Received: 8 January 2023 | Revised: 8 May 2023 | Accepted: 18 June 2023

DOI: 10.1002/aic.18172

AIChE JOURNAL

RESEARCH ARTICLE

Process Systems Engineering

Explicit model predictive control through robust optimization

Iosif Pappas [1,2] | Nikolaos A. Diangelakis [3] | Efstratios N. Pistikopoulos [1,2]

[1]Artie McFerrin Department of Chemical Engineering, Texas A&M University, College Station, Texas, USA

[2]Texas A&M Energy Institute, Texas A&M University, College Station, Texas, USA

[3]School of Chemical and Environmental Engineering, Technical University of Crete, Chania, Greece

Correspondence
Efstratios N. Pistikopoulos, Artie McFerrin Department of Chemical Engineering, Texas A&M University, College Station, TX 77843, USA.
Email: stratos@tamu.edu

Funding information
Texas A and M Engineering Experiment Station, Texas A and M University; U.S. Department of Energy, Grant/Award Numbers: DE-EE0007613, 4550 G WA324

Abstract

A strategy that calculates an explicit state feedback policy to regulate constrained uncertain discrete-time uncertain linear systems is presented. We consider uncertain processes, affected by box-bounded multiplicative uncertainty as well as bounded additive uncertainty with linear state and inputs constraints. The proposed method includes (i) the calculation of a terminal set constraint and (ii) the robust reformulation of state constraints in the prediction horizon. These features allow the derivation of the desired policy by solving a single multiparametric quadratic programming problem that guarantees feasible operation in the presence of uncertainty. Additionally, we employ variable and constraint elimination approaches to enhance the computational performance of the strategy. We demonstrate the steps and benefits of these developments with a numerical example and a chemical engineering case study.

KEYWORDS

explicit model predictive control, multiparametric programming, optimization under uncertainty, robust optimization

1 | INTRODUCTION

Linear Model Predictive Control (MPC) is the established strategy for the supervisory regulation of discrete-time multivariable constrained systems. Its ability to control plants with multiple inputs and outputs and to impose constraint satisfaction have made MPC the widely accepted process control paradigm by both academic and industrial researchers and practitioners. By utilizing an MPC approach, the optimal vector of inputs that minimize a performance index while considering (i) the process model, (ii) the constraint set, and (iii) a finite prediction horizon is calculated; however, only the first control input of the sequence is applied to the system. As soon as the next measurement (or its estimate) is made available, the horizon is rolled over by a single time step, and the optimization problem is resolved. Such online process control procedure is repeated to achieve the desired objective, and as a result facilitates an implicit feedback strategy.

Explicit/Multiparametric Model Predictive Control (mpMPC) aims to precalculate and express the aforementioned vector of optimal inputs as a function of the states of the system which can partially describe the uncertainty vector of the model. That is achieved by recasting the MPC problem to the equivalent multiparametric quadratic programming problem (mpQP) and solving it. Besides the states, (i) the outputs, (ii) the measured disturbances, and (iii) the varying set-points can be treated in a similar manner, that is, as uncertain parameters. The benefits of mpMPC are, (i) the analysis and understanding of how future measurements and measured disturbances will affect the control performance before the closed-loop simulation has occurred, (ii) the alleviation of solving an online optimization program at each sampling instant, highly beneficial for plants with fast dynamics and without significant computational capacity, and (iii) the ability to solve integrated (nested) optimization programs, such as simultaneous design and control, multiobjective and multilevel optimization problems.[1]

Being a model-based strategy, MPC approaches inevitably depend on the accuracy of the underlying model. In practical applications, model uncertainty is prevalent due to the mismatch between the real plant and the model used for the derivation of the control law (endogenous uncertainty) and/or unmeasured disturbances which are not captured by the model (exogenous uncertainty). Consequently, the closed-loop behavior of the system is not as desired and constraint violations may occur, thus derailing the operation of the plant. In an effort to address this challenge within the mpMPC literature, strategies that account for additive and multiplicative (parametric) uncertainty have been developed. For models with additive uncertainty, the research focus of the contribution of Sakizlis et al.[2] was on finding the worst-case uncertainty value of the constraint set through the use of flexibility analysis tools, Kerrigan and Maciejowski[3] solved a min-max problem to derive a robust solution, in Rodríguez-Ayerbe and Olaru[4] a nominal explicit control law was found which is subsequently robustified to account for disturbances, while Rubagotti et al.[5] created approximate robust control laws over simplicial partitions of the state space. Other contributions include the utilization of adjustable robust optimization[6] or tube MPC approaches that took advantage of multiparametric programming as part of the algorithmic procedure for their online implementation.[7] As far as plants with multiplicative uncertainty are concerned, Bemporad et al.[8] provided a min-max solution strategy for models with a linear objective function by solving a sequence of multiparametric linear programming problems and Kouramas et al.[9] solved the case of an uncertain plant with a quadratic objective function and linear constraints using robust optimization and dynamic programming techniques. Sheikhbahaei et al.[10] extended the results of the latter effort to derive fault-tolerant mpMPC policies, Oberdieck[11] robustified the constraint set by calculating the robust counterpart and solving a multiparametric programming problem at each stage which avoided the need for dynamic programming. Furthermore, Sun et al.[12] found robust explicit control laws of dynamic systems that track the necessary optimality conditions, whereas Nguyen et al.[13] calculated the robustness margins of a derived piecewise affine control policy. We note that throughout the article, the term uncertain in the context of mpMPC refers to bounded and measured uncertainty compared to the one in the context of robust optimization which refers to bounded but unmeasured uncertainty.

Even though there have been many developments to tackle the challenges arising from uncertainty considerations in mpMPC, there is a limited number of algorithms for systems with multiplicative uncertainty, a quadratic objective function and linear constraints. This effort aims to alleviate the need of solving multiple multiparametric optimization problems for this type of systems stemming from the use of dynamic programming in the solution strategy. In this respect, and inspired by the benefits of using robust optimization to account for the impact of uncertainty, we build on our preliminary results[14] and present an algorithm that derives a policy that hedges against uncertainty which is defined within a prespecified interval. That is achieved by solving, a single mpQP over a robustified constraint set. Furthermore, we preserve the linearity of the constraints, hence making the off-the-shelf use of an existing multiparametric optimization solver readily possible.

This contribution is organized in five sections. Following the introduction, the problem formulation and the objectives of the study are defined. The methodology and algorithmic developments to derive an uncertainty-immune explicit control policy are exhibited in Sections 3 and 4 respectively. In Section 5 we analytically demonstrate the steps of the approach on a numerical case study as well as on a chemical engineering case study. Finally, in Section 6 we present the conclusions of this study.

