Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 55-17 (2022) 174–179

# From-scratch development and improvement of a problem-based learning course: Nonlinear Model Predictive Control for Chemical and Biochemical Processes ⋆

**Jose Matias** * **Johannes Jäschke** *

* *Chemical Engineering Department, Norwegian University of Science and Technology, Sem Sælandsvei 4, Kjemiblokk 5, Trondheim, Norway*

**Abstract:** This paper describes the development from scratch of a Nonlinear Model Predictive Control course, which was designed for $5^{th}$ year Process Engineering master's students at the Norwegian University of Science and Technology (NTNU). After taking the course, students are able to program their own nonlinear model predictive controller (NMPC) in MATLAB, and have gained a good understanding of the most important issues related to the implementation of this type of controller in chemical and biochemical processes. The course follows problem-based learning principles. It is heavily focused on "doing", where coding assignments are a large part of the course load. The student learning is assessed by an unconventional combination of formative feedback (one-to-one session, where we discuss their assignments and the solution strategies of the students) as well as summative feedback (systematic grading of their assignments and final exam). The student perception of the course, which was assessed by a feedback questionnaire, shows that the course adds value to them by not only helping them write more interesting Master's projects but also by improving their skill set for the job market.



*Keywords:* Pedagogy in control engineering, challenges in control engineering curricula, problem based learning, programming

## 1. INTRODUCTION

The scope of process control has been changing along the years. Typically, it was limited to regulation of key process variables in the presence of disturbances such that a required product specification is achieved safely and consistently (Pistikopoulos et al., 2021). Consequently, the syllabus of control courses has followed the same guidelines. For example, when designing such courses, it is consensus that classical tools such as Laplace transformation, closed-loop analysis, and lead/lag/PID design should be the focus of attention (Rossiter et al., 2019).

However, new developments in optimal control theories and computational methods have been enabling large-size practical control applications (Biegler, 2018). Consequently, the process control scope has broadened and topics such as the minimization of energy consumption, or maximization of the productivity have been gaining attention (Pistikopoulos et al., 2021). Not only in theory, but also in practice (e.g., Rawlings et al. (2018)).

Therefore, instructors teaching process control courses in graduate levels have to face the challenge of designing a syllabus that attends the current needs of the process control community. The challenge consists not only of finding the right course structure to foster relevant technical skills, but also to provide the right balance between fundamental theory and practical applications.

In this paper, we describe the from-scratch development and improvement of a nonlinear model predictive control course. The course was designed for $5^{th}$ year Chemical/Process Engineering master's students at the Norwegian University of Science and Technology (NTNU). The aim was to cover the main state-of-the-art methods for dynamic optimization and control and how to implement them in practice.

To achieve these goals, we designed this course based on problem-based learning principles. The students learn how to code a nonlinear model predictive controller (MPC) from scratch in MATLAB using three different methods (Direct Single Shooting, Multiple Shooting and Orthogonal Collocation on Finite Elements). Then, at the end of the course, they have a lecture about implementation of NMPC, which is complemented by guest lectures from industry.

Based on formal feedback from the students (see Section 6), the students have a good perception of this hands-on approach. They enjoyed the practical part of the coding, despite the heavy workload associated with it. Several students used the concepts and the codes developed in the course in their Master's projects. Also, one of the students that took the course has been hired to work for a company that provides model predictive control solutions, and several others have been hired to work in IT companies.

⋆ The authors acknowledge financial support from the Norwegian Research Council, SUBPRO, grant number: 237893.


10.1016/j.ifacol.2022.09.276

Therefore, we believe that the course has enhanced the capabilities of the students to meet the future challenges that they will face either in research and industry.

