

# Teaching Predictive Control Using Specification-based Summative Assessments<sup>\*,\*\*</sup>

Ian McInerney<sup>\*</sup> Eric C. Kerrigan<sup>\*,\*\*</sup>

<sup>\*</sup> *Department of Electrical & Electronic Engineering, Imperial College  
London, SW7 2AZ London, UK, email:  
{i.mcinerney17,e.kerrigan}@imperial.ac.uk*

<sup>\*\*</sup> *Department of Aeronautics, Imperial College London, SW7 2AZ  
London, UK*

---

**Abstract:** Including Model Predictive Control (MPC) in the undergraduate/graduate control curriculum is becoming vitally important due to the growing adoption of MPC in many industrial areas. In this paper, we present an overview of the predictive control course taught by the authors at Imperial College London between 2018 and 2021. We discuss how the course evolved from focusing solely on the linear MPC formulation to covering nonlinear MPC and some of its extensions. We also present a novel specification-based summative assessment framework, written in MATLAB, that was developed to assess the knowledge and understanding of the students in the course by tasking them with designing a controller for a real-world problem. The MATLAB assessment framework was designed to provide the students with the freedom to design and implement any MPC controller they wanted. The submitted controllers were then assessed against over 30 variations of the real-world problem to gauge student understanding of design robustness and the MPC topics from the course.

*Keywords:* predictive control, virtual laboratory, summative assessments

---

## 1. INTRODUCTION

Model Predictive Control (MPC) is an advanced control method that has found widespread use in many industrial areas, including aerospace, chemical processes, power electronics, and autonomous vehicles. The industrial success of MPC in these areas is due to several factors, including

- (i) the ability to incorporate complex systems and constraints in the controller,
- (ii) a simple and cross-disciplinary conceptual idea of the controller, and
- (iii) increased tooling and support for the design and implementation of the controller.

Factors (i) and (ii) occur because MPC controllers are written as an optimization problem that seeks to minimize an objective subject to constraints that enforce the system dynamics model and the constraints from the designer. Switching between different applications is then conceptually as simple as changing the dynamics model, constraints and the objective in the optimization problem. Factor (iii) then translates this conceptual simplicity into practice, with software tools such as FORCESPRO or MATLAB's Model Predictive Control Toolbox translating the high-level optimization problem into a working controller.

---

<sup>\*</sup> This work has been approved by the Imperial College London Education Ethics Review Process (Study number EERP2122-029).

<sup>\*\*</sup>Supported by the EPSRC Centre for Doctoral Training in High Performance Embedded and Distributed Systems (HiPEDS, Grant Reference EP/L016796/1) and MathWorks.

The growing use of MPC in industry is occurring alongside a shift in the expectations for university control courses. Rossiter et al. (2019) reported that a recent survey asking how a first course on control should be designed showed that respondents from industry ranked topics such as optimal control and optimal state feedback in the top 10 concepts to include, which was higher than topics such as PID and lead-lag controllers. In addition, industry respondents felt that pedagogical techniques such as assessments focusing on concepts and demonstrating control design using authentic simulation/implementation scenarios were as important as assessments that were focused on just the mathematical concepts/theory.

Designing a course covering MPC is not a simple task due to the need to balance the large set of topics that could be covered (e.g. stochastic, robust, nonlinear, economic) and the depth of coverage (e.g. optimization theory, stability, problem formulations). Recently, Faulwasser et al. (2021) discussed where and how such a course on MPC can fit into the curriculum, and concluded that a first course in MPC could fit into the 2nd or 3rd year of a Bachelor's curriculum and cover an introduction to the linear-quadratic problem formulation, numerical optimization, stability, and highlight application areas. Further topics such as nonlinear, distributed or economic MPC could then be covered in a graduate-level course along with a more in-depth discussion of the underlying theory.

