Educational low cost platform for control engineering

Jorge Muñoz, Concepción Alicia Monje, Lisbeth Karina Mena, Carlos Balaguer
RoboticsLab, Universidad Carlos III de Madrid, Avenida Universidad 30, 28911 Leganés (Madrid)
jmyanezb@ing.uc3m.es, cmonje@ing.uc3m.es, lmena@pa.uc3m.es, balaguer@ing.uc3m.es

Abstract

Control engineering is a highly experimental field, therefore, the more practice activities, the better understanding of the basic concepts. On the other hand, the class resources are never as plentiful as desired, and the experiments require expensive equipments and maintenance. This work proposes a low cost experimental platform designed to be used in control engineering subjects. Although the tight budget has some drawbacks related to device quality, particular solutions are proposed, leading to a working platform with very good performance, that has been used in different applications such as lab practice classes and advanced control strategies projects. The platform performance shows that the use in both practice labs or controller design is possible showing very good results.

Keywords: Control engineering, low cost platforms, education, remote laboratory.

1 Introduction

Teaching control engineering is a challenging issue. Besides the important theoretical background, the field is highly experimental, therefore, a good amount of practice is needed in order to achieve an appropriate understanding.

The problem with experimental practice is that the equipments and tools needed for laboratory experiences are usually expensive, therefore, resources must be well planned in order to comply with a budget. Often this results in a reduced number of experiments and a group planning.

Using today's low cost electronics which are readily available, this issue can be addressed in a different way. Low cost lab equipment design can lead to a greater experimentation opportunity, increasing the student motivation and engagement.

Some authors have proposed similar ideas in previous works. For instance, in [1], a flow control system kit is proposed with the intention of providing a portable system with a low price. Although the cost is less than 1000 USD, which is really lower than the commercial equivalent, this platform is only affordable by an institution.

The cost of such equipment can be further reduced or even totally waived using virtual simulations as proposed in [2], but at the cost of losing the benefits in a field practice lab, especially, the incentive of working with a real system disappears.

More aligned with this proposal is the approach of [3], which proposes a DC motor control system with a price around 30 EUR (motor and controller) or 80 EUR (including a Raspberry pi). Note that these estimations are based on 2018 prices of specific vendors, and current cost may differ.

In this work, we have developed a new low cost control platform based on a similar idea, but with a reduced set of components. The proposed system is composed by:

- Single board computer (SBC).
- General purpose input output (GPIO) system.
- H bridge driver.
- Plant.
- Sensor.

Due to the fact that SBC boards like Nvidia Jetson or Raspberry Pi usually include a GPIO system, we propose to use them for direct driver control, reducing even more the platform cost but retaining all the system features. For convenience, all the components can be arranged using a printed circuit board as shown in Fig. 1, or mounted on a 3D printed support (not shown in the figure).

In our case, the hardware selected is:

1. Pololu DC motor model FIT0482.
2. Hall effect encoder with 28 pulse/rev.
3. TB6612FNG driver.
4. Convenience PCB.

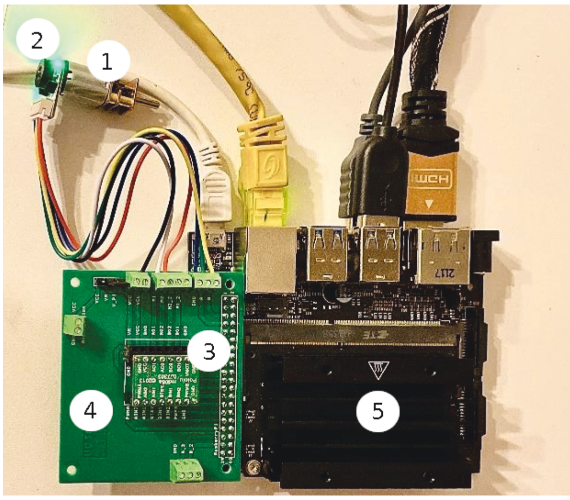

Figure 1: Low cost platform showing all the components.

5. SBC with GPIO Nvidia Jetson.

Access to plant inputs and outputs is needed in order to perform the control tasks. In our case two specific C++ libraries were developed, robot-device [1], which offers a set of classes and functions able to access the hardware operation, and fcontrol [2] featuring helpful control tools like transfer functions, accurate sampling time waits, or controller blocks.

