# A Partial Multiparametric Programming method for Model Predictive Control

Dustin Kenefake [a,b], Sahithi Srijana Akundi [a,b] and Efstratios N. Pistikopoulos [a,b1]

[a] Texas A&M Energy Institute, Texas A&M University, College Station, TX, USA 77843

[b] Texas A&M Department of Chemical Engineering, Texas A&M University, College Station, TX, USA 77843

*Abstract*

Multiparametric programming is a paradigm of manipulating and solving parameterized mathematical programs which allows for explicit algebraic solutions to optimization problems. This has found great interest in the literature, from explicit model predictive control to data science applications, and allows one to address an entire family of associated optimization problems simultaneously. From the multiparametric viewpoint, many optimization-based applications in control and engineering can be succinctly stated as a single multiparametric program, such as with model predictive control and state estimation. In some cases, the full explicit solution of a multiparametric program is not computationally feasible to generate, leading to the development of partial explicit multiparametric methods that can be utilized to reduce the computational burden of online optimization problems while not fully solving the explicit solution of the multiparametric program. Here, a partial explicit method is presented that allows for feasible constraint pruning from parameterized optimization problems by inspecting the optimality conditions of the multiparametric program and deriving overestimators of constraint optimality domains in the parametric space. These overestimators are then used to remove constraints from consideration in the online optimization problem, including feasible constraints, while retaining the exact solution. The utility of this methodology is demonstrated via the application of optimal control on a non-isothermal CSTR system, showing an average decrease in solve time of 40% to solve the online model predictive control problem.

*Keywords*

Multiparametric Programming, Model Predictive Control, Optimization Methods

## Introduction

Multiparametric programming has received much interest in the control literature since it was shown that the model predictive controller (MPC) problem can be reformulated into a multiparametric program (Pistikopoulos et al., 2021; Bemporad et al., 2000). Through the use of multiparametric programming methods, explicit algebraic solutions of the mpMPC (multiparametric MPC) can be computed; these are typically referred to as explicit MPC in the literature (Pappas et al., 2021). Explicit MPC has found many applications in automotive, aeronautical, and chemical industries (Pappas et al., 2021). Dedicated solvers have been developed to solve particular classes of the multiparametric programs that arise in mpMPC (Kenefake and Pistikopoulos, 2022; Oberdieck et al., 2016; Herceg et al., 2013). However, in some cases the complexity of the mpMPC makes solving the full explicit solution prohibitive – that is, collecting all critical regions that compose a solution – leading to the recent development of what will be referred to as partial multiparametric methods in this work. The general form of a continuous multiparametric program can be seen in eqn. 1, with $f(x,\theta)$ denoting the parameterized objective, $g(x,\theta)$ denoting the parameterized equality constraints, $h(x,\theta)$ denoting the parameterized inequality constraints, and $\Theta$ denoting the space of parameters.

$$\min_x f(x,\theta) \tag{1a}$$
$$g(x,\theta) = 0 \tag{1b}$$
$$h(x,\theta) \leq 0 \tag{1c}$$
$$\theta \in \Theta \tag{1d}$$
$$x \in \mathbb{R}^m, \theta \in \mathbb{R}^n \tag{1e}$$

Recently, many partial multiparametric strategies have been explored in the literature such as with the development of the custom Quadratic Program (QP) solver, qpOASES, which is specialized for solving control type problems where the previous solution is utilized as a hot start to the next optimization problem utilizing multiparametric methodologies (Ferreau et al., 2008). A partial explicit approach used by Katz and Pistikopoulos (2020) is based on sampling the parameter space $\Theta$, to collect the critical regions that cover large volumes of the parameter space, and incorporated as hot

[1] Corresponding author: Efstratios N. Pistikopoulos (E-mail: stratos@tamu.edu).

start procedure for the online optimization problem. Methods based on caching critical regions have been implemented, such as in Kenefake et al. (2022), where given a parameter realization, $\theta_0 \in \Theta$, if it is not inside any cached regions, the optimization problem is solved. Using multiparametric programming techniques, the critical region corresponding to the point $\theta_0$ is generated and added to cache.