One possible course for undergraduates was described by Honc et al. (2016), with the course focusing on teaching

Table 1. Schedule of topics covered in the courses

Week	2018, 2019, and 2020 Courses	2021 Course
1	Introduction to predictive control	Introduction to predictive control
2	Predicting the future	State-space modelling and differential equation solvers
3	Unconstrained LQR RHC	Numerical/automatic differentiation & Discretization methods
4	Constrained LQR RHC	Constrained LQR RHC
5	Soft constraints & Setpoint tracking	Soft constraints & Max-type costs and constraints
6	Setpoint tracking & Disturbance rejection	Robustness & Constraint tightening
7	Disturbance rejection	Closed-loop stability and recursive feasibility
8	Rate constraints & Move blocking	The real-time iteration scheme
9	Stability & Robustness	Move blocking
10	Final project (2018, 2019)/NMPC direct collocation (2020)	External constraint handling methods

the fundamentals of linear MPC and other topics such as model derivation, controller tuning and offset-free control, with a final project of applying MPC to the control of the water level in a tank. A slightly more advanced course for Masters-level students was described by Keller et al. (2020), and consisted of both linear and nonlinear MPC and related topics such as optimization theory, discretization methods, and stability theory. At the end of the course, the students were tasked with designing a nonlinear MPC controller for a diesel engine and implementing it in real-time on a laboratory test-bench.

In this paper, we describe the predictive control course taught at Imperial College London between 2018 and 2021 to Masters-level (MSc and final year MEng) students. The majority of students will have completed an introductory control course on state space and transfer function methods (whose content will vary depending on their undergraduate institution), and may not have taken any prior courses on optimization. This course has evolved over the four years, starting with only linear-quadratic MPC and some extensions in 2018 and transitioning to nonlinear MPC and its dependencies in 2021. The course includes laboratory exercises utilizing a laboratory-scale gantry crane, and also uses MATLAB Grader to provide checkpointing assessments for the students during the course. Instead of utilizing a written/exam-based summative assessment at the end of the course, we developed a specification-based controller design problem that allows for a more thorough assessment of the student’s knowledge and understanding of MPC.

In these summative assessments, the student is given only the specification describing a real-world problem, which is moving an overhead gantry crane to a target point in a limited time while avoiding obstacles. We developed a MATLAB/Simulink framework (initially described by the authors in McInerney and Kerrigan (2018)) that provides a closed-loop simulation environment where the students write their own MATLAB functions to implement the target generator, state estimator and controller. The student controllers are automatically tested on a set of over 30 variations of the real-world problem that are generated by changing the obstacles/constraints and adding uncertainty to the simulation model of the gantry crane.

## 2. COURSE LECTURES

The course was taught over a 10-week period each year, with one 2-hour long lecture session per week. The topics covered in the courses can be seen in Table 1.

The 2018 and 2019 courses focused exclusively on linear MPC, with the 2020 course adding a 1-lecture introduction to direct collocation-based nonlinear MPC in week 10. The 2021 course was redesigned to primarily focus on nonlinear MPC and its associated prerequisites, and contained only a 1-lecture overview of linear MPC.

### 2.1 2018, 2019 & 2020 courses

The main focus of the course for 2018–2020 was to teach how to apply linear predictive control to a system by using the Linear Quadratic Regulator (LQR) formulation of MPC as a Receding Horizon Controller (RHC). Notably, this course did not contain any lectures dedicated to optimization theory or optimization solvers, with the students instructed to instead use the *mpcqp solver* function when implementing the MPC controllers in the course.

Instead of utilizing a singular textbook for the course, the students were given a reference list containing approximately 30 books and papers that covered the material taught in the course.

*Course topics* The course was split into three parts, with the basics of linear MPC in the first, several extensions to linear MPC in the second, and more advanced topics in the third. The first part began by introducing feedback control and the idea of predicting the future trajectory of a system. Then the unconstrained finite-horizon LQR formulation was introduced along with the algorithms to construct the prediction matrix and receding-horizon state feedback controller matrix. Finally, both the uncondensed and condensed constrained LQR formulations were introduced, along with the algorithms to construct the appropriate constraint matrices and Hessians.

In the second part, the students were introduced to more advanced linear MPC concepts taken from various research papers in the MPC field, such as soft constraints (Sckaert and Rawlings, 1999), offset-free control (Pannocchia et al., 2015), and move blocking (Cagienard et al., 2007).

