Ref. Ares(2024)6898539 - 30/09/2024

Fostering Opportunities Towards Slovak Excellence in Advanced Control for Smart Industries

D2.2. Report on the software-development activities

Date by 30 September 2024
v.1

Project: 101079342 — FrontSeat —HORIZON-WIDERA-2021-ACCESS-03

Funded by the European Union

TABLE OF CONTENTS

Funded by the European Union

DELIVERABLE INFORMATION

Work package	WP2 - Promoting Excellence in Automatic Control Research
Task	Task T2.2: Software development and support
Due date	30 September 2024
Submission date	30 September 2024
Deliverable lead	Slovak University of Technology - STUBA
Version	v1
Authors	Juraj HOLAZA – STUBA Juraj ORAVEC – STUBA
Reviewers	Radoslav PAULEN – STUBA Miroslav FIKAR – STUBA Martin MONNINGMANN – RUB
Dissemination Level	PU - Public

Abstract	The project aims at increasing the research and academic prospects of Slovak University of Technology in Bratislava, Slovakia (STUBA) and at initiating the evolution of STUBA into a modern, reputed excellent institution that performs high-quality research in advanced automatic control, educates top-quality scholars and industrial practitioners, and is successful in active dissemination and exploitation of its research and innovation efforts. For this purpose, STUBA teams up with two renowned research groups in automatic control from RU Bochum, Germany (RUB) and Pisa University, Italy (UNIPI). The specific goals of the action are to reinforce the collaboration with the two research groups from Western Europe, to intensify research in advanced automatic control, to open up new collaboration channels through academic and industrial networking, to train excellent young/senior researchers and project managers, and to effectively disseminate and exploit the research results of STUBA. The unique features of the project are: - Adoption/amendment of internal research project-related rules and procedures and develop project management toolbox, - Research efforts aiming at the continued creation of high-quality research results and software tools, - Establishment of a series of guest scientific and academic lectures, - Exchanges and training of project managers and research (junior and senior) personnel, - Organisation of conferences and invited sessions, seminars with industry, and annual summer schools, - Preparation and implementation of a new PhD curriculum at STUBA, - Establishment of an academic-industrial research and innovation cluster.
Keywords	Control theory and optimization; Sensor networks, embedded systems, hardware platforms;

Funded by the European Union

	Embedded systems; Monitoring and control systems; Embedded systems in automation and control.

Document Revision History

Version	Date	Description of change	List of contributors

Disclaimer

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

Funded by the European Union

PARTNERS

The consortium of FrontSeat consists of 3 partners, as presented here below.

STUBA

Slovak University of Technology in Bratislava

RUB

Ruhr University Bochum

UNIPI

University of Pisa

Funded by the European Union

ABBREVIATIONS

Abbreviation	Expanded Version
eMPC	Explicit Model Predictive Control
MPC	Model Predictive Control
MPT+	Multi-Parametric Toolbox Plus
PWA	Piecewise affine

Funded by the European Union

EXECUTIVE SUMMARY

This report summarizes the software development efforts of the project FrontSeat. The developments were focused on predictive control software. Multi-Parametric Toolbox Plus (MPT+) is an open-source extension to the MATLAB-based Multi-Parametric Toolbox (MPT) aimed for the design, analysis, and deployment of Model Predictive Control (MPC) policies. It extends the MPT into new complexity reduction techniques of explicit Model Predictive Control (eMPC) strategies and tube-based MPC designs. It can be downloaded from its GitHub page or comfortably managed (install, update, uninstall) using the tbxManager. During its two years of development, MPT+ has already achieved 89 downloads, 2 publications (peer-reviewed scientific papers indexed in Web of Science and Scopus databases), 4 update releases, and 4 technical presentations (including 2 lectures delivered to the world-wide audience at the scientific conferences organized under IEEE).