In this work, a partial multiparametric method that generates regions in the parameter space that contain all points where each inequality constraint participates in an optimal active set is proposed. These regions are then used during the real time optimization step to remove constraints that cannot participate in the optimal solution from consideration. By removing constraints from the online optimization problem, the computational complexity of the optimization is reduced. The utility of this approach is then demonstrated in the optimal control of a non-isothermal continuously stirred tank reactor (CSTR).

## Methods

The Methods section is separated into three separate subsections. The first subsection describes the optimality conditions of a multiparametric program, where the KKT (Karush-Kuhn-Tucker) conditions are derived and the complementarity conditions are reformulated to avoid unnecessary nonlinear constraints by using indicator constraints. The second subsection describes the general behavior of domains of optimal constraints in the parameter space and describes the procedure of generating overestimating regions of these domains. The last subsection describes how the information obtained from these regions can be used in real time to reduce the computational overhead of the optimization problem when a parameter has been realized, e.g. $\theta_0 \in \Theta$, while still obtaining the exact optimal solution.

### *Multiparametric KKT Conditions*

The KKT conditions define the necessary conditions for the solution of an optimization problem (Karush, 2014; Kuhn and Tucker, 2014). The multiparametric KKT conditions can be derived for the general multiparametric program as seen in eqn. 2, with eqns. 2b, 2c, 2e referring to primal feasibility, eqn. 2a referring to the stationarity condition, eqn. 2d referring to the complementarity conditions, eqn. 2f referring to the dual feasibility, and eqn. 2g referring to the parameter feasibility.

$$\nabla_x f(x,\theta) + \langle \nabla_x g(x,\theta), \mu \rangle + \langle \nabla_x h(x,\theta), \lambda \rangle = 0 \quad (2a)$$
$$g(x,\theta) = 0 \quad (2b)$$
$$h(x,\theta) + \mathbf{I}_{m\times m} s = 0 \quad (2c)$$
$$\langle \lambda, s \rangle = 0 \quad (2d)$$
$$s \geq 0 \quad (2e)$$
$$\lambda \geq 0 \quad (2f)$$
$$\theta \in \Theta \quad (2g)$$

In the case of mpLP (multiparametric Linear Programs) and mpQP (multiparametric Quadratic Programs), all constraints are linear except for the complementarity conditions

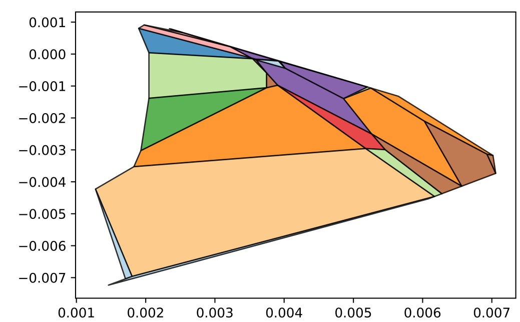


Figure 1: Subset of the full explicit solution from Fig. 2, where a particular constraint appears in the optimal active set. It can be seen that this is a non-convex polytope.

(eqn. 2d). This is ameliorated via introducing an auxiliary binary variable $y$ for every inequality constraint, and reformulating the complementarity conditions via the indicator constraint formulation as shown in eqn. 3. In the case of mpLPs and mpQPs with this reformulation, the mpKKT conditions are mixed integer linear representable, allowing the utilization of a modern mixed integer linear programming (MILP) solver, such as Gurobi, to solve this problem (Gurobi Optimization, LLC, 2022). Alternatively, a Big-M formulation can be utilized for the reformulation.

$$\langle \lambda, s \rangle = 0 \implies \lambda_i s_i = 0 \implies \begin{array}{l} y_i = 1 \implies s_i = 0 \\ y_i = 0 \implies \lambda_i = 0 \end{array} \quad (3)$$