The final part of the course contained more advanced topics that were not contained in either the checkpointing or summative assessments. In the 2018/2019 courses, the only advanced topic covered was stability theory for linear MPC, based on Mayne et al. (2000). In the 2020 course, an additional advanced topic introducing direct collocation-based nonlinear MPC was added based on the tutorial paper by Kelly (2017).

*Laboratory exercises* The course contained two physical laboratory activities that the students performed during

weeks 3/4 and 5/6. These activities used the INTECO 3D overhead gantry crane (INTECO, 2014) connected to a computer running MATLAB/Simulink.

During the first laboratory, the students were given an unconstrained LQR controller that moved the crane from a starting point to a target point. The students were asked to modify the various controller parameters (i.e. cost matrices and horizon length) to see how they affect the closed loop system response and gain an intuition about the tuning of the controller.

In the second laboratory, the students were given a framework with a constrained linear MPC controller. The students were asked to modify the cost matrices, horizon length, and constraints (i.e. tighten/loosen them) to gauge their effect on the controller response. In this laboratory, the students were also introduced to the idea of computational complexity of the controller, with the framework recording the time used by the optimization solver at each sample so that the students could see the effect the controller changes had on the computations needed.

## 2.2 2021 course

The 2021 course was redesigned to focus on nonlinear MPC and utilized the textbook by Rawlings et al. (2020) as a reference. The focus of this course was still on applying MPC, however the switch to nonlinear MPC also required the introduction of more background material to the course content.

In addition to the switch to nonlinear MPC, the COVID-19 pandemic necessitated the course be redesigned to be a virtual course with one weekly 1-hour video lecture slot. This was done by using a flipped-classroom approach, where the students watched video lectures on each topic before attending the weekly session, and then participated in activities/short quizzes during the weekly session.

*Course topics* This course consisted of two parts, the first covering the necessary background material to formulate a nonlinear MPC problem and the second covering various formulations/extensions of nonlinear MPC. The background material covered in the first part included numerical differential equation solvers (e.g. Runge-Kutta methods, collocation methods), discretization methods for nonlinear state-space equations (e.g. single/multiple shooting, direct collocation), and computing derivatives (i.e. numerical and automatic differentiation).

In the second part of the course, the students were introduced to extensions of the nonlinear MPC formulation. These included soft constraints, constraint tightening and robustness (Saltik et al., 2018), the real-time iteration (Gros et al., 2020), move blocking (Chen et al., 2020), and external constraint handling (Nie and Kerrigan, 2020).

*Laboratory exercises* Due to the COVID-19 pandemic, the course could not use the in-person laboratory equipment, so a new virtual laboratory was developed. The students were given a high-fidelity simulation model of the crane in feedback with a constrained linear MPC controller and were asked to perform the same experiments as the previous-year’s in-person laboratories (i.e. modifying controller parameters to see the closed-loop response).

Table 2. Checkpointing assessments

Topic	Problem
<b>Modelling</b>	Gantry crane model derivation
<b>Unconstrained RHC</b>	Prediction matrix construction
	Cost function matrix construction
	Linear RHC law construction
<b>Constrained RHC</b>	Stage constraint matrix
	Trajectory constraint matrix
	QP constraint matrix
	Receding horizon controller
	Soft constraint matrices
<b>Differential equations<sup>1</sup></b>	Runge-Kutta Methods
	Implicit/Explicit Euler
	Writing an ODE45 method
<b>Linear equations<sup>1</sup></b>	Solution uniqueness
	Solving linear systems
	Least squares method
<b>Quadrature methods<sup>1</sup></b>	Riemann sums
	Trapezoidal method
	Simpson’s rule
<b>Numerical optimization<sup>1</sup></b>	fmincon w/ manual differentiation
	fmincon w/ auto differentiation

<sup>1</sup> Only in 2021 course

This virtual laboratory was implemented inside a MATLAB Live Script that could be run using MATLAB Online, allowing for students to do the laboratory without needing to have MATLAB installed on their computers. The use of the Live Script also allowed the instructions and formatted equations to be included in the same file as the code, allowing the students to more easily understand the lab.