## 2. BACKGROUND: MOTIVATION FOR DEVELOPING THE COURSE

We designed the course in 2019 to address four issues that students at our department faced at the time:

(1) Lack of experience with and knowledge about state-of the art methods for process automation and control;
(2) Every year, ca 15-20 students at our department work on Master's projects that involve some form of dynamic optimization and model predictive control; However, in the Chemical Engineering curriculum, there was no course providing the necessary prerequisites, such that these students needed to learn the basic principles on their own. Courses offered by other departments were too detailed and included many aspects that are were relevant for most of the work at our department;
(3) Chemical Engineers are required to be increasingly qualified in digital automation technologies, and programming complex computer programs;
(4) Chemical engineering research has been largely dominated by bioengineering in the last decades (Varma and Grossmann, 2014), while control is still mainly taught with focus on oil refineries and petrochemical plants (Pistikopoulos et al., 2021). The students need to understand the capabilities and potential of process control when applied to bioprocesses.

Faced with this scenario, we created the "Nonlinear Model Predictive Control for Chemical and Biochemical Processes control". The course was designed with 3,75 credits (approx. 15 hours of classroom teaching) and is taught as normal course module (1/2 semester). The main activities in this course are focused on programming and implementing an advanced model predictive controller in a bioreactor case study.

To address the four needs above, the course is heavily focused on "doing", where the concept was to give a minimum necessary content of theory, and then let the students work on implementing their own controller software.

Due to this slightly different approach from traditional control courses, we decided to use an unconventional combination of formative and summative assessment of the student learning. The mandatory programming exercises count 70% of the final grade. Each exercise is designed to solve a problem, and the complexity builds up during the course. After each exercise, the students receive one-to-one formative feedback, where we discuss the solution approach, and the choices the students made. In addition, each assignment is graded and the students receive written feedback. The final oral exam (30% of the final grade) is more focused on the theoretical background.

## 3. INTENDED LEARNERS

Although the syllabus was developed for $5^{th}$ year Chemical/Process Engineering master's students, the course is open for students from different departments and backgrounds. The prerequisite is only coding experience, preferably in Matlab and Python. No *a priori* process modeling/system identification knowledge is required since we provide all models used in the course. Moreover, we only use simple models (for instance, see the bioreactor model in Bequette (2003) - Module 8) because we do not want the students to struggle with complex system equations, while the main focus should be on how to formulate and solve dynamic optimization problems.

## 4. QUICK TOUR OF THE COURSE

Figure 1 shows the syllabus of the course offered in 2021. The lectures are designed to last for 90 min in total (two blocks of 45 minutes plus a 15-minute break).

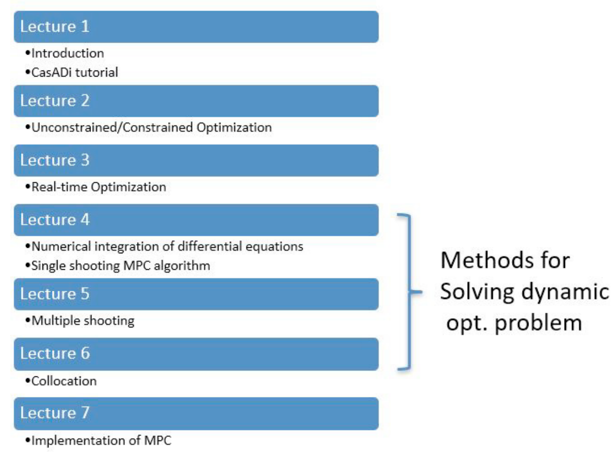


Fig. 1. Course Syllabus - Fall/2021

At the beginning of the course (*Lecture 1*), we give a quick introduction to model predictive control (MPC), its capabilities, and implementation challenges. Several practical examples are given for contextualization. In the first week, we also offer a lecture/tutorial about CasADi (Andersson et al., 2019), which is an open-source tool for nonlinear optimization and algorithmic differentiation. CasADi interface in MATLAB is used throughout the course for the coding of a nonlinear model predictive controller.

In *Lecture 2*, we review important concepts in unconstrained and constrained optimization. They are necessary for the implementation of different methods for numerical optimal control. At this point of the course, we offer an extra tutorial introducing IPOPT (Wächter and Biegler, 2006), which is the solver used during the course. We also introduce some MATLAB debugging concepts to help the students with their assignments, such as: how to add breakpoints to your MATLAB script, how to step into functions and scripts, etc. The attendance to this extra tutorial is voluntary, but it was well received.