### *Constraint Domain Boundary*

There are many options to represent the domain of parameters where a constraint is optimal for a given multiparametric program, $\mathcal{X}_i \subseteq \Theta$. As the computational cost of solving the bilevel substituted problem described in eqn. 6 is not computationally trivial, there is a trade off between tightness of the domain description and the computational time to describe this domain. The full description can be trivially constructed from the full explicit solution which requires the computationally difficult solution of a multiparametric program to be solved. On the other hand, it does not require any prior computation to consider that all constraints could be active at any parameter realization. An example of an explicit solution and the domain of parameters where a constraint is optimal can be seen in Figs. 1 and 2. The space of parameters, $\theta$, where any given constraint appears in the optimal active set is not necessarily convex, as can be seen in Fig. 1.

The AABB (Axis Aligned Bounding Box) was selected as an overestimating domain for the set of parameters where a constraint is optimal. This domain description was chosen for three primary reasons: 1) Only $2n$ optimization problems need to be solved to get the full description, 2) AABBs require very little storage to save the full description, 3) checking if a point is inside of an AABB is computationally cheap. It should be noted that an AABB is simply a hyper rectangle where the faces are in line with the coordinate axes, here intersected with $\Theta$, giving $\mathcal{X}_i \subseteq \text{AABB}_i \subseteq \Theta$. Compu-

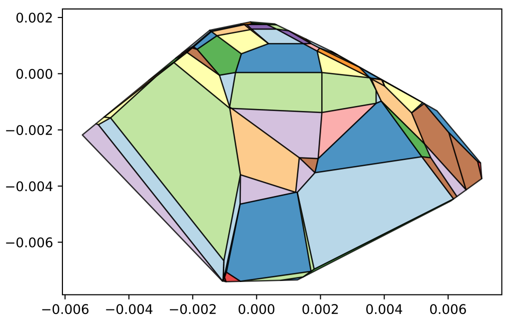


Figure 2: The full explicit solution of an mpQP, showing all critical regions and refining the optimality of each component.

tational savings for the online problem are expected when $\text{AABB}_i \subset \Theta$, as that implies there are $\theta \in \Theta$ such that constraint $i$ is not in the optimal basis solution.

$$\text{AABB} = \{\theta \in \Theta : \underline{\theta_i} \leq \theta_i \leq \overline{\theta_i}\} \tag{4}$$

The axial bounds of each AABB can be computed by solving the bilevel optimization problem as described in eqn. 5; however, this form cannot be solved with many modern optimization solvers and requires the utilization of equilibrium programming, which is not computationally attractive. Here, eqn. 5 shows the optimization problem for finding the minimum bounds of the $j^{\text{th}}$ parameter, $\theta_j$, given inequality constraint $k^{\text{th}}$ is active, such that the resulting optimization variable, $x$, is still optimal with respect to the lower level optimization problem.

$$\begin{aligned}
&\min \quad \theta_j && \text{(5a)}\\
&\text{s.t.} \quad h_k(x,\theta) = 0 && \text{(5b)}\\
&\qquad x \in \arg\min f(x,\theta) && \text{(5c)}\\
&\qquad\qquad \text{s.t.} g(x,\theta) = 0 && \text{(5d)}\\
&\qquad\qquad\quad h(x,\theta) \leq 0 && \text{(5e)}\\
&\qquad\qquad\quad \theta \in \Theta && \text{(5f)}\\
&\qquad x \in \mathbb{R}^m, \theta \in \mathbb{R}^n && \text{(5g)}
\end{aligned}$$

This bilevel optimization problem is reformulated into a single level optimization problem by incorporating the mp-KKT conditions into the upper level optimization problem, similar to the reformulation of the flexibility index problem as proposed by Grossmann and Floudas (1987). The computational performance of this formulation can be improved by removing constraints that are redundant $\forall \theta \in \Theta$, and by including valid bounds on $x$, $s$, $\theta$, $\lambda$, $\mu$. The procedure for calculating bounds on feasibility for $x$, $s$, and $\theta$ via simple optimization-based methods is well-known in the literature.

