. For the EduBal two motors with an included gear transmission ratio of 30:1 are used. Each can deliver a wheel torque of 0.18 N m at a longitudinal velocity of 2.34 m s^{-1} at maximum efficiency. The shaft angle of the motor is directly measured using a magnetic encoder. The third *Logic board* combines a 9-axis inertial measurement unit (IMU) for measuring acceleration, angular rate and magnetic field, a Wi-Fi chip, user interaction elements and SPI and I2C headers for hardware extensions. All necessary robot components are compiled in a list which is available at the repository together with the CAD designs and PCB schematics. At the time of writing the cost of the necessary hardware for one unit is approximately €200.

4.2 Interfaces

For direct interaction with the running model on the robot, it features four buttons, three LEDs and a buzzer on the front that are freely programmable. Using an attached USB cable, Simulink models can be run in External mode. In this mode, students can visualize live data in a Simulink Scope and change model parameters online. For remote interaction the onboard ESP8266 Wi-Fi chip can be configured to create an access point or join an existing network. It is connected to the microcontroller's serial port and converts the incoming and outgoing data stream to UDP

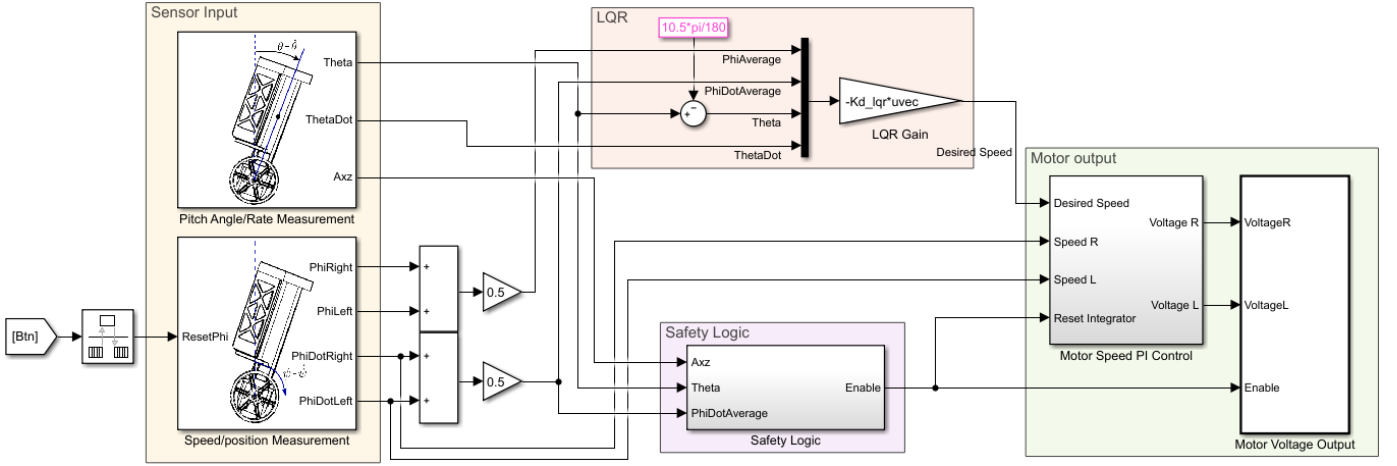


Fig. 7. Implementation of the full state-feedback controller as explained in Section 3.2 using blocks from the provided Simulink library

packages. Depending on the Simulink model implementation, arbitrary data can be sent and received wirelessly. For demonstration purposes, we implemented wireless keyboard control and live-plotting of sensor signals in a MATLAB GUI. Depending on the application, EduBal can be extended with additional sensors and processing hardware. New 3D-printed parts can be attached to the chassis and the *Logic Board* offers access to I2C and SPI headers to allow communication with the microcontroller board. As an example we designed a vision module and used it with EduBal (Fig. 9).

4.3 Software

Part of the EduBal repository is a software framework implemented in MATLAB/Simulink. It provides tools to set up the hardware, i.e. an IMU calibration and a DC motor identification procedure. An analytical model of the robot dynamics derived in Section 2 is included alongside a respective simulation. In order to make the robot operational, a reference Simulink model is contained that offers the possibility to choose from different controllers and allows remote control operation. All models are built using blocks from a provided Simulink library. Included are blocks for accessing hardware interfaces such as the IMU and encoder input, motor output, user interaction elements and Wi-Fi communication. For angle estimation, a complementary filter and a Kalman filter implementation are available. An example implementation of the full state-feedback controller from Section 3.2 that uses the provided blocks is shown in Fig. 7.

4.4 Safety logic

As the robot is used by students it must operate safely under all circumstances. For that purpose a few safety functions are implemented directly in the Simulink model. Consequently the model must run for them to be active.

- *Battery monitor*: Using the balancer cable of the LiFePo battery, the individual cell voltages are monitored. In case of low voltage an audible warning is given to prevent battery draining damage.