Funded by
the European Union

SOFTWARE TOOLBOXES

Multi-Parametric Toolbox Plus (MPT+)

Model Predictive Control (MPC) is a sophisticated control strategy widely utilized in both academia and industry. Its popularity stems from its robustness and versatility, allowing for the incorporation of physical, economic, and environmental constraints directly into the optimization problem. MPC can predict system behaviour over a finite time horizon, enhancing performance and maximizing production efficiency. However, these benefits come with challenges, particularly in the implementation of MPC policies. In industrial applications, MPC requires multiple stages of development, verification, and validation before deployment. Given the time and complexity involved, there is a significant demand for advanced software toolboxes to facilitate rapid prototyping and analysis.

Multi-Parametric Toolbox Plus (or shortly MPT+) is an open-source MATLAB-based toolbox for the design, analysis, and deployment of optimal controllers, such as MPC policies, for constrained linear systems. MPT+ is a direct extension of the well-established Multi-Parametric Toolbox (MPT), which has garnered over 35,000 downloads and more than 1,300 citations worldwide over two decades since releasing the MPT3.

MPT+ focuses on integrating new, cutting-edge control approaches in MPC design that are currently being explored by research groups globally. The MPT+ toolbox is easily accessible via its dedicated GitHub page and can be installed manually or through the MATLAB-based tbxManager, which simplifies the process with straightforward commands to install, update, or uninstall the toolbox. MPT+ introduces novel methods for formulating MPC policies, with particular emphasis on low-complexity explicit Model Predictive Control (eMPC) and tube-based MPC designs. The following sections describe each approach in detail.

Design and verification of low-complexity explicit MPC controllers

To ensure theoretical guarantees of optimality and stability, the MPC optimization problem must be solved within the duration of a single sampling period. This task is challenging, especially on typical industrial hardware with limited computational resources. The conventional eMPC implementation has two phases: construction of the eMPC controller (offline phase), and (ii) real-time evaluation of control actions (online phase). Therefore, eMPC offers a solution by precomputing the MPC control law offline for all possible initial conditions, resulting in a piecewise affine (PWA) control law. Each piece of the PWA feedback law is defined over a specific domain, known as a critical region. In the online phase, obtaining the control action is reduced to a simple function evaluation, making it feasible even on low-end industrial hardware (solving the point location problem/exploring the look-up table). However, even for small to moderate-sized problems, eMPC can become overly complex due to the large number of PWA critical regions. This complexity can lead to a significant memory footprint, which may hinder the implementation of the predictive controller. To address this issue, complexity reduction techniques are employed to decrease the number of affine pieces and regions. These techniques can be generally categorized into those that do not introduce suboptimality into the evaluation of optimal control action and those that do.

The MPT+ toolbox incorporates three approaches from each category to effectively manage complexity of constructed eMPC policies. Specifically, MPT+ has integrated features of the *LowCom* package presented in (1) and extended it by the polynomial approximation method presented in (2).

Let us assume that eMPC policy was constructed in the MPT framework as follows:

```
eMPC = MPCController(model,N).toExplicit()
```

where *N* is the prediction horizon and *model* denote MPT object with information about the prediction model, constraints, and penalizations. For more information on how to construct eMPC readers are referred to the official [MPT]() web site or to the [MPT+]() web site, where one can find also all examples of this low-complexity explicit MPC package.

Low-Complexity Explicit MPC without Induced Suboptimality

To reduce complexity of eMPC policies without inducing suboptimality, i.e., without sacrificing any control performance, MPT+ allows users to use one of three reduction techniques.

Optimal-region-merging-based complexity reduction

This method is based on merging critical regions that share the same expression of the closed-loop feedback law and whose union is convex. The method is described in more detail in (3). The optimal-region-merging technique can be applied using the *simplify* function:

```
eMPCsimple = eMPC.simplify('orm')
```

This returns a new, simplified explicit controller object that performs the same way as the original eMPC. The corresponding simplified polytopic partition can be analysed by plotting analogously to the original explicit MPC:

```
eMPCsimple.partition.plot()
```

Clipping-based complexity reduction

The clipping-based method exploits the continuity of the explicit MPC feedback law to remove critical regions where the control action is saturated at a minimal or maximal value. The space covered by such saturated regions is replaced by extensions of the unsaturated critical regions, followed by straightforward applying a clipping filter. The technique is detailed in (4). This complexity reduction can be achieved by:

```
eMPCsimple = eMPC.simplify('clipping')
```