$$\begin{aligned}
&\min \quad \theta_j && \text{(6a)}\\
&\text{s.t.} \nabla_x f(x,\theta) + \langle \nabla_x g(x,\theta), \mu \rangle + \langle \nabla_x h(x,\theta), \lambda \rangle = 0 && \text{(6b)}\\
&g(x,\theta) = 0 && \text{(6c)}\\
&h(x,\theta) + \mathbf{I}_{m\times m} s = 0 && \text{(6d)}\\
&y_l = 1 \implies s_l = 0, \forall l \in \{1,\ldots,p\} && \text{(6e)}\\
&y_l = 0 \implies \lambda_l = 0, \forall l \in \{1,\ldots,p\} && \text{(6f)}\\
&s \geq 0 && \text{(6g)}\\
&\lambda \geq 0 && \text{(6h)}\\
&\theta \in \Theta && \text{(6i)}\\
&s_k = 0 && \text{(6j)}\\
&x \in \mathbb{R}^m, \theta \in \mathbb{R}^n, y \in \{0,1\}^p, s \in \mathbb{R}^p_+, \lambda \in \mathbb{R}^p_+ && \text{(6k)}
\end{aligned}$$

If the problem is infeasible when solving for the bounds of the AABB of an inequality constraint, this implies that the constraint does not participate in any optimal active set for any parameter realization $\theta_0 \in \Theta$. It immediately follows that this constraint can be removed from the model formulation for all problem realizations, even if that constraint is non-redundant.

*Online Procedure*

The AABB overestimators can then be incorporated into an online procedure where, if the parameter realization $\theta_0 \in \Theta$ is not contained in the $i^{\text{th}}$ AABB, then this constraint can be removed from consideration from the online optimization problem. Additionally, it is clear that if $\theta_0 \notin \Theta$, then the optimization problem is infeasible.

$$\begin{aligned}
&\min_x \quad f(x,\theta_0) && \text{(7a)}\\
&\text{s.t. } g(x,\theta_0) = 0 && \text{(7b)}\\
&\quad h_i(x,\theta_0) \leq 0, \quad \forall i \in \{i \in \{1,\ldots,p\} : \theta_0 \in AABB_i\} && \text{(7c)}
\end{aligned}$$

The solution of this optimization problem must be checked for feasibility of the constraints that was removed, ensuring that a false feasible solution is not found (e.g. a situation when a parameter realization $\theta_0$ generates an infeasible optimization problem). This represents the blank regions surrounding the explicit solution in Fig. 2. This can be done by simply substituting the solution of the optimization problem $x^*$ and the parameter realization $\theta_0$ into the multiparametric program constraints, and checking if the resulting constraints are consistent. If the substituted constraints are consistent, then the optimal solution to the optimization problem has been verified. On the other hand, if the constraints are not consistent, then the original optimization problem with $\theta_0$ substituted and all constraints included is also infeasible.

$$h_j(x^*,\theta_0) \leq 0, \quad \forall j \in \{j \in \{1,\ldots,p\} : \theta_0 \notin AABB_j\} \tag{8}$$

## Example Problem

The methods developed so far can be demonstrated on a simple mpQP, as shown in eqn. 9. Here, $\Theta$ is real numbers between 0 and 1 inclusive, $\Theta = [0,1]$.

$$\begin{aligned}
&\min_{x} \quad \frac{1}{2}x^2 && \text{(9a)}\\
\text{s.t.} \quad & x - 1 - \theta \leq 0 && \text{(9b)}\\
& -x + \theta \leq 0 && \text{(9c)}\\
& \theta \in \Theta && \text{(9d)}
\end{aligned}$$

Firstly, the AABBs for each constraint are generated by solving the optimization problem as seen in eqn. 10. To find the lower bound of the $k^{\text{th}}$ constraint of the $i^{\text{th}}$ parameter, the following optimization problem, as shown in eqn. 10, is solved for each parameter of interest, $\theta_j$, and inequality constraint. To find the upper bound, the direction of the optimization problem can simply be changed to maximization. As there are not any equality constraints in this multiparametric program, the terms relating to $g(x,\theta)$ and $\mu$ are dropped.