2 | PROBLEM STATEMENT

In this work, we take into account system uncertainty by considering that the model is not deterministic, but rather that it belongs to a family of models specified within a range of uncertainty. Also, we assume that the future states are influenced by the presence of additive uncertainty. The objective of this study is to regulate a system whose dynamics are dictated by the following linear state space description:

$$x_{k+1} = A(\delta A)x_k + B(\delta B)u_k + Cd_k, \tag{1}$$

where

$$A = \bar{A} + \delta A. \tag{2}$$

$$B = \overline{B} + \delta B. \tag{3}$$

$$\delta A \in \mathcal{A} = \left\{ \delta A \in \mathbb{R}^{m \times m} \,|\, -\epsilon_\alpha |\bar{A}| \le \delta A \le \epsilon_\alpha |\bar{A}| \right\}. \tag{4}$$

$$\delta B \in \mathcal{B} = \left\{ \delta B \in \mathbb{R}^{m \times n} \,|\, -\epsilon_\beta |\overline{B}| \le \delta B \le \epsilon_\beta |\overline{B}| \right\}. \tag{5}$$

In Equation (1), $x \in \mathbb{R}^m$, $u \in \mathbb{R}^n$ and $d \in \mathbb{R}^p$ are the vectors of the states, control actions and additive disturbances of the system respectively, while k represents the time index of the discrete model. We assume that a matrix $\overline{\Phi}$ consists of elements $\phi_{i,j}$, then $|\overline{\Phi}|$ consist of elements $|\phi_{i,j}|$. In contrast to classic mpMPC formulations, the considered model is uncertain and depends on the scalars ϵ_α and ϵ_β that are used to form a box uncertainty set and is described in Equations (2)–(5). Specifically, apart from the deterministic (nominal) component of the model, specified by the matrices $\bar{A} \in \mathbb{R}^{m \times m}$ and $\overline{B} \in \mathbb{R}^{m \times n}$, there is also an uncertain component represented by the matrices δA and δB. This uncertain part of the model depends on ϵ_α and ϵ_β which can realize values in [0,1) and define the range that δA and δB can obtain based on their nominal values $\bar{A}$ and $\overline{B}$ respectively. For instance, if $\epsilon_\alpha = 0.1$ and $\epsilon_\beta = 0.2$, each element of the state matrix is allowed to deviate by 10% of its nominal value while each element of the inputs matrix is allowed by 20% from its nominal value.

Additionally, our system, described by model (1), is subject to constraints that dictate that boundaries of its operation or desired specifications are of the following form:

$$g(x_k, u_k, d_k) \le 0. \tag{6}$$

Note that the constraint set is also uncertain since it is a function of the system states at time step k, the values of which change based on the process model.

With that in mind, the problem definition of the study is to solve the MPC problem, described by the formulation (7):

$$\begin{aligned}
\min_{u_0,\dots,u_{N-1}} \quad & x_N^T P_R x_N + \sum_{k=0}^{N-1} x_k^T Q_R x_k + u_k^T R u_k \\
\text{s.t.} \quad & x_{k+1} = A(\delta A)x_k + B(\delta B)u_k + Cd_k, \; k = 0,\dots,N-1 \\
& A = \bar{A} + \delta A \\
& B = \overline{B} + \delta B \\
& \delta A \in \mathcal{A} = \left\{ \delta A \in \mathbb{R}^{m\times m} \,|\, -\epsilon_\alpha |\bar{A}| \le \delta A \le \epsilon_\alpha |\bar{A}| \right\} \\
& \delta B \in \mathcal{B} = \left\{ \delta B \in \mathbb{R}^{m\times n} \,|\, -\epsilon_\beta |\overline{B}| \le \delta B \le \epsilon_\beta |\overline{B}| \right\} \\
& x_k \in \mathcal{X}, \quad k = 0,\dots,N \\
& u_k \in \mathcal{U}, \quad k = 0,\dots,N-1 \\
& d_k \in \mathcal{D}, \quad k = 0,\dots,N-1 \\
& x_N \in \mathcal{T} \\
& x_0 = x(0) .
\end{aligned} \tag{7}$$

The considered problem (7) includes a prediction horizon of N steps into the future. $Q_R \in \mathbb{R}^{m\times m}$ and $R \in \mathbb{R}^{n\times n}$ are the cost weight matrices penalizing the states and inputs, respectively, and $P_R \in \mathbb{R}^{m\times m}$ is a terminal matrix derived by solving the Riccati equation for discrete systems. The Riccati equation needs to be solved before solving the above MPC problem. The sets $\mathcal{X}$,$\mathcal{U}$, and $\mathcal{D}$ require the states, inputs and the additive disturbances to belong in them for all steps into the future, while the terminal set constraint $\mathcal{T}$ enforces the states to exist in it at the Nth time step. We assume that the terminal constraint contains the origin in its interior. Additionally, we assume that the matrices Q_R and P_R are positive semi-definite and that R is positive definite.

The main challenge associated with solving problem (7) with state-of-the-art multiparametric programming solvers is the presence of left-hand side uncertainty in the constraint set. For clarification purposes, assume the model box state bounds represented by $\hat{x}$ that need to be satisfied for every time step and a prediction horizon of one. For the state constraints at the end of the horizon this is expressed as follows:

$$x_1 \le \hat{x}. \tag{8}$$

Propagating the states for arbitrary values of ϵ_α, ϵ_β, δA, and δB we have:

$$A(\delta A)x_0 + B(\delta B)u_0 + Cd_0 \le \hat{x}. \tag{9}$$

$$(\bar{A} + \delta A)x_0 + (\overline{B} + \delta B)u_0 + Cd_0 \le \hat{x}. \tag{10}$$

Since we do not have prior information of the uncertainty values, we cannot substitute their values and solve it as an mpQP. Also, current publicly available software implementations of multiparametric optimization cannot consider left-hand side uncertainty in the problem description. There are contributions in the literature that can solve this class of problems exactly, but that is the case for small-scale problems because that requires the analytic solution of a nonlinear system of equations.[15] The objective of this work is to solve (7) explicitly and derive $u = h(x,d)$ for all values within the uncertainty range of δA and δB, and as a result, there is an *a priori* guarantee of feasible operation while obtaining the benefits of the offline policy.

3 | METHODOLOGY

The backbone of the proposed methodology is the enforcement of constraints to the problem formulation that will prevent constraint violation. That is achieved by reformulating the original state constraints with their robust counterpart that will guarantee feasibility from the first time step until the Nth step, and by imposing a suitable terminal set constraint, $\mathcal{T}$, that establishes the desired feasibility property after the Nth time step.