1.1 Plant model

In the first place, in order to perform plant simulations or control, it is essential to obtain a model. Since the physical parameters of the motor can be measured, a model based on these parameters can be obtained. The values shown in Table 1 were directly measured from the motor (see [4]).

Parameter	Value	Units
Inertia (J)	4.0771E-08	kgm^2
Viscous friction (fv)	4.7047E-08	Nms
Torque constant (k)	0.00335915	Nm/A
Armature resistance (R)	10.3	Ω
Electric inductance (L)	0.0000711	H

Table 1: Motor measured constants.

Having these constants, the state space equations of the DC motor are defined as follows:

[1] https://gitlab.com/uc3m-sofia/robot-device
[2] https://gitlab.com/munoz.yanez/fcontrol

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & fv/J & k/J \\ 0 & k/L & R/L \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ 1/L \end{bmatrix} u(t), \tag{1}$$

where $x(t) = [\theta(t), \dot{\theta}(t), i(t)]^T$ are the system states over time, and $u(t)$ is the system input (voltage in this case).

Using the same motor constants, the transfer function model of the system can be defined through Eq. (2).

$$G_v = \frac{k}{(J \quad s + fv)(R + L \quad s) + k^2}, \tag{2}$$

where G_v is the transfer function model defining a single input in voltage, and a single output in motor angular velocity for the plant.

The system shown in Fig. 1 has a cost close to 100 EUR because a high end SBC was used, but using low end components the cost should be around 70 EUR, which is really affordable. This is very interesting, as students can buy it and have their own system for personal use.

As expected, there are some drawbacks as a consequence of the low price. Parts quality are not the best, making the plant modeling and control challenging indeed.

In this case, the encoder resolution is really low (28 pulses per motor shaft revolution), degrading the velocity signals, and the high viscous friction results in an important dead zone (around 10% of the actuation range). Before using the platform in a student setup, the problems to be solved are:

- Low signal quality.
- Dead zone.

Given the described sensor limitations, the output signal quality is not good enough to prepare a practice session, but a well designed control scheme can solve these problems. In the following sections these solutions are proposed and verified.

2 Platform optimization

The motor axis angular position (θ) signal has an acceptable accuracy, but the angular velocity ($\dot{\theta}$) shows spikes due to the low encoder resolution and noise. An example of the open loop velocity measurement is shown in Fig. 2.

This is a problem, for instance in the case of a proportional integral derivative (PID) controller,

https://doi.org/10.17979/spudc.9788497498418.0295

as the spikes will make the derivative component unstable.

2.1 Signal quality enhancement

Two solutions are proposed to smooth the velocity measurements: 1) filter the output signal and 2) use a state observer to estimate the velocity variable through the system state.

In the filter case, the system dynamics must be conserved while removing the unwanted effects. Figure 2 shows a time constant below 0.03 s (exactly 0.027 s), which provides a system frequency around 37 rad/s. Using a lowpass filter just above, for instance 40 rad/s, will keep the system dynamics while removing unwanted measurement values. The filter results are shown in Fig. 2.

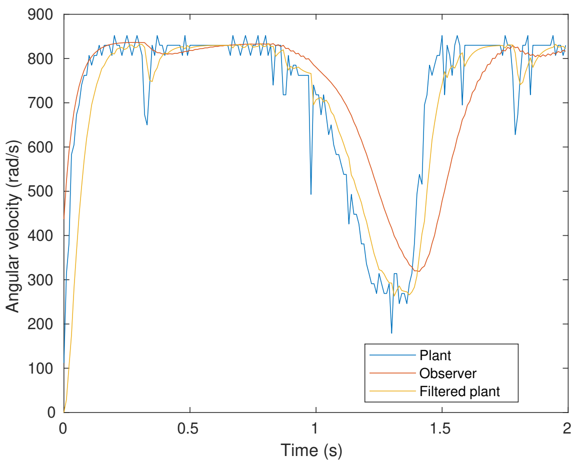

Figure 2: Plant velocity output (rad/s) showing disturbances. Original and enhanced signal versions.

The other option proposed to enhance the signal is a state observer. This technique uses a state space model with a feedback loop that corrects the state estimation based on the plant measurements of inputs and outputs. Adjusting properly the observer gains, the model state values can be very close to the real plant. The observer application results can be seen in Fig. 2. For further detail on the state observers theory see, for instance, [5].