$$\begin{aligned}
&\min \quad \theta_j && \text{(10a)}\\
\text{s.t.} \quad & x + \lambda_1 - \lambda_2 = 0 && \text{(10b)}\\
& x - 1 - \theta + s_1 = 0 && \text{(10c)}\\
& -x + \theta + s_2 = 0 && \text{(10d)}\\
& y_l = 1 \implies s_l = 0, \forall l \in \{1,2\} && \text{(10e)}\\
& y_l = 0 \implies \lambda_l = 0, \forall l \in \{1,2\} && \text{(10f)}\\
& s_1, s_2 \geq 0 && \text{(10g)}\\
& \lambda_1, \lambda_2 \geq 0 && \text{(10h)}\\
& 0 \leq \theta \leq 1 && \text{(10i)}\\
& s_k = 0 && \text{(10j)}\\
& x \in \mathbb{R}, \theta \in \mathbb{R}, y \in \{0,1\}^2, s \in \mathbb{R}^2, \lambda \in \mathbb{R}^2 && \text{(10k)}
\end{aligned}$$

It can be shown that both inequality constraints (9b and 9c) are non-redundant, meaning that they both form facets of the feasible space. However, when computing the AABBs for constraints 9b and 9c it was found that constraint 9b does not participate in any optimal active set, and can be removed from the online calculation $\forall \theta \in \Theta$. The AABB for constraint 9c was computed, and is shown in eqn. 11. In this example, $\Theta = \text{AABB}_{9c}$. However this procedure identified a constraint that was entirely surpurfluous while still being feasible: constraint 9b.

$$\text{AABB}_{\{9c\}} = \{\theta \in \Theta : 0 \leq \theta \leq 1\} \tag{11}$$

Now that the AABBs for the inequality constraints have been calculated, the online procedure is performed. A parameter realization of $\theta_0 = 0.5$ will be used to demonstrate the methodology. After utilization of the AABB to remove constraints, the resulting substituted problem, with $\theta_0$ substituted as $\theta$, is shown in eqn. 12. As constraint 9b has been eliminated via the previous analysis, it does not need to be included in the substituted problem. The solution to the substituted problem, eqn. 12, is $x^* = \langle 0.5 \rangle$.

$$\begin{aligned}
&\min_{x} \quad \frac{1}{2}x^2 && \text{(12a)}\\
& -x + 0.5 \leq 0 && \text{(12b)}
\end{aligned}$$

The solution $x^*$ is checked against the constraint 9b to ensure that $(x^*, \theta_0)$ is a feasible solution. Here, by the sensible use of optimality conditions and bounding, non-redundant constraints can be removed a priori from consideration of the substituted problems and thus simplifying the resulting optimization problem.

$$x^* - 1 - \theta_0 \leq 0 \rightarrow -1 \leq 0 \tag{13}$$

## Case Study: Non-Isothermal CSTR MPC

The proposed methodology will be demonstrated on an ideal non-isothermal CSTR adopted by Kazantzis and Kravaris (2000). The reaction considered in the CSTR is the irreversible reaction between sodium thiosulfate and hydrogen peroxide to generate sodium trithionate and water.

$$2Na_2S_20_3 + 4H_20_2 \rightarrow Na_2S_3O_6 + Na_2SO_4 + 4H_2O \tag{14}$$

For ease of notaion, $Na_2S_20_3$ and $H_20_2$ are denoted as $A$ and $B$ respectively. The Arrhenius rate law of $A$ is as shown in eqn. 15, where $k_0$ is the pre-exponential factor, $E$ is the activation energy, $R$ is the ideal gas constant, $T$ is the reactor temperature, and $c_A$ and $c_B$ denoting the reactor concentrations.