*Lecture 3* focuses on Real-time Optimization (RTO), specifically on the two-step method proposed by (Chen and Joseph, 1987). Despite not being strictly a model predictive control method, RTO is used to provide a simplified optimization setting, which creates context about how the abstract concepts of unconstrained/constrained optimization of *Lecture 2* can be applied to chemical engineering. Also, we use RTO to introduce the concept of repeatedly optimizing the process that is revisited in the last part of the course when the model predictive controller is implemented. During the lecture, we present a brief overview of common draw-backs of the traditional two-step RTO, which are used as motivation to the relevance of optimizing the transient behavior of a system (dynamic optimization).

Coming to the solution of dynamic optimization (*Lecture 4 - 6*), we present three state-of-the-art strategies based on Nonlinear Programming (NLP). We present them in order of complexity, i.e., direct single shooting, multiple shooting, and orthogonal collocation on finite elements. For more details about the methods, please refer to Biegler (2010).

Since single shooting and multiple shooting employ a numerical integrator, we start this block of lectures with a quick overview of numerical integration methods. Also, in *Lecture 5*, we give a short introduction about direct and adjoint sensitivity methods as well as automatic differentiation, which will be required for providing gradient information to the NLP solver. Finally, in (*Lecture 7*), specific cases of NMPC/MPC implementations are illustrated with process examples. In addition to the guidelines for practical implementation, we also focus the discussion on problem formulation and computational issues.

The course is complemented by one or two guest lectures from industry (so far Equinor, Cybernetica AS, and Perstorp have presented), that demonstrate how the concepts learned in the course are directly relevant in an industrial environment.

### *4.1 Exercises*

As mentioned before, the course is heavily focused on "doing". The main work is done by the student by solving problems in the exercises given the concepts introduced in the lecture. At the end of each exercise, we also added 1-2 questions for reflection, such that the students develop a deeper understanding of the matter. The exercises were designed to be solved in about 3 to 4 hours each. The six assignments of the course are shown in Figure 2.

| - | CasADi tutorial |
|---|---|
| Exercise 1 | Constrained and unconstrained optimization |
| Exercise 2 | RTO exercise |
| Exercise 3 | Single shooting |
| Exercise 4 | Multiple shooting |
| Exercise 5 | Collocation |
| Exercise 6 | MPC implementation |

Fig. 2. Course Assignments - Fall/2021

By considering Figures 1 and 2, it can be seen that the exercises are carefully designed to mirror the contents presented in the theoretical lectures. The students have one week to work on the assignments. However, before the mandatory assignments, the students have access to a CasADi example (in the form of a tutorial), where we show them how to use CasADi functionalities of interest. *Exercise 1* is also an "introductory" assignment. Here, the students practice how to write an optimization problem in CasADi/MATLAB and how to characterize the optimal solution in toy mathematical examples.

After the students have learned how to use the software with toy examples, we start to focus on NMPC-related tasks. In *Exercise 2*, the students first develop a simple bioreactor model and have to solve a steady-state economic optimization problem, which is simpler to code. In comparison to *Exercise 1*, the bioreactor provides a more realistic example to practice the coding of optimization problems. The bioreactor example is used throughout the course, such that the students become familiar with it, and can focus on understanding the new concepts in the exercises. In the further course of this exercise, the students code a version of the two-step steady-state RTO without the model adaptation step. This provides them with a code framework for repeatedly solving a process optimization problem, which is used as a basis for repeatedly solving the NMPC problem in the last exercise.

As the course evolves (*Exercise 3 to 5*), The students to implement more and more sophisticated dynamic optimization approaches to optimize the performance of the bioreactor in terms of its transient economic performance. They build on the methods of the previous assignments and use them as a baseline for discussing the performance and implementation shortcomings. Finally, in *Exercise 6* the students put all the pieces of the previous course together, and implement their own version of a Nonlinear model predictive controller, that, at each sample time, queries the plant state, computes the optimal predicted trajectory and implements the computed first input move.