3.1 | Robustification

As mentioned before, since the model is uncertain and allowed to vary at each time step, we need to be able to find solutions that are immune to uncertainty. In this work, that is achieved by employing the robust counterpart of the original problem formulation.[9,16] Assume that the elements of the nominal and uncertain matrices of Equation (1) are $\bar{a}_{i,j}$, $\bar{b}_{i,j}$ and $\delta a_{i,j}$ and $\delta b_{i,j}$ respectively. In addition, $c_{i,j}$ is the (deterministic) element of the additive disturbance matrix (C). Again, for clarification purposes, assume that we would like to robustify the state constraints on the first time step of the prediction horizon. Then for each constraint, i, and for zero infeasibility tolerance, we need to satisfy the following expression:

$$\begin{aligned}
&\sum_{j=1}^{j=m} \bar{a}_{i,j} x_{0|j} + \sum_{j=1}^{j=m} |\delta a_{i,j}||x_{0|j}| + \sum_{j=1}^{j=n} \bar{b}_{i,j} u_{0|j} \\
&+ \sum_{j=1}^{j=n} |\delta b_{i,j}||u_{0|j}| + \sum_{j=1}^{j=p} c_{i,j} d_{0|j} \le \hat{x}_i,
\end{aligned} \tag{11}$$

$$\begin{aligned}
&\sum_{j=1}^{j=m} \bar{a}_{i,j} x_{0|j} + \epsilon_\alpha \sum_{j=1}^{j=m} |\bar{a}_{i,j}||x_{0|j}| + \sum_{j=1}^{j=n} \bar{b}_{i,j} u_{0|j} \\
&+ \epsilon_\beta \sum_{j=1}^{j=n} |\bar{b}_{i,j}||u_{0|j}| + \sum_{j=1}^{j=p} c_{i,j} d_{0|j} \le \hat{x}_i,
\end{aligned} \tag{12}$$

where $x_{0|j}$, $u_{0|j}$, and $d_{0|j}$ represent the jth component of the state vector, input and additive disturbance vectors at the initial time step. The left-hand side of the above state constraint is the worst (most tight) formulation of the state constraints due to the addition of the absolute values for the uncertain elements of the model. We repeat this procedure for all state constraints in the prediction horizon. Consequently, if the solution satisfies the robust counterpart of the constraints for all time steps until the Nth step of the horizon, then we ensure that the operation of our system will be feasible until this point of time.

A challenge associated with Equations (11) and (12) is the inclusion of the absolute values for the states and inputs of the model which cannot directly be fed into multiparametric optimization solvers. The alleviation from such a challenge can be achieved by replacing the absolute values with artificial variables and their corresponding box constraints:

$$\sum_{j=1}^{j=m} \bar{a}_{i,j}x_{0|j} + \epsilon_\alpha \sum_{j=1}^{j=m} |\bar{a}_{i,j}|z_{0|j} + \sum_{j=1}^{j=n} \overline{b}_{i,j}u_{0|j} + \epsilon_\beta \sum_{j=1}^{j=n} |\overline{b}_{i,j}|v_{0|j} + \sum_{j=1}^{j=p} c_{i,j}d_{0|j} \leq \hat{x}_i \tag{13}$$

with

$$-z_{0|j} \leq x_{0|j} \leq z_{0|j}. \tag{14}$$

$$-v_{0|j} \leq u_{0|j} \leq v_{0|j}. \tag{15}$$

Similarly, this process can be illustrated further by considering state constraints for the second step of the prediction horizon:

$$x_2 \leq \widehat{x_i}. \tag{16}$$

By following the robustification approach that we described, we obtain the following result:

$$\sum_{j=1}^{j=m} \bar{a}_{i,j}x_{1|j} + \epsilon_\alpha \sum_{j=1}^{j=m} |\bar{a}_{i,j}|z_{1|j} + \sum_{j=1}^{j=n} \overline{b}_{i,j}u_{1|j} + \epsilon_\beta \sum_{j=1}^{j=n} |\overline{b}_{i,j}|v_{1|j} + \sum_{j=1}^{j=p} c_{i,j}d_{1|j} \leq \hat{x}_i \tag{17}$$

with

$$-z_{1|j} \leq x_{1|j} \leq z_{1|j}. \tag{18}$$

$$-v_{1|j} \leq u_{1|j} \leq v_{1|j}. \tag{19}$$

However, the expression contains x_1 which depends on the matrices A and B, and hence, it needs to be further robustified. Specifically:

$$\begin{aligned} &\sum_{j=1}^{j=m} \bar{a}_{i,j}\bar{a}_{j,i}x_{0|j} + \epsilon_\alpha \sum_{j=1}^{j=m} |\bar{a}_{i,j}||\bar{a}_{j,i}|z_{0|j} + \sum_{j=1}^{j=m} \bar{a}_{i,j}\overline{b}_{j,i}u_{0|j} \\ &+\epsilon_\beta \sum_{j=1}^{j=m} |\bar{a}_{i,j}||\overline{b}_{j,i}|v_{0|j} + \sum_{j=1}^{j=m} \bar{a}_{i,j}c_{j,i}d_{0|j} + +\epsilon_\alpha \sum_{j=1}^{j=m} |\bar{a}_{i,j}|z_{1|j} + \sum_{j=1}^{j=n} \overline{b}_{i,j}u_{1|j} \\ &+\epsilon_\beta \sum_{j=1}^{j=n} |\overline{b}_{i,j}|v_{1|j} + \sum_{j=1}^{j=p} c_{i,j}d_{1|j} \leq \hat{x}_i, \end{aligned} \tag{20}$$

with

$$-z_{1|j} \leq x_{1|j} \leq z_{1|j} \tag{21}$$

$$-v_{1|j} \leq u_{1|j} \leq v_{1|j}. \tag{22}$$

$$-z_{0|j} \leq x_{0|j} \leq z_{0|j}. \tag{23}$$

$$-v_{0|j} \leq u_{0|j} \leq v_{0|j}. \tag{24}$$

The procedure that we just followed for the state constraints at the initial time step is repeated for all time steps, and the constraint set is concatenated. Note that we do not need to robustify the box input constraints since we have full control of the decision variables of the problem. As far as the objective function is concerned, we assume that a nominal scenario is realized and not a worst-case scenario, and consequently it is not robustified. As it can be seen from the derived expressions, the robustification process introduces artificial variables and the corresponding constraints. Naturally, this introduces computational challenges which we address in the following section of the article.