$$-r_A = k_0 \exp\left(\frac{-E}{RT}\right) c_A c_B \tag{15}$$

With the assumption of constant mixture density, volume, and feed composition ($2c_A(t) = c_B(t)$), the dynamic model of the CSTR is represented by the following system of ordinary differential equations, as stated in Pappas et al. (2020):

$$\begin{aligned}
\frac{dc_A}{dt} &= \frac{F}{V}(c_{A,in} - c_A) + r_A && \text{(16a)}\\
\frac{dT}{dt} &= \frac{F}{V}(T_{in} - T) + \frac{(\Delta H)_R}{\rho c_p} r_A - \frac{U_A}{V \rho c_p}(T - T_j) && \text{(16b)}
\end{aligned}$$

Considering a nominal value of the manipulated variable of $\frac{F}{V} = .2\,\text{s}^{-1}$, there are three steady states of the process system. The most economically advantageous steady state was found to be at $c_A = 0.667$, and $T_s = 308.499$ as considered by Pappas et al. (2020) (as a note, this steady state is unstable). Deviation variables are used to shift the desired steady state to the origin, as shown in eqn. 17.

$$x = \begin{bmatrix} c_A - c_{A_{ss}} \\ T - T_{ss} \end{bmatrix} \qquad u = \frac{F}{V} - \frac{F}{V}\Big|_{ss} \tag{17}$$

The dynamic model was linearized and discretized with a time step of 0.2 s assuming zero-order hold, and the resulting discrete linear system that represents the dynamic model are shown in eqn. 18.

$$x_{k+1} = \begin{bmatrix} 0.9194 & -0.0013 \\ 5.8554 & 1.1394 \end{bmatrix} x_k + \begin{bmatrix} 0.0685 \\ -6.9670 \end{bmatrix} u_k \tag{18}$$

The multiparametric program for this can be effectively stated in eqn. 19, with the initial state being equal to the the uncertain parameters,$\theta_0$ , as the state is not known ahead of time. This is a multiparametric program in the class mpQP, as all constraints are affine and the objective is both convex and quadratic. It should be noted that many linear MPC problems can be trivially converted into the mpMPC form. Here the MPC is set with a control and output horizon of $N = 40$, where $Q = \begin{bmatrix} 1 & 0 \\ 0 & 0.05 \end{bmatrix}$, $R = \begin{bmatrix} 10 \end{bmatrix}$, and $A$, $B$ are as presented in eqn. 18.

$$\min_{u,x} \quad x_N^T P x_N + \sum_{i=0}^{N-1} \left( x_i^T Q x_i + u_i^T R u_i \right) \tag{19a}$$

$$x_0 = \theta_0 \tag{19b}$$

$$x_{t+1} = A x_t + B u_t, \quad \forall t \in \{0, \ldots, N-1\} \tag{19c}$$

$$\underline{u} \leq u_t \leq \overline{u}, \quad \forall t \in \{0, \ldots, N\} \tag{19d}$$

$$\underline{x} \leq x_t \leq \overline{x}, \quad \forall t \in \{0, \ldots, N-1\} \tag{19e}$$

$$\theta \in \Theta \tag{19f}$$

Here, the mpMPC has 240 inequality constraints and, after equality constraints are eliminated, 40 variables relating to the $u_t$ variables. After removing constraints that are redundant for every parameter realization $\theta \in \Theta$, 116 constraints were removed from the problem formulation and only 124 inequality constraints remain. The AABBs for this mpMPC are generated based on the constraint domain boundary procedure with a computational cost of 602 s, on an Intel i7-4790 desktop with 16 GB of RAM utilizing Gurobi 9.5.2 to solve the MILPs. Inspecting the AABBs reveals an additional 42 constraints that never participate in any optimal active set combination for any realization of system state and thus can be removed, leaving 82 constraints that need to be considered for any realization of the MPC problem.