Comparing the results, it is obvious that the observer has a better accuracy and provides smother results for velocity estimation. Just in the case of a motor shaft blocking (torque disturbances), the output tracking is worse compared to the filter. That is because the observer estimation depends both on inputs and outputs of the system, and the motor blocking disturbances are applied while keeping the input constant (in this case a 3V input). On the other hand, the filter has a worse predictive behavior as it depends just on output signal and does not have a system model, but can track more closely the plant output in the case of torque disturbances.

Once the signals are correct, both in velocity and position, it is possible to close a loop in that variables, allowing us to propose a solution to the dead zone issue.

2.2 Dead zone removal

Dead zone can be addressed using a closed loop scheme diagram like the one shown in Fig. 3, where C is a controller tuned for some desired specifications, and G_v is our plant in velocity actuation mode, being voltage the input (u) and the smooth velocity signal the output (y).

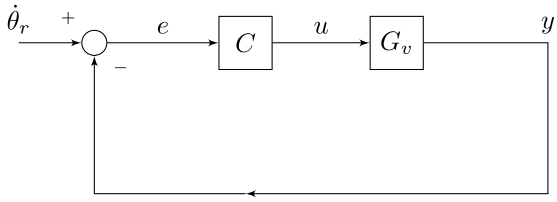

Figure 3: Velocity closed loop for dead zone removal.

In this way, using a proportional integral controller (PI) in the block C can solve the dead zone in the following way: whenever the velocity goes to zero, the integrator ramps up the control signal until the system starts moving, allowing to reach any speed at the motor shaft, no matter how small.

Once the problems are addressed, the educational applications proposed will be discussed.

3 Educational applications

In order to use the platform as an educational tool, two interesting activities can be proposed: lab experiments (local or remote), and controller design projects.

The laboratory experiments can be designed using the platform as a plant to be controlled. For instance, the student exercise could be finding the PID controller that fulfills some specifications, or perform a system identification, or any control task involving inputs, outputs and controllers.

Usually, this kind of activity requires a specific interface in order to act on the system behavior, like controller parameter configuration, or set the control reference. Given that the chosen platforms work under the linux [3] environment, there are lots

[3]https://www.kernel.org/

of tools that can be used. Probably, the best option in this case is to use a web interface, which provides a rich environment for user interaction and allows both local and remote connection possibilities. Figure 4 shows an example of the PID tuning lab web interface used during the remote lab experience.

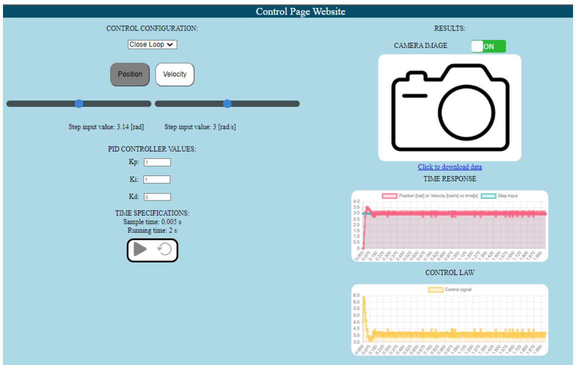

Figure 4: Example of web interface designed to practice position and angular velocity control using feedback and a PID controller.

Notice how the dialogues allow to configure the parameters and target reference, and also run the experiment.

In the case of a controller design project, the student must have a platform for personal use (own or borrowed). In this way, a direct access to the plant inputs and outputs is available through the libraries mentioned before, allowing us to define any control scheme desired.

An obvious application for this setup is to prepare the final project at engineering degrees. In this case, the platform allows to define a control strategy with the student, and develop specific algorithms and experiments.

At the moment, we have concluded three final degree projects using this platform, and two are in the way. Some details of these project results are discussed next.

4 Results

The following projects have been developed using the low cost platform:

- Robust control strategies based on Internal Model Control for servo motors and soft robotics [6].
- Design and application of linear quadratic controller in electromechanical systems [4].
- Development and implementation of control algorithms based on Extremum Seeking Control for robotic systems [7].

Each project has implemented a different control scheme to obtain a servomechanism with an accurate and constant velocity or position output. The three strategies proposed were internal model control (IMC), linear quadratic regulator (LQR) and extremum seeking control (ESC). An overview of the results obtained are discussed below.