3.2 | Terminal set calculation

The robust counterpart prevents from constraint violation until the *N*th step of the prediction horizon. The feasibility of the system after that point in time is warranted through the construction of a terminal set constraint. Specifically, we calculate a robust control invariant set and impose it as a terminal constraint to the problem formulation.

Definition (Reference 17): A set C is a robust control invariant set for system (1) if:

$$x_k \in \mathcal{C} \Rightarrow \exists u_k \in \mathcal{U} | A(\delta A)x_k + B(\delta B)u_k + Cd_k \in \mathcal{C}, \forall d \in \mathcal{D}, \forall t \geq 0. \tag{25}$$

In essence, we are looking for a set of states, such that there is a control input which will ensure that the propagation of the system will still exist in the same set.

The calculation of the robust control invariant set can be achieved through the determination of the one-step backward reachable set

$$\begin{aligned} Pre(\mathcal{X}) = \{x | \exists u \in \mathcal{U} | A(\delta A)x + B(\delta B)u + Cd \in \mathcal{X}, \forall d \in \mathcal{D}, \\ \forall \delta A \in \mathcal{A}, \forall \delta B \in \mathcal{B}\}. \end{aligned} \tag{26}$$

Based on that, the calculation of the robust control invariant set is a repetitive calculation during which starting from the initial set of states of the system, $\mathcal{X}$, we perform the following calculation

$$\mathcal{S}_{r+1} = Pre(\mathcal{S}_r) \cap \mathcal{S}_r, \tag{27}$$

where $\mathcal{S}_0 = \mathcal{X}$. This procedure is repeated until $\mathcal{S}_{r+1} = \mathcal{S}_r$. As soon as this set is calculated, it is set as the terminal set of our problem, $\mathcal{T}$. Hence, state feasibility after the end of the horizon is ensured. Overall, the robustification procedure guarantees that the states will enter the terminal set, and because the states can never violate this set, the states are guaranteed to exist in this region which contains the origin.

Please note that such a terminal set might not be possible to be found due to the terminal set algorithm not converging, something that we have not experienced when applying the algorithm to the case studies that we considered. We refer the reader to the excellent works of Blanchini[18] and Bertsekas[19] for further information on the topic.

4 | ALGORITHM FOR EXPLICIT MPC

In the previous section, we presented the required robustness considerations of the proposed control strategy. Apart from that, here we present further developments on how this approach can be materialized from an algorithmic implementation point of view.

4.1 | Fourier–Motzkin elimination

The robustifed problem has an updated set of inequality constraints in its description, due to the robust counterpart reformulation and the inclusion of the terminal set constraint. Importantly, the artificial vectors of variables, z and v, are now part of the model. It is established in the multiparametric programming literature and practice, that constructing the map of critical regions for a multiparametric program depends on the number of its variables and constraints. That is crucial especially for algorithms that rely on exploring a tree whose nodes are the combinations of active constraints that define a multiparametric solution.[20] Hence, before constructing our problem and passing it to a multiparmetric programming solver, we would like to make it more computationally friendly. Fourier–Motzkin elimination is an algorithm that aims to that cause.

Given a constraint of form (26), assume that the vector η is the concatenation of the state, input and disturbance vector as well as the artificial variables generated from the robustification, $\eta = \left[x^T u^T d^T v^T z^T\right]^T$, then a state constraint of the problem can be rewritten in a compact form as:

$$\sum_{j=1}^{j=l} \sigma_{i,j}\eta_j \le \hat{x}, \tag{28}$$

where $l = 2n + 2m + p$ and $\sigma_{i,j}$ are the coefficients multiplying the vector η. Assume that we would like to eliminate the variable η_l. Then we can isolate this variable as follows:

$$\eta_l \le \frac{1}{\sigma_{i,l}}\left(\hat{x} - \sum_{j=1}^{j=l-1} \sigma_{i,j}\eta_j\right), \tag{29}$$

or

$$\eta_l \ge \frac{1}{\sigma_{i,l}}\left(\hat{x} - \sum_{j=1}^{j=l-1} \sigma_{i,j}\eta_{0|j}\right), \tag{30}$$

depending on the sign of the coefficient $\sigma_{i,j}$. By following this procedure we calculate an upper bound (see Equation 29) or a lower bound (see Equation 30) depending on the sign of the inequality. The Fourier–Motzkin elimination requires that this procedure is performed for all constraints that include η_l and calculates a series of lower and upper bounds for this particular variable. If a value for the variable that we would like to eliminate, η_l, which satisfies the conditions that can be found such that each upper bound exceeds each lower bound, then the variable can be eliminated. In our considered mpMPC application, we would like to eliminate the artificial state variables, z, because they do not play any role in our optimization problem. Nevertheless, due to the pairing of the lower and upper bounds in additional constraints, we increase the complexity of the formulation from a constraints standpoint. Thankfully, many of these constraints are redundant which we can eliminate.

4.2 | Constraint set reduction

The redundant constraints of the problem can be removed by solving a linear programming problem for each constraint that is checked. Assume that t is a scalar variable, and that we have a constraint set.

The redundancy of the ith constraint can be identified if we solve the following optimization problem:[21]

$$\begin{array}{ll} \min\limits_{\eta} & t \\ \text{s.t.} & \hat{g}_j\eta + \left\|\hat{g}_j\right\|_2 t \le b_j \quad \forall j \ne i \cdot \\ & \hat{g}_i\eta = b_i \end{array} \tag{31}$$

If the problem is feasible with $t > 0$, then the constraint is nonredundant since there exists a ball of radius t that can be created. If $t = 0$ then the constraint intersects the feasible region at a point and it is weakly redundant. If the problem is infeasible, the constraint does not play any role in the description of the feasible region. Hence, if the problem is feasible with $t = 0$ or it is infeasible, the ith constraint is removed from the problem. We repeat this preprocessing procedure for all constraints of the problem. A graphical representation of the constraint redundancy methodology is presented in Figure 1. After removing these unnecessary constraints, the problem can be passed to an mpQP solver to derive the uncertainty-immune control policy.