For helping the students, we have 90-minute non mandatory help sessions for the assignments. They happen a few days after the exercise is released. The idea is that the students try to solve the problem before coming to the session. Then, during the classroom time, we only discuss specific issues that the students are facing. Since students tend to have similar questions, we encourage the students to discuss the problems among themselves. The goal of these sessions is to engage the students and to create an environment for not only clarifying doubts but also sharing experiences in coding. Moreover, we believe that these non-mandatory classes are essential for building up a relationship between us and the students.

Table 1 shows an example of a week plan that was used when we offered the course in 2021. Here, "*Exercise - out*" indicates when the assignment becomes available to the students, and "*Exercise - in*" the assignment due date.

Table 1. Example of course week plan.

| | *Monday* | *Tuesday* | *Wednesday* | *Thursday* | *Friday* |
|---|---|---|---|---|---|
| Week 1 | *Lecture 1*<br>*Exercise 1* - out | - | - | - | Help session<br>*Exercise 1* |
| Week 2 | *Lecture 2*<br>*Exercise 2* - out | *Exercise 1* - in | - | - | Help session<br>*Exercise 2* |
| Week 3 | *Lecture 3*<br>*Exercise 3* - out | *Exercise 2* - in<br>*Exercise 1*<br>grades out | - | One-to-one feedback<br>session *Exercise 1* | Help session<br>*Exercise 3* |

### 4.2 Evaluation/student learning assessment

The examination form is 70% the exercises and 30% the oral exam on the theoretical background. This is well received by the students, who perceive it as fair due to the large amount of work required to code the programs and to solve the problems.

During the course, the students were encouraged to work together and help each other in understanding the problems, but were required to hand in the exercises individually. The assignments are evaluated following a systematic grading scheme, and formal written feedback on how the students solved the problems is given individually. In addition to this summative assessment, we also have one-to-one formative feedback sessions after they get the written feedback. Here, we give them a more individual environment to them to ask questions about the concepts taught in class and the coding. Despite the increase in the workload, these one-to-one meetings are important to understand how the student learning is evolving and if they are absorbing the contents taught in class.

In the final oral examination (30% of final grade), the students were asked generic questions about the approaches, as well as specific questions to the exercises that they had handed in. This made it possible to assess the students understanding of the topic in general, but also of their own code.

## 5. EVOLUTION AND CONTINUOUS IMPROVEMENT OF THIS COURSE

This course has evolved based on the feedback of the students. Here is a timeline showing the course improvements. The student feedback is shown in Section 6.

### 5.1 2019 (Short course at the University of São Paulo)

This course was first taught as an invited short course (1 week, Masters/PhD level) at the Chemical Engineering Department of the University of Sao Paulo, Brazil. In the first version, the exercises were done in the Julia programming language (Bezanson et al., 2017). The course concept was to give a minimum necessary content of theory, and then let the students work on implementing their own controller software. The students were given coding exercises that were used to practice the approaches. For the final evaluation, the students worked in pairs on a project. The goals was to apply and compare two of the approaches considered in the course on a larger case study of their choice.

The course was well received, and 12 of 14 students successfully completed it. This is very good, since the course required a significant amount of work and was not mandatory. It was communicated to us orally that the students considered it useful for their work on their masters and PhD theses. For example, one of the projects developed in this course led to a journal publication, see de Oliveira et al. (2021).

It was also communicated to us that some students struggled with learning a new programming language (Julia) at the same time as learning complex topics from dynamic optimization and model predictive control. That is, the activities and the learning outcomes were not aligned constructively, as Julia was not the main topic of this course.

### 5.2 2019 (Regular module at NTNU)

When we ran the course for the first time at NTNU as a normal course module (1/2 semester), we chose to change some aspects to help the students focus on the essential part of the course: methods for setting up and solving a Nonlinear Model Predictive Controller. The first modification was related to the programming language. We changed from Julia to MATLAB/CasADi, a language that students in our department were more familiar with and that had a better documentation at that point. This allowed us to align learning activities and the learning outcomes. As a consequence, the students were able to quickly focus on the control part (instead of being slowed down by "cryptic" error messages in Julia). We also changed the assessment from a single graded final project to a different form, where the exercises count 70% of the final grade and an oral exam with the remaining 30%.