4.1 IMC regulator

Internal model control scheme is based on two models, the plant model and a filter defining the desired system behavior. The idea behind this controller is really simple. Using the inverse plant model, it is theoretically possible to cancel the plant dynamics and get a perfect reference signal tracking. The problem with this approach is that, as plant models use to be causal, the inverse usually don't. This is where the filter becomes important. Choosing an adequate transfer function (normally adding poles to compensate the inverse model order) for that filter, the regulator can be causal and the behavior can be specified by the filter dynamics.

With that strategy any performance can be achieved using an open loop control scheme, but given that model accuracy is never perfect, there will be differences between model and plant, leading to a different behavior. In order to achieve error canceling, a feedback branch is normally used, which compensates the system input in the case of a mismatch between plant and model behaviors. A block diagram of this scheme is presented in Fig. 5.

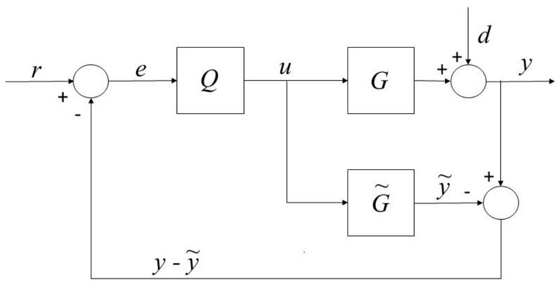

Figure 5: IMC control block diagram.

In the figure, plant (G) and plant model ($\tilde{G}$) are fed with the controller output signal (u), which is obtained as the controller's (Q) response to the error input (e). The controller transfer function is computed as shown in Eq. (3).

$$Q = \frac{f}{G}, \tag{3}$$

https://doi.org/10.17979/spudc.9788497498418.0295

where f is the filter that specifies the system behavior, usually of the form $1/(\lambda s + 1)^n$, being n the non-causal order to cancel in the model inverse function. See [8] for more details on this control strategy.

During this project the IMC scheme was successfully applied to the proposed low cost plant, obtaining very good results. In this case the goal was to develop a velocity servo. The performance specification filter used in this case is the following:

$$f = \frac{0.05z}{z^2 - 1.6z + 0.65}. \tag{4}$$

Using a discrete version of the plant model described in (1), and the filter shown in (4), the controller Q was obtained according to (3). The IMC application to control the motor velocity is shown in Figs. 6 and 7 for a 500 rad/s step input.

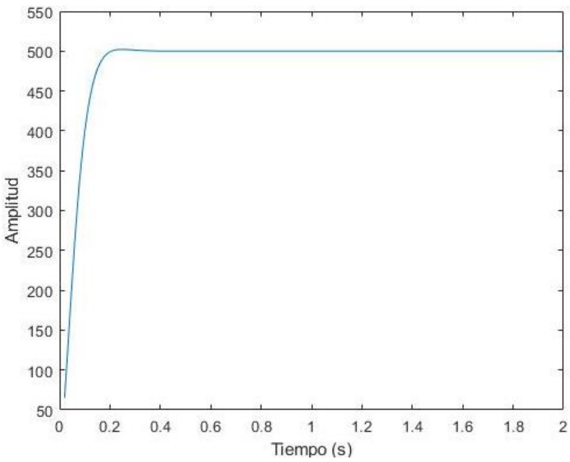

Figure 6: Performance specification in the IMC controller case.

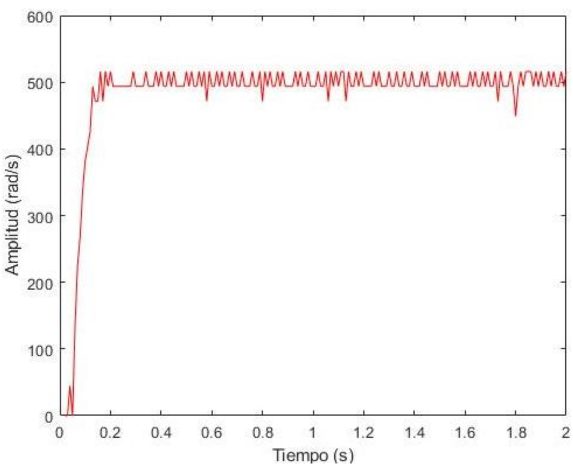

Figure 7: Real behavior of the IMC controller.