4.3 | Strategy summary

After performing the afore-described methodological and algorithmic steps, and by propagating the states using the state space model (1), the original problem is equivalent to an mpQP problem of the following form:[1]

$$\begin{array}{ll} \min\limits_{\omega} & \frac{1}{2}\omega^T Q\omega + \eta^T H^T\theta + \theta^T Q_\theta^T\theta \\ \text{s.t.} & A\omega \le b_i + F\theta \\ & P_A\theta \le P_b \end{array}, \tag{32}$$

where $Q \in \mathbb{R}^{2n\times 2n}$, $H \in \mathbb{R}^{2n\times 2m}$, $Q_\theta \in \mathbb{R}^{2m\times 2m}$, while $\omega = \left[u^T v^T\right]^T \in \mathbb{R}^{2n}$ is the vector of decision variables and $\theta = \left[x^T d^T\right] \in \mathbb{R}^{m+q}$ is the vector of uncertain parameters. The linear constraints of the problem, written in

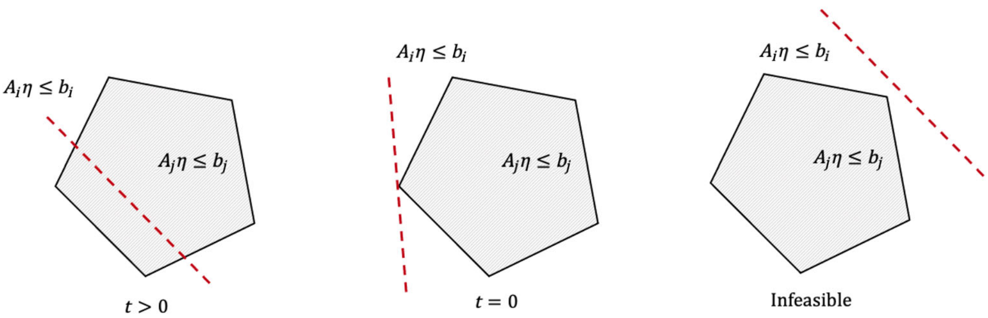

FIGURE 1 Graphical representation of identifying of redundant constraints in problem (7). Left to right: nonredundant constraint, weakly redundant constraint, redundant constraint.

compact form, are defined by the matrices A, F, and the vector b, while the matrices P_A and P_b define the boundaries of the uncertain parameters. Many mpQP algorithm exist to solve such a problem. Many of them are available to be utilized through an off-the-shelf fashion. Ultimately, the computational performance of the algorithm depends on the ability of the underlying multiparametric optimization algorithm to solve the problem. Recently, our group has published the PPOPT solver,[22] that among other features includes, parallel implementations of multiparametric optimization algorithms. The overall strategy presented in this work is presented in Table 1.

5 | PRACTICAL APPLICATIONS

5.1 | Numerical case study

Consider the following system described by the following uncertain discrete-time model of form (1)

where $\bar{A} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$, and $\bar{B} = \begin{bmatrix} 0 \\ 0.6 \end{bmatrix}$.

The states are equal to the outputs of the system, which are assumed to be measurable at each time instant, k. The weights of the objective function of the mpMPC are $Q_R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $R = 0.01$, and $P_R = \begin{bmatrix} 2.633 & 1.649 \\ 1.649 & 2.694 \end{bmatrix}$.

We consider a prediction horizon, $N = 2$, and that bounds for the states and inputs as:

$$-7 \le x_k \le 7, \quad \forall k = 0,1,2. \tag{33}$$

$$-1 \le u_k \le 1, \quad \forall k = 0,1. \tag{34}$$

The uncertainty ranges for the matrices of the states and inputs are, $\epsilon_\alpha = \epsilon_\beta = 0.2$. The objective is to stir the system to the original as presented in formulation (7).

Based on the presented steps of our strategy, we firstly calculate the terminal set $\mathcal{T}$, and include it in the constraint set of the problem. The system is subsequently robustified, and after the implementation of the Fourier–Motzkin elimination and the removal of redundant constraints the desired control policy is calculated. The overall

TABLE 1 Framework for the solution of multiparametric quadratic programming problems with multiplicative and additive uncertainty.

Step 1: Formulate the general problem MPC formulation as presented in Equation (7).
Step 2: Calculate a robust control invariant set through (27) and set it as the terminal set, $\mathcal{T}$.
Step 3: Robustify the state constraints for all stages by constructing their robust
counterpart reformulation (26).
Step 4: Remove the artificial state variables through Fourier–Motzkin elimination.
Step 5: Remove redundant constraints by solving the linear programming problem (31)
for each constraint.
Step 6: Solve a single multiparametric quadratic programming problem to obtain
an uncertainty-immune control policy.

calculation time for all the above steps was 20.95 s. The system is then simulated in closed-loop for two arbitrarily chosen values for A and B, starting from the initial point, $x_0 = [-1,4]^T$. Figure 2 shows the evolution of the states in closed-loop over time.

We observe that the controller manages to stir the system to the original. Furthermore, from the same initial point, we perform 50 closed-loop simulations, during each on of them the system matrices are randomly varied at every time step. Figure 3 demonstrates the effectiveness of the controller to guarantee feasibility for all of these realizations of the time varying system, where the evolution of the states for all scenarios is plotted on top of the map of critical regions.

Finally, to further illustrate the benefits of the proposed approach, we arbitrarily choose the initial point $x_0 = [1.3,0.7]^T$ and random values for the system matrices A and B. We successfully perform a closed-loop simulation using the proposed approach. However, when we implemented a classical (i.e., nominal—without any robustification) controller, we see that the applied nominal policy fails to achieve the operational target and system becomes infeasible after 21 time steps. These behavior of the robust and the nominal systems is presented in Figure 4.

Nevertheless, there is a price that we need to pay to guarantee feasibility: the reduction of the overall feasible space. Specifically,

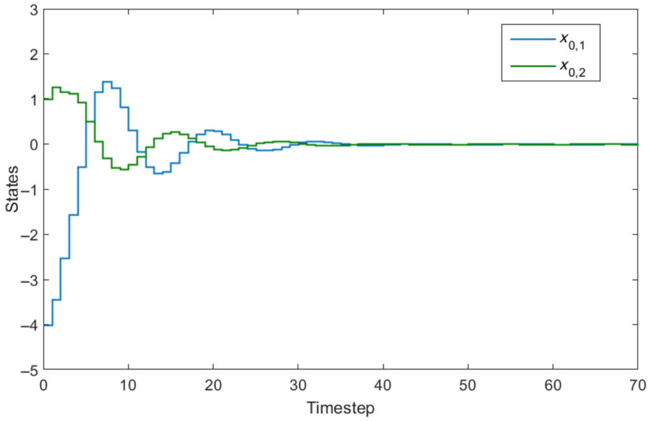

FIGURE 2 Closed-loop simulation of the numerical case study model by applying the robustified control policy.