## 3. CHECKPOINTING ASSESSMENTS

During the course, the students were assigned several small summative assessments on MATLAB Grader to gauge their progress and understanding of the topics being taught. In the 2018–2020 courses, there were three main topics covered in the assessments: system modelling, unconstrained RHC, and constrained RHC. In the 2021 course, four new topics were added to cover the new background material needed for nonlinear MPC: differential equation solvers, linear equations, quadrature methods, and numerical optimization with *fmincon*.

Inside each topic, the student was presented with several MATLAB coding problems to test their knowledge on how to implement ideas discussed in the lectures (see Table 2 for the problems in each topic). For example, in the prediction matrix problem in the Unconstrained RHC topic, the students were asked to write a MATLAB function that took the system’s state-space matrices and desired horizon length as the input and returned the fully formed prediction matrix. The functions the students wrote were then tested against several sets of function inputs to ensure their function worked and was generalizable to other problems.

## 4. SPECIFICATION-BASED SUMMATIVE ASSESSMENT

To assess the learning of the students in the course, we utilized two controller design summative assessments instead of the traditional end-of-course written exam. The first of the assessments was due roughly 50–60% of the way through the course, and tested the student’s

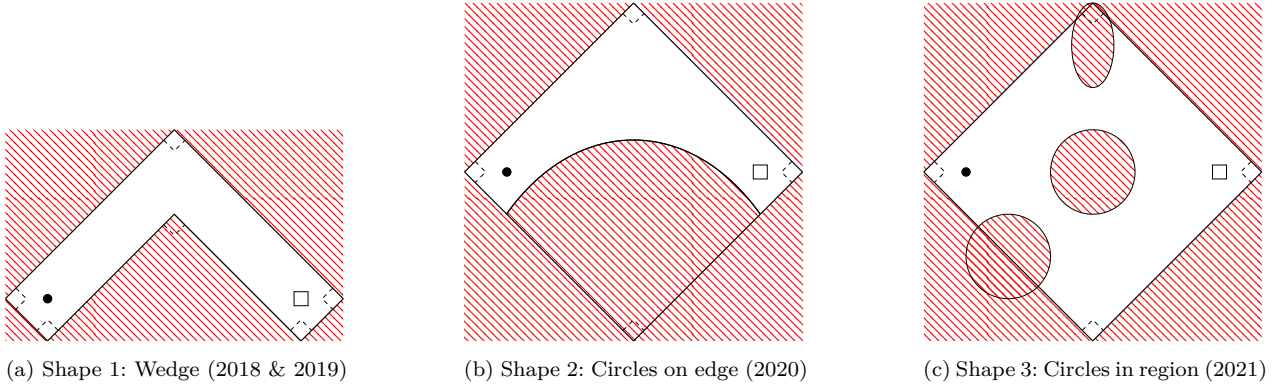


Fig. 1. Sample shape the crane must stay inside. (The allowed region is in white, forbidden region in hatched red, the starting point is the filled circle and the target point is the empty square).

knowledge of the basic predictive control concepts. The second assessment was due at the end of the last week of term and focused on challenging the student to try the more advanced concepts taught in part 2 of the course.

#### 4.1 Assessment Overview

In the controller design assessments, the students were tasked with designing a controller to move a gantry crane from a starting point to a target point while staying inside a specified region and navigating around obstacles. For the first assessment, the constrained region was a simple rectangle (e.g. the left half of the wedge in Figure 1a).

The constrained region for the second assessment evolved during the four years of the course. In the 2018 and 2019 courses, the constrained region was the intersection of two rectangles at a  $90^\circ$  angle to form a wedge, as shown in Figure 1a. In the 2020 course, the region was changed to be a base shape of a single rectangle with circular regions centered on the edge of the rectangle added as obstacles for the crane to avoid, as shown in Figure 1b. For the 2021 course, the shape was formed by a single rectangular region with up to 10 elliptical obstacles placed anywhere in the region, with an example region shown in Figure 1c.