First, the case of an initial state of $x_0 = [0.417, 15]$ was considered. The AABB constraint pruning procedure was then applied with $\theta_0 = x_0$ to further reduce the number of constraints that must be accounted for by an additional 7 constraints. The online optimization problem that was to be solved contained only 75 of the original 240 constraints. This constitutes a reduction factor of approximately 3x in the number of constraints. The online optimization problem is solved, with the reduced constraint set provided by the partial multiparametric procedure, and the solution is exactly equal to the full optimization model with all 240 constraints. The full model takes 187 $\mu$s to compute while the reduced model takes 160 $\mu$s to compute when averaged over 10,000 runs, resulting in a reduction in solve time by 14%. The QPs considered in the case study were solved with the `quadprog` package in a Python 3.10 environment with a desktop that has Intel i7-4790 CPU and 16 GB of RAM, with the online procedure included in the time to solve. This solver utilizes the Goldfarb/Idnani dual algorithm to solve convex quadratic programs as proposed by Goldfarb and Idnani (1983). The system dynamics can be seen in Figure 3, with the solution of optimal control problem shown in Figure 4.

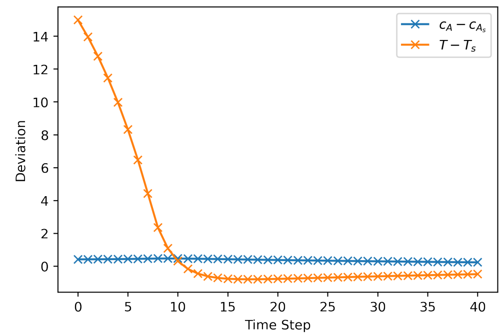


Figure 3: The evolution of the states of the linearized system, given the solution of the MPC problem.

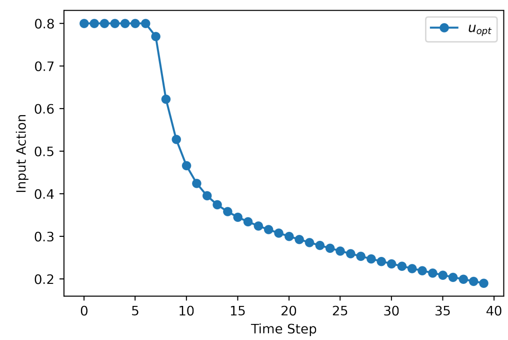


Figure 4: The solution of the MPC problem, $u_{opt}$, with $\theta_0 = x_0 = [0.417, 15]$ as the initial state.

An additional case study was performed where 100,000 points were sampled from $\Theta$, each relating to an initial state of the system. The corresponding MPC problems for each point were solved twice: once utilizing the standard MPC without constraint pruning, and once utilizing the constraint pruning AABB procedure. This amounts to 100,000 QPs being solved, one for each point. The standard MPC with all constraints considered took 38.55 s to compute all solutions, compared to 23.1 s when utilizing the partial multiparametric approach. This amounts to a $\approx 40\%$ reduction in time to solve for the entire problem set.

## Discussion

It can be seen that the method requires a relatively small amount of memory to store the AABBs as each one is stored in the form of two vectors of floating point numbers: the upper and lower bound vectors. For every inequality constraint, given $p$ inequality constraints and $m$ parameters, the storage required to save the AABBs for the entire problem is $2nmp$ bytes, where $n$ is the number of bytes per floating point number. For optimization problems stored via dense matrices, this is typically smaller than the optimization problem definition.

In the case where the reformulated bilevel optimization problem found in the Methods section is too computationally burdensome, instead of solving for the extent of the parameters $\theta_j$ subject to the optimality conditions, this can be relaxed to primal and parameter feasibility. While the range of the AABBs will expand, and therefore the efficiency of the constraint pruning will decrease, the computational complexity of the subproblems is greatly reduced. This is due to removal of the optimality-based constraints from the formulation that are typically the source of computational complexity.