Separation-based complexity reduction

The separation-based technique exploits the saturation properties of continuous explicit MPC feedback laws. Unlike the clipping method, the saturated regions are removed, and only the unsaturated critical regions are retained. A separation function is then devised to uniquely determine the optimal control input for initial conditions not contained within the retained part of the explicit solution. This complexity reduction is described in (5) and can be invoked using:

```
eMPCsimple = eMPC.simplify('separation')
```


Funded by the European Union

Low-Complexity Explicit MPC with Induced Suboptimality

To reduce the complexity of eMPC policies while inducing suboptimality, i.e., sacrificing some control performance, MPT+ provides three reduction techniques. To analyse the performance loss, the average and worst-case performance degradation of the simplified controller *eMPCsimple* can be compared to the optimal one *eMPC* using:

```
[mean_perfor,worst_perfor] = eMPC.comparePerformance(eMPCsimple)
```

The derivation of such simplified controllers *eMPCsimple* is described in the following subsections.

Fitting-based complexity reduction

The two-step fitting procedure was suggested in (6). First, a simpler partition composed of fewer critical regions is designed by solving the MPC problem for a shorter prediction horizon. The parameters of the simpler feedback are then optimized to minimize the suboptimality, represented by the integrated squared error between the original and simpler feedback. This complexity reduction scheme is available via:

```
eMPCsimple = eMPC.simplify('fitting')
```

Minimum-time-based complexity reduction

This strategy, introduced in (7), reduces complexity by solving a simpler cost function. Here, a minimum-time problem is solved, where the objective is to minimize the arrival time to a specified terminal set. The controller can be simplified using:

```
eMPCsimple = EMinTimeController(model)
```

Polynomial-approximation-based complexity reduction

A polynomial approximation is constructed for the piecewise affine (PWA) control law determined by solving the explicit MPC with a linear (1-norm or inf-norm) cost function as per (8). The approximated polynomial controller can be simplified, evaluated, and plotted using:

```
eMPCsimple = PolynomialMPC(eMPC)
```

As the code for the polynomial approximation of the MPC policy is distributed within an open-source framework for the first time, we provide an illustrative case study. In the first step, the standard implicit MPC policy *iMPC* is constructed and subsequently converted into its explicit form *eMPC*. Then, the eMPC policy is approximated by a polynomial control law with a default degree of three, yielding the object *polyMPC*. *Figure 1*, to visually analyse the suboptimality, both the optimal eMPC (blue dashed line) and the approximated polynomial feedback law (red dashed line) are plotted. The yellow area represents the feasible domain of control actions, where recursive feasibility of the original constraints and asymptotic stability of the system are guaranteed.

```
   %% Polynomial Approximation
% LTI system
model = LTISystem('A', [1.2], 'B', [1]);
model.u.min = [ -1 ];
```


Funded by the European Union

```
model.u.max = [  1 ];
model.x.min = [ -4 ];
model.x.max = [  4 ];
% Linear penalty functions
model.x.penalty = OneNormFunction(diag([ 1 ]));
model.u.penalty = OneNormFunction(diag([ 1 ]));
% Prediction horizon
N = 10;
% Construct MPC controller
iMPC = MPCController(model,N)
% Explicit MPC controller construction
eMPC = iMPC.toExplicit
% Polynomial Approximation
polyMPC = PolynomialMPC(eMPC)
polyMPC.plot
x0 = [ -2 ]; % Initial condition
u_approx = polyMPC.evaluate(x0) %control action of the
polynomial
u_optimal = eMPC.evaluate(x0)%control action of the optimal MPC
```