### 5.3 2020 (Regular module at NTNU)

In 2020, we added one-to-one formative feedback sessions as described in Section 4.2. This one-to-one sessions helped foster the students' motivation to really understand the code that they handed in. Otherwise, in these sessions, it would become quite obvious if they had simply copied someone else's code, or did not understand what they did. Additionally, at the end of the course, we have sent out a questionnaire to receive formal feedback on the course.

### 5.4 2021 (Regular module at NTNU)

Based on the feedback forms, we found that the students requested an earlier exam date and that the workload of the assignments was considered heavy. This comes from the students coding their dynamic optimization methods from scratch.

We adjust the slides and contents of some of the lectures to solve this problem; the idea was to focus more on issues that the students who took the course in the previous years had regarding the code. For example, in *Lecture 6*, where we teach orthogonal collocation, the presentation style relied on key concepts rather than on showing how to code the resulting system of equations in MATLAB/CasADi. We adapted the slides as well as the notation to make a parallel between the theory and the coding.

**Remark:** *Both in 2020 and 2021 the course was held in person. Since the classes were small (13 and 4 students, respectively) and the pandemic was relatively under control in Norway, we could safely comply with all NTNU Corona regulations.*

## 6. STUDENT FEEDBACK

The feedback questionnaire is shown in Table 2. It was anonymous and composed by one qualitative question, ten quantitative questions, and one open-ended question. The idea was to evaluate the workload, the students' perception

Table 2. Feedback Questionnaire

| | Question | Possible responses | 2020 (average) | 2021 (average) |
|---|---|---|---|---|
| 1 | The amount of work required for this course was: | Adequate — Too much — Too low | See Figure 3 | |
| 2 | To what degree did the lectures prepare you for the exercise work, the exam, and reaching the learning goals | Answer on the range of 1 (the lectures did not really prepare me for the exercise and the exam) to 5 (the lectures prepared me well for the exercises and the exam) | 3.92 | 4.33 |
| 3 | To what degree did the exercises help me deepen my understanding of the topics? | Answer on the range of 1 (very little degree) to 5 (very high degree) | 4.83 | 4.67 |
| 4 | It was clearly communicated to me what was expected from me in the exercises and the exam. | Answer on the range of 1 (disagree) to 5 (strongly agree) | 4.17 | 4.67 |
| 5 | How do you rate the feedback on the exercises? | Answer on the range of 1 (not useful) to 5 (it helped me improve a lot) | 4.67 | 4.33 |
| 6 | I got the help I needed to do the required tasks and understand the material | Answer on the range of 1 (I disagree) to 5 (I fully agree) | 4.67 | 4.67 |
| 7 | I feel that I achieved the requirement learning outcomes | Answer on the range of 1 (I disagree) to 5 (I fully agree) | 4.75 | 4.33 |
| 8 | I would recommend this course to other students | Answer on the range of 1 (strongly disagree) to 5 (strongly agree) | 4.83 | 4.33 |
| 9 | The quality of the course has met my expectations | Answer on the range of 1 (strongly disagree) to 5 (strongly agree, exceeded my expectations) | 4.67 | 4.67 |
| 10 | The guest lectures helped me to put the learned material into context | Answer on the range of 1 (didn't attend, N/A), 2 (strongly disagree) to 6 (strongly agree) | 4.25 | 3 |
| 11 | Other comments and feedback. Is there something we should do differently next year? (If you want to, you can also let us know your name, then we can follow up) | Open-ended question | See Table 3 | |

about the alignment between the learning activities and the lectures, and also their opinion in the effectiveness of the feedback.

Note that the number of students changed drastically from 2020 to 2021. Hence, it is expected that the numbers fluctuate due to the small sample in 2021. With that disclaimer, the results in the table can be useful to guide us for our thinking when offering the course again. In general, the evaluations are positive and consistent between the two consecutive years.

Regarding the practical part of the course, we want to highlight *Question 3*. Here, we see that the exercises really add value to the course, helping the students understand the topics. *Question 2* shows that the adaptation of the slides in 2021 had a positive impact on preparing the students for the exercise.

However, based on the open-ended question (see Table 3), the students pointed out repeatedly that the course load is heavy. Even though we promoted some changes in the material that were well received, the students did not feel that the overall workload was lighter. This should be the key aspect for improvement in 2022, when the course will be offered again.