The performance in this case is really impressive. Although the output signal is not smoothed at all, the time response of the plant and specifications are really similar, and the IMC feedback branch effectively solves the issues described before.

For more information on this project see [6].

4.2 ESC control strategy

Extremum seeking control is a very wide field (see [9]). In this case, the applied method is currently known as Gradient-Based Extremum Seeking Control (GESC) as described in [10]. It is based on continuous measurement of a cost function which maps an optimization parameter versus the actuation variable. It uses the gradient of that function to track the optimum point in real time. However, in this case, filtered estimates of the gradient are used. An important parameter in this type of control is the slope factor, that controls how fast the algorithm converges to the extreme.

Using an observer based on the plant model described in (1), and a slope factor of 10^{-6}, the results for a 500 rad/s input step is shown in Fig. 8.

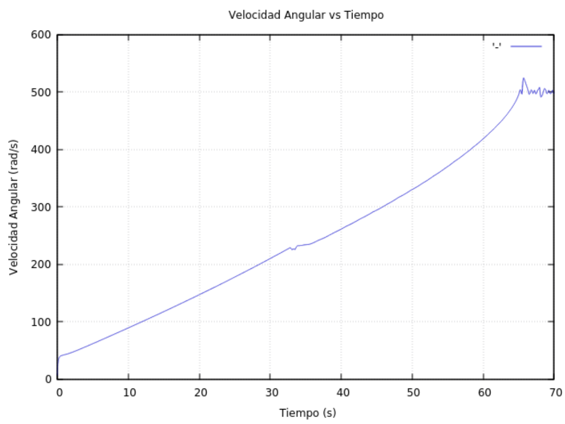

Figure 8: Time response under a 500 rad/s input step for the ESC controller.

Note that this algorithm is quite slow, which is due to a very small slope factor. It is needed because of the noisy signal available in velocity. This factor can be raised when the cost function keeps stable enough along the experiment.

For more information on this project see [7].

4.3 LQR control strategy

This optimization strategy is able to determine the feedback gains in a state feedback control scheme. The final performance specification of the system depends on the choice of matrices Q and R. This

https://doi.org/10.17979/spudc.9788497498418.0295

allows to penalize the feedback if individual states of the state vector by adjusting the Q matrix, or, in a similar way, the inputs vector through R matrix. For more information on this strategy see [11].

Therefore, LQR allows us to adjust what is important to the user. For instance, to save energy in our system when an actuator is expensive, changing matrix R that corresponds to the given actuator can reduce the actuation effort while keeping the stationary behavior.

Using the plant model described in (1), and the identity matrices for both specifications, Q and R, the LQR application to control the motor position provides the results shown in Figs. 9 and 10 for a 1000 rad input step.

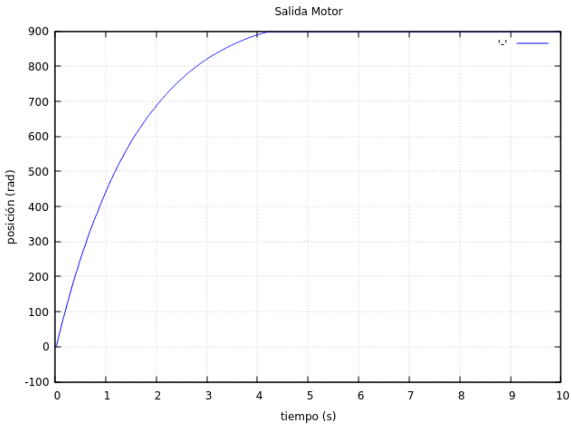

Figure 9: Real behavior of the LQR controller case.

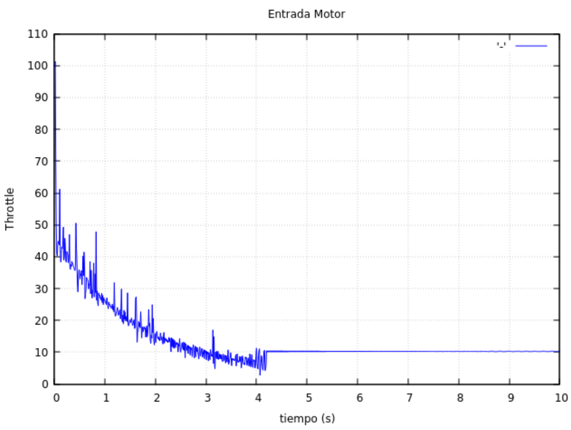

Figure 10: Control signal of the LQR controller.