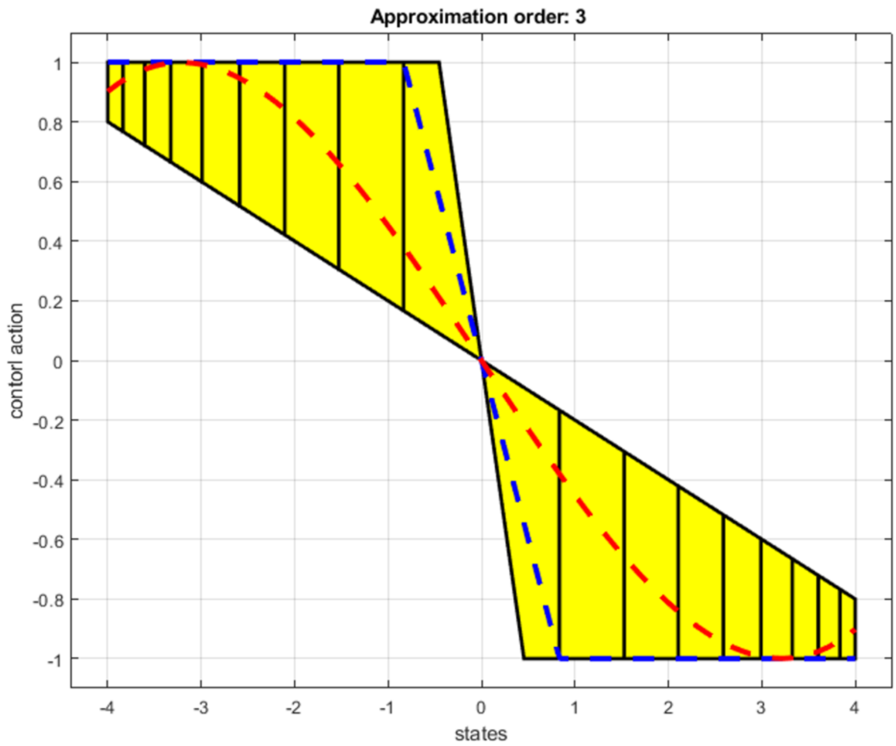

Figure 1: Polynomial approximation (red dashed line) of the eMPC policy (blue dashed line) with emphasized stabilizing domain (yellow area).

Tube MPC design

Robust MPC design for linear time-invariant (LTI) systems is affected by parametric and/or additive disturbances. Although MPT benefits from various advanced methods introduced in (9), such as evaluating forward/backward robust invariant sets, evaluating worst-case scenario-based robust (explicit) MPC design is often intractable and time-consuming. The complexity of MPC controller design is determined by the number of decision variables and the length of the prediction horizon, hence limiting robust MPC design to systems of modest complexity.

Funded by the European Union

The main benefit of implementing Tube MPC is that it designs robust MPC in a computationally tractable way mimicking nominal (non-robust) MPC design. In this approach, the control action consists of a sequence of nominal inputs that are executed in the nominal model, and the uncertainties are eliminated by a perturbation feedback law.

Recent version of MPT+ toolbox distinguishes two types of the tube MPC policies:

- Rigid Tube MPC policy,
- Implicit Tube MPC policy.

In what follows, both approaches are introduced as well as their implementation within the MPT+ framework.

Rigid Tube MPC policy

The basic principle of the rigid tube MPC has two main parts. First a stabilizing simple controller is constructed that can stabilize the system around the origin. Next a tube is constructed such that it denotes the minimal set of all states where the simple precomputed controller can keep all future states within this tube even if any uncertainty and/or disturbance will impact the controlled system. In the control theory, this tube is also referred to as a minimal robust positive-invariant set. Finally, with all ingredients in hand, the original robust MPC design problem can be reformulated as a simpler problem utilizing nominal MPC that is easier to solve. It should be noted that the main drawback of this approach is that the tube is difficult to construct for higher dimensional systems.

MPT+ enables, by default, the rigid tube MPC design according to (10), where the tube is constructed based on (11). By assuming that we have defined the MPC policy in MPT framework and stored it as *model*, then the rigid tube based MPC policy can be derived by the command

```
iMPC = TMPCController(model,N,option)
```

where *N* denotes the prediction horizon and *option* represents user defined options that can further tune the resulting tube MPC stored as an object *iMPC*. Subsequently one can perform closed-loop simulations from the initial point *x0* and with *Nsim* steps via a simple command

```
ClosedLoopData = iMPC.simulate(x0,Nsim)
```

The closed loop data can be visualized, see Figure 2, to easily analyse performance of the constructed MPC.