For the 2018–2020 courses, the objective of the controller was to simply move the crane to the target point within a specified time limit and without violating any of the constraints. With the switch to nonlinear MPC in the 2021 course, the assessment was updated with two major changes: (i) turning the control design problem into a fixed final time problem (i.e. the crane needed to be at the target point exactly at the specified time), and (ii) the addition of a constraint that limited the amount of total work the controller could perform when moving the crane.

*Specification* In the assessment document, the students were only given a set of formal performance specifications that their controller must meet, and were then free to implement any MPC formulation they wished. These performance specifications consisted of two parts: the equilibrium condition for the crane at the target point and the definition of successful completion of a testcase, shown in Definitions 1 and 2, respectively, for the 2021 course.

*Definition 1.* (Equilibrium). Obtaining equilibrium in the simulation run means that:

- the  $x$  and  $y$  position states of the cart are within  $\epsilon_t$  of the target point at  $t = T_f$  seconds,
- the  $x$  and  $y$  position states of the payload are within  $\epsilon_t$  of the target point at  $t = T_f$  seconds,
- the velocity of the cart ( $\dot{x}$  and  $\dot{y}$ ) and angular velocity of the pendulum ( $\dot{\theta}$  and  $\dot{\psi}$ ) are within  $\epsilon_r$  of 0 at  $t = T_f$  seconds, and
- the inputs are within  $\epsilon_r$  of 0 at  $t = T_f$  seconds.

All comparisons are made using the infinity norm.

*Definition 2.* (Successful Completion). Successful completion of a testcase means that:

- the system is at equilibrium (as defined in Definition 1) at  $t = T_f$  seconds,
- the inputs remain in the interval  $[-1, 1]$  during the entire time interval  $[0, T_f]$ ,
- the work done by the cart over  $T_f$  seconds is not more than  $W_{max}$ , i.e.  $W \leq W_{max}$ , and
- no constraints are violated over the time interval  $[0, T_f]$ .

*Marking* The submitted controller designs were evaluated using both the laboratory gantry crane hardware and a high-fidelity simulation model of the gantry crane in MATLAB, and were tested against over 30 region shapes/sizes and sets of obstacles to gauge the generalizability of the student controller.

When testing against the simulation model, the students were only given one of the testcases used to mark the controllers (the default shape testcase) before the submission deadline. The remaining testcases were kept secret, and only given to the students along with their results. The secret testcases were designed to expose the controllers to a variety of situations, and were generated by

- narrowing/widening the constrained region,
- moving the target point closer to the constraints,
- adding more elliptical obstacles, and
- perturbing the model parameters away from the nominal model.

To ensure fairness in marking, every student controller was tested using the same set of secret testcases.

When evaluating the controller using the hardware gantry crane, each student was given a 30 minute timeslot on the actual hardware to experiment and tune their submitted

Table 3. Number of students choosing each design option.

(a) 2018 course (out of 30 students)

Cost Function	
Quadratic cost	30
Stabilizing terminal penalty	15
Constraints	
Soft constraints	14
Multiple constraint sets	29
State Estimator	
Kalman filter	15
Other state estimator	5
Other Features	
Offset-free tracking	18
Move blocking	1

(b) 2021 course (out of 26 students)

Setup - Path Planning		Controller - Optimizer	
MATLAB <i>nlmpc</i>	2	MATLAB <i>nlmpc</i>	4
MATLAB <i>fmincon</i>	4	MATLAB <i>fmincon</i>	20
Other path planning	3	Real-time iteration	3
Setup - Nonlinearities		Controller - Model	
Nonlinear cost	2	Nonlinear/Time-varying	9
Nonlinear ellipses	9	Linear	15
Controller - Other Features		Controller - Cost	
Constraint tightening	15	Nonlinear	5
Soft constraints	4	Quadratic	23
State estimator	5		

controller. Because the hardware gantry crane already includes unmodelled elements/disturbances, no additional disturbances/uncertainties were added, and the controllers were only marked using the default shape testcase.

The students were required to write a short (1–2 page) report on the MPC controller they implemented. They also underwent an oral exam, where they were asked about their controller design and how it met the specifications given in the assessment document.