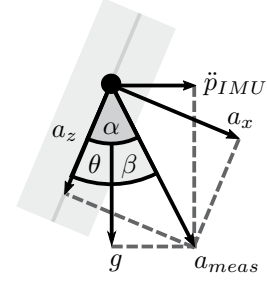


Fig. 8. Coordinate transformation from sensor to global frame to estimate longitudinal acceleration from IMU measurements

- *Angle limiter*: In order to prevent unstable behavior for poor controller implementations, motor control is deactivated for a body angle exceeding the range of $-30^\circ < \theta < 40^\circ$.
- *Wheel slip monitor*: Due to its small size, students tend to pick up the robot during operation. Depending on the controller implementation this leads to a rapid acceleration of the wheels. Putting the robot back down in this condition may cause unsafe behavior. To detect when the robot is picked up, we monitor the offset

$$\Delta \ddot{p} = \ddot{p}_{IMU} - \ddot{p}_{Whl} \quad (18)$$

where \ddot{p}_{IMU} is the longitudinal acceleration as measured by the IMU and $\ddot{p}_{Whl} = r\ddot{\varphi}$ is the expected longitudinal acceleration from observed wheel acceleration. When the robot is picked up or when it drives on slippery ground, the resulting wheel slip is detectable as a peak in Δp . Fig. 8 shows the IMU attached to the tilted body. To calculate longitudinal acceleration \ddot{p}_{IMU} the measured accelerations a_x, a_z in the sensor frame are transformed to the global frame using the estimated pitch angle θ :

$$a_{meas} = \sqrt{a_x^2 + a_z^2} \quad (19)$$

$$\alpha = \arctan\left(\frac{a_x}{a_z}\right) \quad (20)$$

$$\beta = \alpha - \theta \quad (21)$$

$$\ddot{p}_{IMU} = \sin(\beta) \cdot a_{meas} \quad (22)$$



Fig. 9. EduBal equipped with a vision module consisting of a Raspberry Pi 3B and camera used in the task of following a line on the ground

We purposely neglect the small influence of the rotational acceleration $\ddot{\theta}$ on \ddot{p}_{IMU} as it can only be estimated by a discrete-time derivative and is thus susceptible to noise.

5. EDUCATIONAL USE

The motivation behind EduBal is for it to serve as an example system for a variety of control tasks. Sections 2.2 and 3.2 are examples for tasks from system identification and control design that were given to RWTH Master students as part of a lecture about methods of Rapid Control Prototyping (Abel and Bollig, 2006). In the laboratory, 30 students in groups of three worked together in solving the tasks and implemented their controllers both in simulation and on the actual robot using the MATLAB/Simulink package. The Institute of Automatic Control developed the EduBal robot to replace a former human-sized balancing robot demonstrator. When introduced with the EduBal robots, students showed high curiosity to learn about the involved hardware. Provided with instructions, they were able to handle the robot without close supervision. The system was interactively analyzed and the controllers tuned using Simulink's External mode. After solving the given tasks students quickly started a discussion about new creative ways of applying concepts from advanced control-theoretic lectures to this system.

At the Institute for Dynamic Systems and Control of ETH Zurich, EduBal was adopted and it is planned to be used as part of both undergraduate and graduate lectures in control theory. In the exercises students derive a linear model of the robot and analyze its stability. The position control problem from Section 3.1 was developed to analyze transient system behavior of the stabilized robot as well as to design a SISO controller using the root-locus method. Students are also tasked to examine the effects of parameter uncertainty and gearbox backlash nonlinearity. As the Bachelor's course involves a number of 420 students, 20 robots were built beforehand.

Apart from the use in coursework, EduBal has been the subject of research to individual students. In order to improve robust behavior for varying system parameters, the

implementations of a Sliding Mode and a H_∞ controller were evaluated. With the goal of automatically following a line on the ground, the robot was extended with a vision module shown in Fig. 9. Based on the camera input a student team implemented a line detection algorithm and a lateral guidance controller that provides a wheel velocity reference to the EduBal onboard controller.

6. CONCLUSION

In this work we presented EduBal, an open-source design of a balancing robot aiming for the interactive education of control theory. The robot is built from 3D printed parts and custom circuit boards and due to its low cost can be built in greater numbers for the use in laboratories. Students can fully program the robot via Simulink using the provided block library for interfacing with sensor inputs and motor output. As an educational tool, it allows teaching a wide area of control theory in practice of which example tasks from system identification, SISO and MIMO control were provided.

Current work aims for the integration of the robot into advanced control curriculum. For that purpose the implementation of (Nonlinear) Model Predictive Control is investigated. Targeting the automatic parametrization of the available controllers in experiment, safe bayesian optimization is examined with the robot.

REFERENCES

- Abel, D. and Bollig, A. (2006). *Rapid Control Prototyping: Methoden Und Anwendungen*. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Frankovský, P., Dominik, L., Gmiterko, A., Virgala, I., Kurylo, P., and Perminova, O. (2017). Modeling of Two-Wheeled Self-Balancing Robot Driven by DC Gearmotors. *International Journal of Applied Mechanics and Engineering*, 22(3), 739–747.
- Gonzalez, C., Alvarado, I., and Peña, D.M.L. (2017). Low cost two-wheels self-balancing robot for control education. *IFAC-PapersOnLine*, 50(1), 9174–9179.
- Howard, B. and Bushnell, L. (2015). Enhancing linear system theory curriculum with an inverted pendulum robot. In *2015 American Control Conference (ACC)*, 2185–2192. IEEE, Chicago, USA.
- Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klapotcz, A., Magnenat, S., Zufferey, J.C., Floreano, D., and Martinoli, A. (2009). The e-puck, a Robot Designed for Education in Engineering. In *9th Conference on Autonomous Robot Systems and Competitions*, 59–65. Castelo Branco, Portugal.
- Saab, S. and Kaed-Bey, R. (2001). Parameter identification of a DC motor: An experimental approach. In *8th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, volume 2, 981–984. IEEE, Malta.
- Soriano, A., Marín, L., Vallés, M., Valera, A., and Albertos, P. (2014). Low Cost Platform for Automatic Control Education Based on Open Hardware. *IFAC Proceedings Volumes*, 47(3), 9044–9050.