## Conclusion

In this work, a partial explicit multiparametric programming method was developed that allows for the removal of feasible constraints from the online optimization problem while still resulting in the exact solution. This removal of feasible constraints was developed by inspection of the optimality conditions of multiparametric programs, and determination of overestimating regions in the parameter space $\Theta$ where a constraint could participate in the optimal active set of the parameter substituted problem, $\theta_0 \in \Theta$. The proposed approach was successfully applied to the control of a non-isothermal CSTR reactor at an unstable steady state, showing an average reduction of 40% in computational time to solve the online MPC problem. There are synergies between the partial multiparametric programming algorithms in the literature and this proposed algorithm that could be exploited to further reduce the time to solve. In the proposed method, the online optimization problem must still be solved at every step. These optimization problems could be accelerated via the initialization procedure proposed by Katz and Pistikopoulos (2020) and the caching procedure as proposed by Kenefake et al. (2022) to reduce the computational burden.

## Acknowledgements

We would like to acknowledge the financial support of the Texas A&M Energy Institute.

## References

Bemporad, A., M. Morari, V. Dua, and E. Pistikopoulos (2000). The explicit solution of model predictive control via multiparametric quadratic programming. In *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No.00CH36334)*, Volume 2, pp. 872–876 vol.2.

Ferreau, H. J., H. G. Bock, and M. Diehl (2008). An online active set strategy to overcome the limitations of explicit mpc. *International Journal of Robust and Nonlinear Control 18*(8), 816–830.

Goldfarb, D. and A. Idnani (1983). A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical programming 27*(1), 1–33.

Grossmann, I. E. and C. A. Floudas (1987). Active constraint strategy for flexibility analysis in chemical processes. *Computers & Chemical Engineering 11*(6), 675–693.

Gurobi Optimization, LLC (2022). Gurobi Optimizer Reference Manual.

Herceg, M., M. Kvasnica, C. Jones, and M. Morari (2013, July 17–19). Multi-Parametric Toolbox 3.0. In *Proc. of the European Control Conference*, Zürich, Switzerland, pp. 502–510. http://control.ee.ethz.ch/ mpt.

Karush, W. (2014). Minima of functions of several variables with inequalities as side conditions. In *Traces and Emergence of Nonlinear Programming*, pp. 217–245. Springer.

Katz, J. and E. N. Pistikopoulos (2020). A partial multiparametric optimization strategy to improve the computational performance of model predictive control. *Computers & Chemical Engineering 142*, 107057.

Kazantzis, N. and C. Kravaris (2000). Synthesis of state feedback regulators for nonlinear processes. *Chemical Engineering Science 55*(17), 3437–3449.

Kenefake, D., I. Pappas, S. Avraamidou, B. Beykal, H. S. Ganesh, Y. Cao, Y. Wang, J. Otashu, S. Leyland, J. Flores-Cerrillo, and E. N. Pistikopoulos (2022). A smart manufacturing strategy for multiparametric model predictive control in air separation systems. *Journal of Advanced Manufacturing and Processing 4*, e10120.

Kenefake, D. and E. N. Pistikopoulos (2022). Ppopt-multiparametric solver for explicit mpc. In *Computer Aided Chemical Engineering*, Volume 51, pp. 1273–1278. Elsevier.

Kuhn, H. W. and A. W. Tucker (2014). Nonlinear programming. In *Traces and emergence of nonlinear programming*, pp. 247–258. Springer.

Oberdieck, R., N. A. Diangelakis, M. M. Papathanasiou, I. Nascu, and E. N. Pistikopoulos (2016). Pop–parametric optimization toolbox. *Industrial & Engineering Chemistry Research 55*(33), 8979–8991.

Pappas, I., N. A. Diangelakis, and E. N. Pistikopoulos (2020). A strategy for the exact solution of multiparametric/explicit quadratically constrained nmpc problems. *IFAC-PapersOnLine 53*(2), 11380–11385. 21st IFAC World Congress.

Pappas, I., D. Kenefake, B. Burnak, S. Avraamidou, H. S. Ganesh, J. Katz, N. A. Diangelakis, and E. N. Pistikopoulos (2021). Multiparametric programming in process systems engineering: Recent developments and path forward. *Frontiers in Chemical Engineering 2*, 620168.

Pistikopoulos, E. N., N. A. Diangelakis, and R. Oberdieck (2021). *Multi-parametric optimization and Control*. Wiley.