The performance in this case is really good, but, as the controller was applied to the open loop plant, there is a dead zone as discussed before that prevents the system to achieve zero error. Note that, after reaching an actuation throttle value of 10 (10% of the full actuation value), the motor stops and the control cannot continue.

Nevertheless, this issue can be solved with a low level loop as described before, and taking that system as the plant to design the controller.

For more information on this project see [4].

5 Conclusions

This paper presents the development of a low-cost platform for educational applications in control engineering. A platform with these characteristics has a large number of advantages, among others, a greater availability of equipment to prepare laboratory experiments and the possibility of personalized control projects for each student.

However, the reduced price implies a deterioration of the system characteristics, making control more difficult. This could be an asset in describing possible failures in real systems, although in order to design a practice similar to those currently existing, two problems must first be solved: optimize the output system signals and remove the motor's dead zone.

The proposed solutions involve filtering the signals or estimating them using state observers, or creating a low-level control loop. The solution to these problems extends the platform purpose, allowing the design of many different practice sessions, from basic to advanced levels of teaching.

However, the platform can also be applied (and has been applied) in the development of control projects where the student solves these problems by proposing a control technique, which can be existing or novel.

In this sense, three projects have been carried out with different control strategies: IMC, ESC and LQR. The results are very good in all cases, although the ESC control is not as good as the others.

These results validate the platform as a real alternative to the current control mock-ups, but have several extra features, such a reduced cost and size, or the possibility to run the practice in a remote location.

Acknowledgement

The research leading to these results has received funding from the project SOFIA: Articulación blanda inteligente con capacidades de reconfiguración y modularidad para plataformas robóticas, with reference PID2020-113194GB-I00, funded by the Spanish Ministry of Economics, In-

 https://doi.org/10.17979/spudc.9788497498418.0295

dustry and Competitiveness, and from the project RoboCity2030-DIH-CM, Madrid Robotics Digital Innovation Hub, S2018/NMT-4331, funded by "Programas de Actividades I+D en la Comunidad de Madrid" and cofunded by Structural Funds of the EU.

References

[1] A. G. Abdullah, D. L. Hakim, M. A. Auliya, A. B. D. Nandiyanto, and L. S. Riza, "Low-cost and portable process control laboratory kit," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 16, no. 1, pp. 232–240, 2018.

[2] I. Arora, K. M. Moudgalya, K. Venkata, V. Chakraborty, R. Rokade, and R. Rakhi, "A low cost, scalable, virtual control laboratory," *2011 9th IEEE International Conference on Control and Automation (ICCA)*, pp. 1139–1144, 2011.

[3] T. Docekal and M. Golembiovsky, "Low cost laboratory plant for control system education," *IFAC-PapersOnLine*, vol. 51, no. 6, pp. 289–294, 2018. 15th IFAC Conference on Programmable Devices and Embedded Systems PDeS 2018.

[4] S. Vázquez, "Diseño y aplicación de regulador cuadrático lineal en sistemas electromecánicos," bachelor's thesis, Universidad Carlos III de Madrid, 2022.

[5] N. S. Nise, *Control Systems Engineering*. Wiley, 7 ed., 2019.

[6] M. D. Marcos, "Estrategias de control robusto basado en internal model control para servomotores y robótica blanda," bachelor's thesis, Universidad Carlos III de Madrid, 2022.

[7] B. Nadal, "Desarrollo e implementación de algoritmos de control basados en extremum seeking control para sistemas robóticos," bachelor's thesis, Universidad Carlos III de Madrid, 2022.

[8] A. Datta, *Internal Model Control Schemes*, pp. 47–58. London: Springer London, 1998.

[9] C. Zhang and R. Ordóñez, *Introduction*, pp. 3–29. London: Springer London, 2012.

[10] P. Li and R. Horowitz, "Control of smart exercise machines. ii. self-optimizing control," *IEEE/ASME Transactions on Mechatronics*, vol. 2, no. 4, pp. 248–258, 1997.

[11] K. J. Åström and R. M. Murray, "Feedback systems: An introduction for scientists and engineers," *Feedback Systems: An Introduction for Scientists and Engineers*, 01 2008.