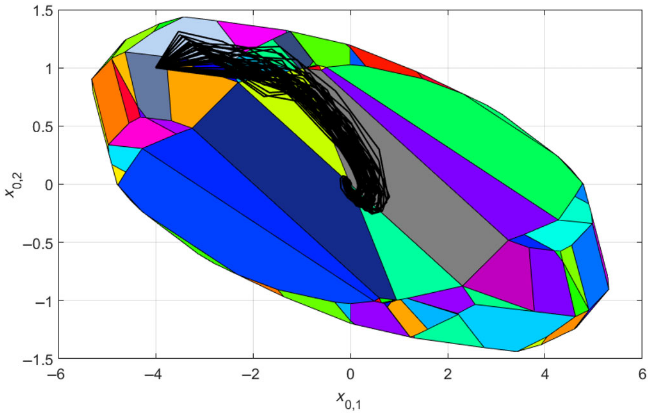

FIGURE 3 Multiple closed-loop simulations of the numerical case study model by applying the robustified control policy while varying the matrices of the model at each time step.

from the three illustrations presented in Figure 5, we see the feasible state space of the system for three separate cases of ϵ_α and ϵ_β, starting from the 10% uncertainty in the system's matrices up to 20% uncertainty. On the one hand, we enforce constraint satisfaction for a wider range of values for the system which is particularly beneficial for systems where we can describe them with high accuracy, but we are allowed to operate our system in a shrank range of states. In the end, the choice of the desired uncertainty range that needs to be respected depends on the practitioner and by taking into account the characteristics of the process, for example when a system is operated at critical conditions.

5.2 | Chemical engineering case study

In this case study, we consider the operation of a bioreactor with substrate inhibition. The objective of this problem is to demonstrate how the developed robust optimization strategy can be used to handle uncertainty in practical applications arising in engineering. The

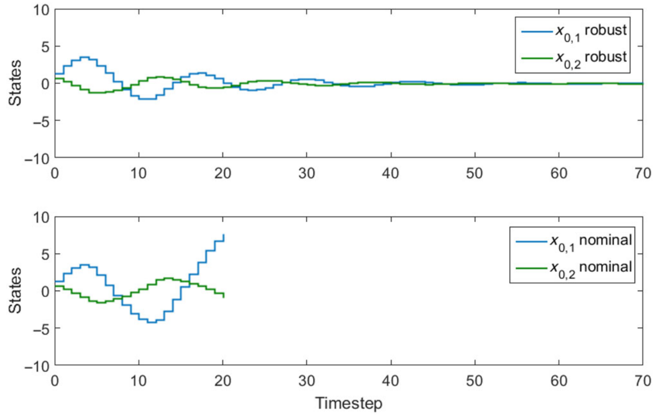

FIGURE 4 Closed-loop simulation of the system considered in the numerical case study. Top figure: robust solution, lower figure: solution without robustification.

information about the considered high fidelity model is adopted from Reference 23. The aforementioned reactor produces biomass, X, by using substrate, S. Due to the growth dynamics, large values of substrate inhibits the production of biomass. The following dynamic model of the reactor describes its operation:

$$\begin{aligned}\frac{dX}{dt} &= (\mu - D)X \\ \frac{dS}{dt} &= (S_F - S)D - \frac{\mu X}{Y} \\ \mu &= \frac{\mu_{\max} S}{k_m + S + k_1 S^2},\end{aligned} \tag{35}$$

where μ is the specific growth rate, D the dilution rate, S_F the concentration of the substrate feed, $\mu_{\max}$ the maximum specific growth rate, Y the biomass yield, and k_m and k_1 kinetic constants. Table 2 includes the parameters used in the model.

The goal of this control problem is to regulate the biomass and subsrtate concentrations around the steady state of $X_s = 1.512$ %w/v and $S_s = 0.175$ %w/v. Directly using the dynamic model in a multi-parametric programming formulation is challenging. Hence, we develop a surrogate linear state space model of the original high fidelity model. That is achieved by performing step changes to the dilution rate, which is our manipulated variable, which affects the biomass and substrate concentrations which are our controlled variables. MATLAB's System Identification Toolbox is used for this purpose. The fitted state space model is able to capture the behavior the high fidelity model as shown in Figure 6. Note that the figure is plotted by subtracting the values at desired for operation steady state. The nominal matrices of the model are $\bar{A} = \begin{bmatrix} 0.918 & 0.425 \\ 0.018 & 0.104 \end{bmatrix}$ and $\bar{B} = \begin{bmatrix} -0.027 \\ 0.299 \end{bmatrix}$.

Nevertheless, there is a 19% and 16% deviation of the two outputs, respectively, between the two models, due to the linear and discrete approximation of the surrogate model. Based on that, we define that $\epsilon_\alpha = \epsilon_\beta = 0.2$ to take into account the uncertainty when designing the robust controller. Following the aforementioned strategy, we