Funded by the European Union

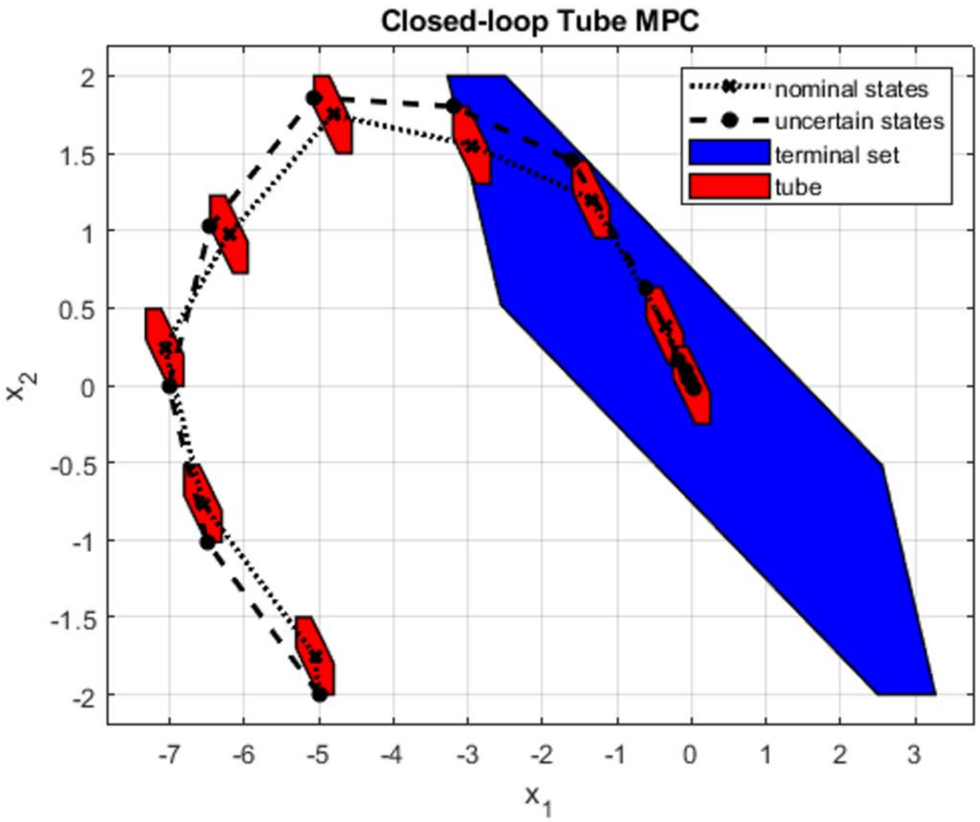

Figure 2: Visualize closed-loop performance of a rigid tube MPC initialized from state [-5, -2]. The dashed line represents uncertain states of the controlled system, dotted line denotes nominal states of the optimization problem, red objects are tubes at respective time step, and the blue object is the terminal set.

It should be emphasized that the final tube MPC object *iMPC* is compatible with the syntax of other MPT classes used for conventional MPCs. For example, we can use an overloaded function to create explicit tube MPC via typing

```
eMPC = iMPC.toExplicit
```

To evaluate the control action

```
iMPC.evaluate(x0)
eMPC.evaluate(x0)
```

or to make visual simulations

```
eMPC.clicksim
```

to name a few. For more information and case studies, readers can visit our dedicated rigid tube web page.