Another point of interest is the low score given to the guest lecture in 2021. In 2020, we had two guest lectures that discussed practical aspects of linear and nonlinear MPC implementations, which directly correlates with the course. In turn, in 2021, we had a different lecturer that focused more on PID controllers. Although interesting discussion about practical implementation took place during the lecture, it was slightly out of context from the course. Consequently, the students justifiably gave a low grade in the feedback questionnaire, despite the fact that the lecture quality was high.

In addition to the formal feedback, colleagues have informally confirmed to us that they can do more interesting projects with their students, because the students are able to understand and implement their own model predictive control routines. This comes from doing the implementation as part of this course. Also, as mentioned, several students have been hired to work in IT companies. This shows that the programming and automation skills learned in this course are of value beyond the field of chemical engineering and process control.

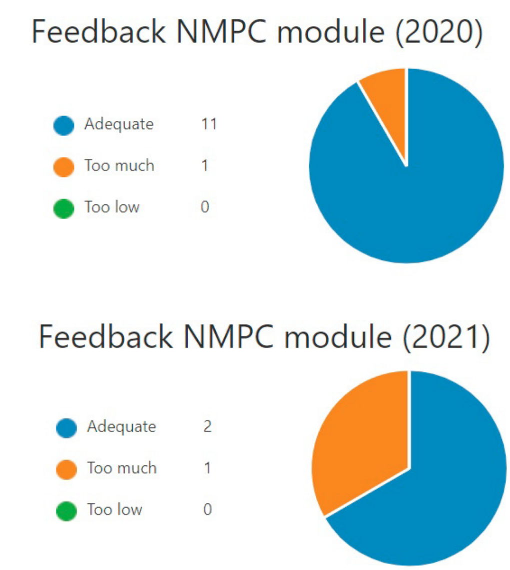


Fig. 3. Feedback questionnaire: Qualitative question response (Question 1)

## 7. CONCLUSION

In this course, we present topics related to state-of-the-art research on Nonlinear Model Predictive Control (NMPC), but also try to cover topics of industrial practice. The course was designed based on problem solving principles, where we used the lectures for conveying only the main information regarding the implementation and solution of NMPC, whereas the students work on assignments to gain knowledge and skills related to the topic.

The course was created to attend four needs that the students of our department faced: (1) lack of knowledge about state-of-the-art methods for process automation, (2) a course about these methods tailored for their needs as Chemical/Process engineers; (3) improving their programming skills; and (4) a deeper understanding of advanced control applied to biochemical systems.

Table 3. Feedback questionnaire: Open-ended responses (Question 11)

| **Feedback (selected)** |
|---|
| *2021* |
| The only things I want to emphasize is that I think the workload for this module is way to much, with the module only being 3.75 spt. |
| *2020* |
| Excellent course. The lectures were structured well, understandable and had a clear connection to the exercises. One thing to consider would be to move the exam closer to the end of lectures, instead of in December. |
| It is quite difficult to use Casadi in the Single Shooting exercise since it is quite different from MATLAB. Maybe we should have a lecture on how to use Casadi (basic syntax, etc.)? |
| Earlier exam date (closer to last exercise) |
| I enjoyed the practical part of the exercises, and it really helped the course just being a lot of a dump of theory. Also, while I put down adequate as response for exercises, along with the other module, the workload could become a tad bit much, since these exercises and the course lectures happen in a relatively short timespan. Especially since matlab is relatively new for me, learning the details of that along the way added some difficulty. So be careful not to shorten down the time to work on those exercises. I want to praise the lectures and learning material, they were clear and understandable, and going back through the lecture notes later is not a problem. NMPC was the most enjoyable class this semester, and overall I find the module quality above most other courses I have taken in my 4th and 5th year. |

We believe that the course created satisfied these needs. First, the students that take the course learn how to solve dynamic optimization problems using different state-of-the-art methods with a focus on real-time control applications (*need 1*). The control aspects taught in the course are more related to a chemical/process engineering perspective rather than discussing a Engineering Cybernetics aspects. For example, the assignments center on practical implementation aspects rather than stability proofs (*need 2*). To complete them, the students are required to develop their own MATLAB scripts and understand how to code abstract concepts with continuous guidance and feedback from the the lecturer and the teaching assistant team (*need 3*). Finally, despite presenting several examples of applications in chemical engineering, the exercises are based on the control of a biochemical system (*need 4*). However, we still want to cover specific aspects of bioprocess control the next time we offer this course.