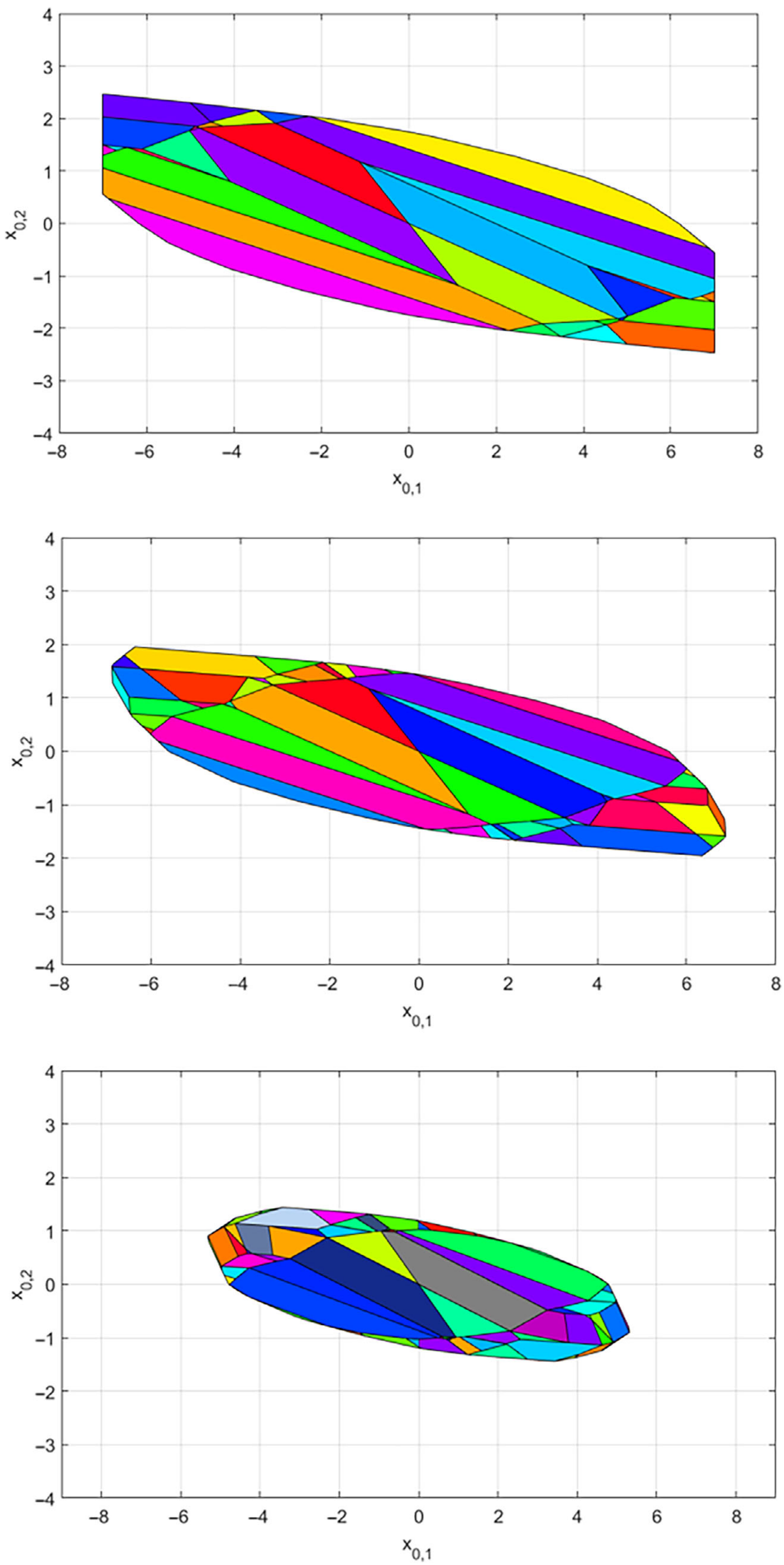

FIGURE 5 Comparison of the feasible space of the problem based on the for different values of ϵ_α and ϵ_β that are used for the robustification against the uncertainty. Top figure: $\epsilon_\alpha = \epsilon_\beta = 0.1$, middle figure: $\epsilon_\alpha = \epsilon_\beta = 0.15$, lower figure $\epsilon_\alpha = \epsilon_\beta = 0.2$.

TABLE 2 Chemostat operational information adopted from Reference 23.

Process description	Notation	Value
Nominal substrate concentration (%w/v)	S_F	4
Nominal dilution rate (h^{-1})	D_s	0.30
Inhibition constant (%w/v)	k_1	0.45
Saturation constant (%w/v)	k_m	0.12
Maximum specific growth rate (h^{-1})	μ_{max}	0.53
Biomass yield ($g\ g^{-1}$)	Y	0.40

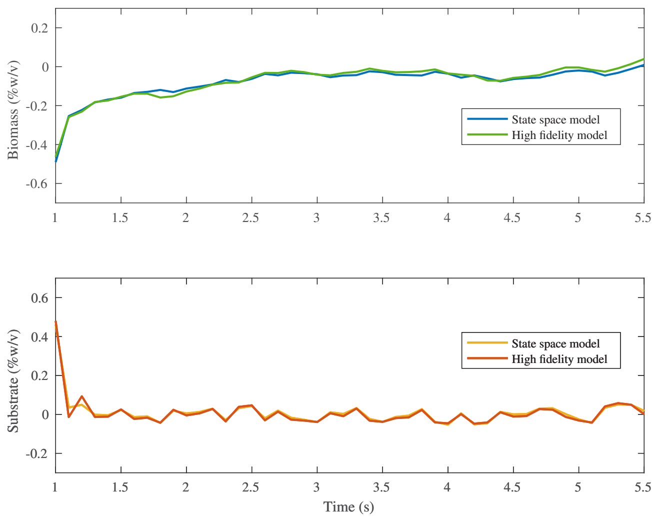

FIGURE 6 Comparison of the high fidelity bioreactor model with the fitted linear state space model. Top figure: biomass, lower figure: substrate.

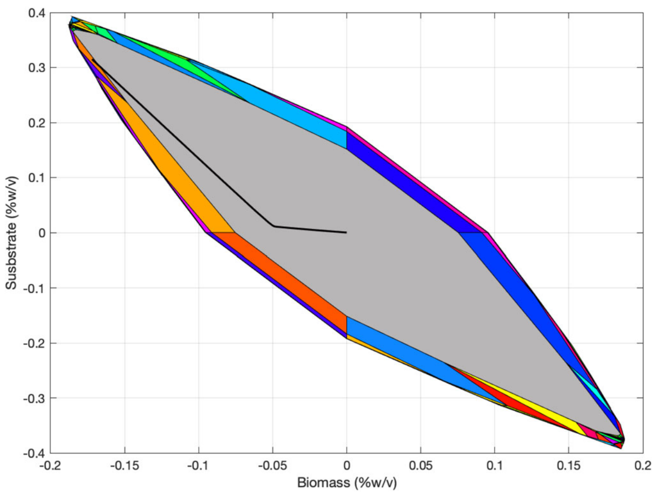

FIGURE 7 Closed-loop simulation of the of the high fidelity dynamic model using the robustified control policy. The evolution of the states is illustrated on top of the map of critical regions.

develop the robust explicit control actions. The weights of the objective function of the mpMPC are $Q_R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $R = 0.01$, and $P_R = \begin{bmatrix} 3.575 & 1.222 \\ 1.222 & 1.580 \end{bmatrix}$. A prediction horizon of 4 is considered. The explicit solution is comprised by 520 critical regions. The derived control actions are applied in closed-loop to the original high fidelity bioreactor model. The closed-loop simulation of the high fidelity model using the robustified control policy is illustrated in Figure 7.

From the aforementioned simulation it is shown that the controller can drive the system to the desired operating point. As also demonstrated in the numerical case study, a balance needs to be struck between the desired level of robustness and the reduction of the feasible state space as a result of the robustification. This level depends on the application considered at each case.