Implicit tube MPC policy

Implicit tube MPC strategy differs from its rigid counterpart in the definition of the tube. Specifically, here the tube is not constructed as a geometric object (geometric set), instead it is implicitly defined as a set of constraints. Consequence of this formulation is twofold. Firstly, such implicit tube enables the tube MPC policy to be defined also for larger systems, i.e., scalability with the state dimension is improved (significantly increased). On the other hand, the implicit formulation introduces additional optimization variables (and associated constraints) into the optimization problem to be solved, i.e., complexity of the resulting tube MPC policy is more complex. For this reason, if we want to construct tube MPC policy for a

Funded by the European Union

small dimensional system, then the rigid approach is more convenient. However, if we aim for a higher dimensional system, then the implicit approach should be applied.

MPT+ enables the implicit tube MPC design according to (12). By assuming that we have defined the MPC policy in MPT framework and stored it as *model*, then the implicit tube based MPC policy can be derived with the same command as the rigid tube based MPC however with the specified optional parameter *TubeType* to be set to *implicit* as follows

```
option = {'TubeType','implicit'}
iMPC = TMPCController(model,N,option)
```

where N denotes the prediction horizon and *option* represents user defined options that can further tune the resulting tube MPC stored as an object *iMPC*. Subsequently, one can also use all inherited, or overloaded functions of MPT. Since both implicit and set-based tube MPC designs provide the same control performance, their computational complexity just scales differently with the problem size, hence by performing the same closed-loop simulation we would get the same results as reported in Figure 2. For more information and case studies, readers can visit our dedicated implicit tube web page.

MPT+ Summary

This section summarizes relevant indicators of the software that were achieved within two years of the MPT+ development. All achievements of the MPT+ are compactly listed in Table 1, while sources and explanations to the presented numbers are provided next.

No. of downloads	**No. of releases**	**No. publications**	**No. of technical presentations**
89	4	2	4

Table 1: List of MPT+ achievements

No. of downloads – The MPT+ toolbox is freely-available from its public repository located at GitHub, and, simultaneously, provides the interface also for users of the user-friendly tbxManager toolbox. The indicator of downloads can be access via GitHub link. It is important to note that this number does not include cloning of the repository project, what is preferred approach by advanced users of widely-used version control systems Git. Hence, it is assumed, that MPT+ toolbox has a wider impact on the research, pedagogic, and/or industrial community.

No. of releases – All releases are listed in the GitHub link, where they can be downloaded. Users are informed about these releases, i.e., what was added, improved, or go t fixed, via our changelog that is also available in the dedicated GitHub page.

MPTplus R20230314 - introduced the Rigid Tube MPC design
MPTplus R20230331 - Introduced framework for the explicit Tube MPC design. - Added the package for the design and verification of low-complexity explicit MPC controllers in MPT3: LowCom. - Introduced polynomial-based approximation of explicit control law. - Enabled depicting of the indexed explicit partition for the general explicit MPC controller.

Funded by the European Union

MPTplus R20230614 - Introduced the polynomial approximation of the explicit MPC controllers towards the uncertain LTI systems.
MPTplus R20240424 - Introduced the implicit Tube MPC design enabling handling also the large-scale uncertain LTI systems.

Table 2: List of releases

No. of publications – All publications (scientific papers) are available in Zenodo and are listed as follows:

Tube MPC Extension of MPT: Experimental Analysis **Authors:** Juraj Holaza, Katrin Kvasnicová, Erika Pavlovičová, Juraj Oravec **Conference:** Proceedings of the 24th International Conference on Process Control, Slovakia, 2023 **Indexed in scientific databases:** Web of Science, Scopus **Zenodo views**: 124 **Zenodo downloads:** 88
A software package for MPC design and tuning: MPT+ **Authors:** Juraj Holaza, Lenka Galčíková, Juraj Oravec, Michal Kvasnica **Conference:** 62nd IEEE Conference on Decision and Control (CDC), Singapore, 2023 **Indexed in scientific databases:** Web of Science, Scopus **Zenodo views**: 53 **Zenodo downloads:** 38

Table 3: List of publications

Except for the scientific papers, the extensive technical documentation is freely-available online at the homepage of the MPT+ toolbox at GitHub to support the community of the users.

Moreover, the community of the users is continuously keeping in touch with the recent updates and events promoting the dissemination and communication of the MPT+ toolbox using the dedicated webpage of the MPT+ *story* at GitHub/story.