Regarding the pedagogical aspects, we tried to design student activities (exercises) that give the students the opportunity to become active in learning. For example, by solving problems (e.g. setting up and solving the dynamic optimization problem), the student needs to think about how the basic principles apply to this particular case. In face of complex and challenging coding exercises, the students become engaged in the course as well as obtain a deeper understanding of the tools presented during the lectures. According to the feedback received, this approach has been well received by the students. They use the concepts learned during the course in their Master´s thesis. Also, the set of skills developed while solving the exercises is useful beyond the context of their academic life.

We believe that the same active learning principles can be applied to more traditional control courses. It is possible to transform the formal lectures into more engaging project-based and challenge-based learning. For example, in classical control courses, learning topics such as Bode and Nyquist plots can be extremely overwhelming for the students. Questions such as "Why are we learning this?" or "When will we use this in the practice?" can easily appear. Based on our experience developing this course, the assimilation by the students of these complex topics can be facilitate if we focus on the application first. For instance, we could start by discussing robustness of the control design, highlighting why gain and phase margins are important as "safety nets" since we rarely know our system perfectly. Then, present the Bode plot as a tool for solving the problem. Finally, we tie it all together with careful designed exercises, where the students learn to solve the problem (here, control design robustness) under our guidance and mentoring. According to our experience, the students will come up with their own questions while solving the exercises and clearly see the meaning of the concepts presented to them, which, we believe, facilitates their own learning as a whole.

## REFERENCES


Andersson, J., Gillis, J., Horn, G., Rawlings, J.B., and Diehl, M. (2019). CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1), 1–36.

Bequette, B.W. (2003). *Process control: modeling, design, and simulation.* Prentice Hall Professional.

Bezanson, J., Edelman, A., Karpinski, S., and Shah, V.B. (2017). Julia: A fresh approach to numerical computing. *SIAM review*, 59(1), 65–98. URL `https://doi.org/10.1137/141000671`.

Biegler, L.T. (2018). New directions for nonlinear process optimization. *Current Opinion in Chemical Engineering*, 21, 32–40.

Biegler, L.T. (2010). *Nonlinear programming: concepts, algorithms, and applications to chemical processes.* SIAM.

Chen, C.Y. and Joseph, B. (1987). On-line optimization using a two-phase approach: An application study. *Industrial & Engineering Chemistry Research*, 26(9), 1924–1930.

de Oliveira, R.D., Guedes, M.N., Matias, J., and Le Roux, G.A. (2021). Nonlinear predictive control of a bioreactor by surrogate model approximation of flux balance analysis. *Industrial & Engineering Chemistry Research*, 60(40), 14464–14475.

Pistikopoulos, E.N., Barbosa-Povoa, A., Lee, J.H., Misener, R., Mitsos, A., Reklaitis, G.V., Venkatasubramanian, V., You, F., and Gani, R. (2021). Process systems engineering–the generation next? *Computers & Chemical Engineering*, 107252.

Rawlings, J.B., Patel, N.R., Risbeck, M.J., Maravelias, C.T., Wenzel, M.J., and Turney, R.D. (2018). Economic mpc and real-time decision making with application to large-scale hvac energy systems. *Computers & Chemical Engineering*, 114, 89–98.

Rossiter, J.A., Zakova, K., Huba, M., Serbezov, A., and Visioli, A. (2019). A first course in feedback, dynamics and control: findings from an online pilot survey for the IFAC community. *IFAC-PapersOnLine*, 52(9), 298–305.

Varma, A. and Grossmann, I.E. (2014). Evolving trends in chemical engineering education. *AIChE Journal*, 60(11), 3692–3700.

Wächter, A. and Biegler, L.T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1), 25–57.