#### 4.2 Assessment Framework

To administer the specification-based summative assessment, we developed a MATLAB/Simulink framework that allowed the students to write four MATLAB functions containing their solution: *mySetup*, *myMPCController*, *myTargetGenerator*, and *myStateEstimator*. These functions were submitted through MATLAB Grader, which also ran preliminary tests to ensure the submitted code had no syntax errors.

The marking framework iterated through all the testcases, running a closed-loop simulation of the crane system for each testcase, and then saving the resulting state and control trajectories for later analysis. Each of the student functions implemented a specific component of the closed-loop system, with the *mySetup* function running before the simulation began to allow offline computation of variables that were used in the other three controller functions.

After all student controllers were run on every testcase, the marking framework then generated marks for the students by comparing the saved trajectories against the given specification and determining any violations that occurred. These violations were then plugged into a marking rubric to turn the testcase results into actual course marks. The students then received a report containing plots of all the trajectories and a listing of all the specification violations that occurred.

## 5. STUDENT SOLUTIONS AND OBSERVATIONS

Overall, an analysis of the student controllers submitted for the final assessment in the course shows that the specification-based summative assessment framework provided a large degree of freedom to the students. This can be seen in Table 3, where we show the different design options chosen by students in both the 2018 and 2021 courses.

In 2018, the students were limited to using only the *mpcqpoptimizer* optimization function (due to limitations in MATLAB’s real-time code generation), meaning all 30 students implemented a quadratic MPC formulation that simply changed the constraint sets in the optimizer as the crane moved around the wedge. However, there was more diversity in the other design choices, with half the students choosing to implement soft constraints, another half implementing state estimation and offset-free tracking, and one student choosing to implement move blocking.

In the 2021 course, when there was no prescribed optimizer to use, the students chose to implement a diverse set of controllers, with the majority using *fmincon*, but with 4 using the built-in MATLAB Model Predictive Control Toolbox *nlmpc* function, and another 3 implementing a custom real-time iteration scheme. Additionally, the framework provided the freedom to implement path planning to avoid the obstacles, which 9 students chose to do (with 3 of them implementing an ASTAR-based path planner).

The majority of the students in 2021 only utilized the *fmincon* optimizer to add the nonlinear elliptical constraints to the optimization problem, and still utilized a linear dynamics model and a quadratic cost function. To add robustness to their designs, the 2021 cohort utilized constraint tightening more than soft constraints, and relatively few implemented state estimation.

Based on our observations, the students engaged with and enjoyed the in-person lab components of the course, but sometimes spent too much time completing the final summative assessments. This appears to be a result of having a lot of freedom in the design, which results in the students continually trying out new and more advanced methods (and having to spend time debugging them), only to see marginal improvements in the performance of their controller.

## 6. LESSONS LEARNED

During the past four years, we have encountered several issues and problems when using the specification-based summative assessments in the course, with the two largest issues being: ambiguity in the specification and errors in the student code submissions.

## 6.1 Students find ambiguities/loopholes

The largest issue we faced was properly and completely defining the specification given to the students inside the assessment document so that it contained what we wanted to assess. When drafting the specification, we fell into the trap of including our implicit assumptions on what the controller should do when we interpreted the specification instead of interpreting it as written. For example, in the 2018–2020 courses the definition of equilibrium contained conditions similar to

The  $x$  and  $y$  position states of the cart are within  $\epsilon_t$  of the target point within 5 seconds,

which as written means the student controller only needed to be at the target point satisfying all the conditions for a single time instant to meet the specification. In reality, we were implicitly thinking the student controllers should enter and stay within  $\epsilon_t$  of the target.

These types of errors in the specification are the hardest to fix, since the assessment criteria could not be modified once given to the students. Instead, we modified the analysis framework to match the given specification, and in the 2021 course redesigned the assessment to be a fixed final time problem, removing this ambiguity.

## 6.2 Don't trust student code

While the framework was designed to be an automated system that could simply be started and then left to run, during our initial use of it we encountered many unexpected errors that crashed the framework. These errors mainly come from the student code, and include: incorrect computations inside the controller function causing MATLAB to throw an error, the student controller giving out-of-bound or invalid values that then cause the dynamical simulation to error, and internal errors in the numerical solvers that then crash the entire MATLAB process. In response, we implemented four levels of error catching/handling in the framework to catch and gracefully handle any errors that occurred, including a command line script to monitor and restart MATLAB if it crashed.