No. of technical presentations – In this indicator two types of presentations were carried out: i.) Presentations for the involved research groups within the FrontSeat project and ii.) conference presentations. The main difference between those types is that while the i.) aimed to brainstorm new ideas and features that can be added to the MPT+, presentations of ii.) targeted to introduce and inform new functionalities of this toolbox to a wide research audience.

Tube MPC Extension of MPT: Experimental Analysis **Speaker:** Erika Pavlovičová **Event:** 24th International Conference on Process Control, Slovakia **Date:** 7. June 2023
A software package for MPC design and tuning: MPT+ **Speaker:** Lenka Galčíková **Event:** 62nd IEEE Conference on Decision and Control (CDC), Singapore **Date:** 14. December 2023
Tube MPC Extension of MPT **Speakers:** Juraj Holaza and Juraj Oravec **Event:** Research Seminar on Smart Cybernetics - organized in the framework

Funded by the European Union

of the FrontSeat project at STUBA **Date:** 26 May 2023
Tube Model Predictive Control **Speaker:** Lenka Galčíková **Event:** Research Seminar on Smart Cybernetics - organized in the framework of the FrontSeat project at STUBA **Date:** 6 October 2023

Table 4: List of publications

Funded by the European Union

REFERENCES

1. *Design and Verification of Low-Complexity Explicit MPC Controllers in MPT3.* **M. Kvasnica, J. Holaza, B. Takács, D. Ingole.** Linz, Austria : s.n., 2015. Proceedings of the European Control Conference 2015. pp. 2600–2605.

2. *Stabilizing polynomial approximation of explicit MPC.* **M. Kvasnica, J. Löfberg, M. Fikar.** 10, 2011, Automatica, Vol. 47, pp. 2292–2297.

3. *Optimal complexity reduction of polyhedral piecewise affine systems.* **T. Geyer, F.D. Torrisi, M. Morari.** 7, 2008, Automatica, Vol. 44, pp. 1728-1740.

4. *Clipping-Based Complexity Reduction in Explicit MPC.* **M. Kvasnica, M. Fikar.** 7, 2012, IEEE Transactions on Automatic Control, Vol. 57, pp. 1878–1883.

5. *Complexity reduction of explicit model predictive control via separation.* **M. Kvasnica, J. Hledík, I. Rauová, M. Fikar.** 6, 2013, Automatica, Vol. 49, pp. 1776–1781.

6. *Nearly-optimal simple explicit MPC regulators with recursive feasibility guarantees.* **B. Takács, J. Holaza, M. Kvasnica, S. Di Cairano.** Firenze, Italy : s.n., 2013. 52nd IEEE Conference on Decision and Control.

7. *Stabilizing low complexity feedback control of constrained piecewise affine systems.* **P. Grieder, M. Kvasnica, M. Baotić, M. Morari.** 2005, 2005, Automatica, Vol. 41, pp. 1683-1694.

8. *Stabilizing polynomial approximation of explicit MPC.* **M. Kvasnica, J. Löfberg, M. Fikar.** 10, 2011, Automatica, Vol. 47, pp. 2292–2297.

9. *Reachability analysis and control synthesis for uncertain linear systems in MPT.* **M. Kvasnica, B. Takács, J. Holaza, D. Ingole.** Bratislava, Slovak Republic : s.n., 2015. 8th IFAC Symposium on Robust Control Design.

10. *Robust model predictive control of constrained linear systems with bounded disturbances.* **D.Q. Mayne, M.M. Seron, S.V. Raković.** 2, 2005, Automatica, Vol. 41, pp. 219-224.

11. *Invariant approximations of the minimal robust positively invariant set.* **S. V. Rakovic, E. C. Kerrigan, K. I. Kouramas and D. Q. Mayne.** 3, 2005, IEEE Transactions on Automatic Control, Vol. 50, pp. 406-410.

12. *The implicit rigid tube model predictive control.* **Raković, Saša V.** 2023, Automatica, Vol. 157.

Funded by the European Union