6 | CONCLUSION

In this work, we developed a multiparametric programming-based strategy for the regulation of constrained uncertain discrete-time linear systems in the presence of additive and multiplicative uncertainty. Specifically, we robustified the constraint set of the problem and calculated a suitable terminal set that guarantee an uncertainty-immune operation of the considered system. Also, we enhanced the computational performance of the underlying explicit MPC problem by removing redundant constraints and artificial variables. A key benefit of the approach is that the derivation of the explicit state feedback control law is achieved by solving a single multiparametric programming problem. We demonstrated the algorithmic steps and the application of the proposed approach through a numerical case study as well as through a chemical engineering problem. Our findings show that the derived control laws manage to regulate the system in the presence of any considered value of uncertainty within the considered uncertainty range. As far as the computational cost related to deriving the robust solutions is concerned, the considered case studies required less than a minute each to derive the full explicit solution. However, there is price that is paid when deriving the robust solution, namely the inherent conservativeness and the increased size of the problem stemming from the robustification procedure. Moving forward, there are multiple research avenues that can followed and can be based on this work since this can act as the backbone of methodologies for simultaneous design, control and scheduling, multiobjective MPC and many more applications that utilize multiparametric optimization.

AUTHOR CONTRIBUTIONS

Iosif Pappas: conceptualization (equal); data curation (lead); formal analysis (lead); investigation (lead); methodology (equal); software (equal); validation (lead); visualization (lead); writing – original draft (lead); writing – review and editing (lead). **Nikolaos A. Diangelakis:** conceptualization (equal); data curation (supporting); formal analysis (supporting); investigation (supporting); methodology (equal); project administration (supporting); software (equal); supervision (supporting);

validation (supporting); writing – original draft (supporting); writing – review and editing (supporting). **Efstratios N. Pistikopoulos:** conceptualization (equal); formal analysis (supporting); funding acquisition (lead); investigation (supporting); methodology (supporting); project administration (lead); supervision (lead); writing – original draft (supporting); writing – review and editing (supporting).

ACKNOWLEDGMENTS

The financial support from the Department of Energy and the Clean Energy Smart Manufacturing Innovation Institute, Energy Efficient Operation of Air Separation Processes: DE-EE0007613, 4550 G WA324 as well as from the Texas A&M Energy Institute is gratefully acknowledged.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Iosif Pappas https://orcid.org/0000-0003-2336-1848
Nikolaos A. Diangelakis https://orcid.org/0000-0003-4803-6136
Efstratios N. Pistikopoulos https://orcid.org/0000-0001-6220-818X

REFERENCES

1. Pappas I, Kenefake D, Burnak B, et al. Multiparametric programming in process systems engineering: recent developments and path forward. *Front Chem Eng*. 2021;2. doi:10.3389/fceng.2020.620168
2. Sakizlis V, Kakalis NM, Dua V, Perkins JD, Pistikopoulos EN. Design of robust model-based controllers via parametric programming. *Automatica*. 2004;40(2):189-201.
3. Kerrigan EC, Maciejowski JM. Feedback min-max model predictive control using a single linear program: robust stability and the explicit solution. *Int J Robust Nonlinear Control*. 2004;14(4):395-413.
4. Rodríguez-Ayerbe P, Olaru S. On the disturbance model in the robustification of explicit predictive control. *Int J Syst Sci*. 2013;44(5):853-864.
5. Rubagotti M, Barcelli D, Bemporad A. Robust explicit model predictive control via regular piecewise-affine approximation. *Int J Control*. 2014;87(12):2583-2593.
6. Tejeda-Iglesias M, Lappas NH, Gounaris CE, Ricardez-Sandoval L. Explicit model predictive controller under uncertainty: an adjustable robust optimization approach. *J Process Control*. 2019;84:115-132.
7. Wang K, Jiang Y, Oravec J, Villanueva ME, Houska B. Parallel explicit tube model predictive control. Paper presented at: 2019 IEEE 58th Conference on Decision and Control (CDC). IEEE; 2019; Nice, France: 7696-7701.
8. Bemporad A, Borrelli F, Morari M. Min-max control of constrained uncertain discrete-time linear systems. *IEEE Trans Automat Control*. 2003;48(9):1600-1606.
9. Kouramas KI, Panos C, Fa NP, Pistikopoulos EN. An algorithm for robust explicit/multi-parametric model predictive control. *Automatica*. 2013;49(2):381-389.
10. Sheikhbahaei R, Alasty A, Vossoughi G. Robust fault tolerant explicit model predictive control. *Automatica*. 2018;97:248-253.
11. Oberdieck RH. *Theoretical and Algorithmic Advances in Multi-Parametric Optimization and Control*. Ph.D. thesis. Imperial College London; 2016.
12. Sun M, Villanueva ME, Pistikopoulos EN, Chachuat B. Methodology for robust multi-parametric control in linear continuous-time systems. *J Process Control*. 2019;73:58-74.
13. Nguyen NA, Olaru S, Rodriguez-Ayerbe P, Bitsoris G, Hovd M. Explicit robustness and fragility margins for linear discrete systems with piecewise affine control law. *Automatica*. 2016;68:334-343.
14. Pappas I, Diangelakis NA, Oberdieck R, Pistikopoulos EN. A robust optimization strategy for explicit model predictive control. In: Yamashita Y, Kano M, eds. *Computer Aided Chemical Engineering*. Vol 49. Elsevier; 2022:409-414.
15. Pappas I, Diangelakis NA, Pistikopoulos EN. Multiparametric/explicit nonlinear model predictive control for quadratically constrained problems. *J Process Control*. 2021;103:55-66.
16. Ben-Tal A, Nemirovski A. Robust solutions of linear programming problems contaminated with uncertain data. *Math Program*. 2000;88(3):411-424.
17. Kerrigan EC. *Robust Constraint Satisfaction: Invariant Sets and Predictive Control*. Ph.D. thesis. University of Cambridge; 2001.
18. Blanchini F. Set invariance in control. *Automatica*. 1999;35(11):1747-1767.
19. Bertsekas D. Infinite time reachability of state-space regions by using feedback control. *IEEE Trans Automat Control*. 1972;17(5):604-613.
20. Pappas I, Diangelakis NA, Pistikopoulos EN. The exact solution of multiparametric quadratically constrained quadratic programming problems. *J Global Optim*. 2021;79(1):59-85.
21. Pistikopoulos EN, Diangelakis NA, Oberdieck R. *Multi-Parametric Optimization and Control*. John Wiley & Sons; 2020.
22. Kenefake D, Pistikopoulos EN. Ppopt-multiparametric solver for explicit mpc. In: Montastruc L, Negny S, eds. *Computer Aided Chemical Engineering*. Vol 51. Elsevier; 2022:1273-1278.
23. Eaton JW, Rawlings JB. Model-predictive control of chemical processes. *Chem Eng Sci*. 1992;47(4):705-720.

How to cite this article: Pappas I, Diangelakis NA, Pistikopoulos EN. Explicit model predictive control through robust optimization. *AIChE J*. 2023;69(10):e18172. doi:10.1002/aic.18172