## 7. CONCLUSIONS

In this paper, we presented the predictive control course taught by the authors over the past 4 years at Imperial College London. This course has evolved from teaching only linear MPC in 2018 to focusing on nonlinear MPC in 2021, and assesses student knowledge of the predictive control concepts using a novel specification-based summative assessment framework. This framework gives the students freedom in their controller design to implement different MPC formulations, while also encouraging them to think about the robustness of their controller. While we observed an increase in the different types of controllers implemented by the students in the 2021 course when the focus changed to nonlinear MPC, there was still a tendency to utilize a linear model and quadratic cost instead of exploring more advanced concepts. Future work and updates to the course could explore ways to push students more towards the non-LQR formulations,

possibly by introducing a different and more nonlinear system or changing the objective of the controller.

## REFERENCES

- Cagienard, R., Grieder, P., Kerrigan, E.C., and Morari, M. (2007). Move blocking strategies in receding horizon control. *Journal of Process Control*, 17(6), 563–570.
- Chen, Y., Scarabottolo, N., Bruschetta, M., and Beghi, A. (2020). Efficient move blocking strategy for multiple shooting-based non-linear model predictive control. *IET Control Theory & Applications*, 14(2), 343–351.
- Faulwasser, T., Lucia, S., Schulze Darup, M., and Mönnigmann, M. (2021). Teaching MPC: Which Way to the Promised Land? *IFAC-PapersOnLine*, 54(6), 238–243.
- Gros, S., Zanon, M., Quirynen, R., Bemporad, A., and Diehl, M. (2020). From linear to nonlinear MPC: Bridging the gap via the real-time iteration. *International Journal of Control*, 93(1), 1–19.
- Honc, D., Sharma K., R., Abraham, A., Dušek, F., and Pappa, N. (2016). Teaching and Practicing Model Predictive Control. *IFAC-PapersOnLine*, 49(6), 34–39.
- INTECO (2014). 3D Crane. URL <http://www.inteco.com.pl/products/3d-crane/>. Accessed: 2021-01-13.
- Keller, M., Ritter, D., Schmitt, L., Hänggi, S., Onder, C., Abel, D., and Albin, T. (2020). Teaching Nonlinear Model Predictive Control with MATLAB/Simulink and an Internal Combustion Engine Test Bench. *IFAC-PapersOnLine*, 53(2), 17190–17197.
- Kelly, M. (2017). An Introduction to Trajectory Optimization: How to Do Your Own Direct Collocation. *SIAM Review*, 59(4), 849–904.
- Mayne, D.Q., Rawlings, J.B., Rao, C.V., and Sokaert, P.O.M. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6), 789–814.
- McInerney, I. and Kerrigan, E.C. (2018). Automated Project-based Assessment in a Predictive Control Course. In *2018 UKACC 12th International Conference on Control (CONTROL)*, 443. IEEE, Sheffield, UK.
- Nie, Y. and Kerrigan, E.C. (2020). External Constraint Handling for Solving Optimal Control Problems With Simultaneous Approaches and Interior Point Methods. *IEEE Control Systems Letters*, 4(1), 7–12.
- Pannocchia, G., Gabiccini, M., and Artoni, A. (2015). Offset-free MPC explained: Novelties, subtleties, and applications. *IFAC-PapersOnLine*, 48(23), 342–351.
- Rawlings, J.B., Mayne, D.Q., and Diehl, M. (2020). *Model Predictive Control: Theory and Design*. Nob Hill Publishing, Madison, Wisconsin, US, 2nd edition.
- Rossiter, J., 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.
- Saltık, M.B., Özkan, L., Ludlage, J.H.A., Weiland, S., and Van den Hof, P.M.J. (2018). An outlook on robust model predictive control algorithms: Reflections on performance and computational aspects. *Journal of Process Control*, 61, 77–102.
- Sokaert, P.O.M. and Rawlings, J.B. (1999). Feasibility issues in linear model predictive control. *AIChE Journal*, 45(8), 1649–1659.