Computers and Chemical Engineering 142 (2020) 107057

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

A partial multiparametric optimization strategy to improve the computational performance of model predictive control☆

Justin Katz[a,b], Efstratios N. Pistikopoulos[a,b,*]

[a] Artie McFerrin Department of Chemical Engineering, Texas A&M University, College Station, TX 77843, USA
[b] Texas A&M Energy Institute, Texas A&M University, College Station, TX 77843, USA

ARTICLE INFO

Article history:
Received 9 March 2020
Revised 14 June 2020
Accepted 14 August 2020
Available online 16 August 2020

Keywords:
Multiparametric programming
Hot start strategies
Model predictive control
Hit and run sampling

ABSTRACT

Determining the optimal manipulated action for large scale model predictive control formulations requires significant computational overhead. It has been demonstrated that the offline, explicit solution provided by multiparametric programming has the capacity to greatly improve the online computational performance of MPC strategies. For large scale problems, developing and deploying the full multiparametric solution remains an open challenge. In this work, a partial multiparametric solution is utilized to improve the initialization procedure for a hot start strategy. The hot start strategy provides an improved technique for determining the optimal solution of large scale MPC formulations, and the partial multiparametric solution ensures the initialization is suitable under varying conditions. The efficacy of the proposed strategy is verified on randomly generated large scale MPC problems.

1. Introduction

The development of control strategies for industrial applications is a relevant and heavily researched field in the chemical engineering discipline, among others (Schäfer et al., 2019; Schultz et al., 2020; Bindlish, 2018; Dua et al., 2008). Model-based techniques, especially model predictive control (MPC), allow for tighter operation and increased profit for more complex processes. In comparison to other control techniques such as proportional-integral-derivative (PID) controllers, MPC controllers allow for a natural representation of systems with multiple-inputs/multiple-outputs and hard constraints (Saletović, 2014). However, a criterion for the widespread adoption of MPC relies on the ability to determine the optimal solution within the time requirements of the process under consideration.

A necessary challenge that must be overcome for the widespread adoption of MPC is the development of the optimal solution in real-time as the problem size grows. Examples of large MPC formulations include processes with fast time scales (Chen et al., 2019; Xi et al., 2013), robust control design (Ning and You, 2019), distributed control architectures (Maxeiner and Engell, 2020), and processes that require surrogate models with correspondingly expanded prediction horizons (Narasingam and Kwon, 2018; 2017). The development of algorithmic strategies to solve these problems in an online setting gatekeep the selection of MPC in modern control design.

One technique to translate solving a large optimization problem into an offline cost is multiparametric programming. Multiparametric programming transforms an implicit optimization formulation involving bounded uncertain parameters into an offline, explicit solution, such that the optimization variables are affine functions of these uncertain parameters. The benefit of the explicit solution has been shown in both online and offline applications. It was demonstrated by Bemporad et al. (2002) that the multiparametric solution has the capacity to improve the online computational performance of MPC by transforming the implicit MPC formulation to an explicit, offline solution. The challenging problem of integrating hierarchical decisions is another avenue involving large scale optimization formulations (Li and Swartz, 2019; Albalawi et al., 2018; Chu and You, 2015). In these formulations, the explicit solution offered by multiparametric programming has demonstrated its applicability and effectiveness (Burnak et al., 2019; Charitopoulos et al., 2019; Burnak et al., 2018; Diangelakis et al., 2017).

Because of the success of multiparametric programming, significant research effort has been mounted to advance relevant theory and develop novel algorithms to tackle larger multiparametric

☆ For the occasion of Professor Sebastian Engell's 65th birthday.

* Corresponding author at: Texas A&M Energy Institute, Texas A&M University, College Station, TX 77843, USA.
E-mail address: stratos@tamu.edu (E.N. Pistikopoulos).

https://doi.org/10.1016/j.compchemeng.2020.107057
0098-1354/

model predictive control problems (Gupta et al., 2011; Wittmann-Hohlbein and Pistikopoulos, 2013; Tøndel et al., 2003; Spjøtvold et al., 2006; Sakizlis et al., 2005; Ahmadi-Moshkenani et al., 2018; Oberdieck et al., 2016). However, there are three inherent issues associated with multiparametric programming, (i) the point location problem, (ii) the memory requirement to store the full multiparametric solution, and (iii) algorithms to obtain the full parametric solution as the problem size increases. Promising research is being conducted in these aforementioned topics (Gupta et al., 2011; Ahmadi-Moshkenani et al., 2018; Xiu and Zhang, 2018; Kvasnica et al., 2015; Oberdieck et al., 2017), but as the problem size grows, these pitfalls will persist.

With increasing problem sizes, developing and utilizing the full multiparametric solution for online applications may not be the correct research direction to move in. Instead, an approach that utilizes either fundamental multiparametric programming theory, or a partial multiparametric solution, to improve online strategies may prove to be more palatable.

Given the known pitfalls of multiparametric programming, many researchers have developed strategies that maintain the benefit of the explicit solution, while avoiding the known challenges. Ziogou *et al.* developed a two step strategy that involves (i) the solution of a multiparametric subproblem that provides 'tight' bounds for (ii) for the original control formulation (Ziogou et al., 2013). Yang and Biegler (2013) utilized sensitivity information, a theoretical cornerstone of multiparametric programming, as a means to improve the computational performance of nonlinear MPC. For continuous time optimal control problems, Hartwich et al. (2011) developed a strategy to improve dynamic optimization with sensitivity information of the given control formulation.

It has been shown that active set strategies utilizing concepts from multiparametric programming can significantly improve the computational performance of determining the optimal control actions for MPC applications (Ferreau et al., 2008; Pannocchia et al., 2011). The work by Ferreau et al. (2008) demonstrated significant online computational savings is achievable by utilizing theoretical aspects of multiparametric programming and not the multiparametric solution. Without adaptation, this hot start strategy presented has the potential pitfall of poor initialization resulting in compromised online performance.

In this work an improved hot start strategy is presented based on the work of Ferreau et al. (2008), and following the work of Katz (2020). The key contribution is the development of a partial mutliparametric solution to strengthen the initialization procedure. The partial multiparametric solution is determined through random walks with the aim of identifying *relevant* critical regions. A computational study is performed involving numerically generated state space models and their corresponding MPC formulations to validate the proposed strategy.

2. Motivation

Developing the model predictive control formulation for a large scale problem is readily implementable with little computational cost. However, solving the resulting problem to optimality is challenging under time considerations. The following motivating example is presented to demonstrate these challenges.

2.1. Large scale MPC

The MPC formulation is based on Eq. (1).

$$\begin{aligned} \min_{u_0,\dots,u_{N-1}} \quad & \vec{x}_N^T P \vec{x}_N + \sum_{i=1}^{N-1} \vec{x}_i^T Q_i \vec{x}_i + \sum_{i=0}^{N-1} \vec{u}_i^T R_i \vec{u}_i \\ \text{s.t.} \quad & \vec{x}_{t+1} = A_d \vec{x}_t + B_d \vec{u}_t, \quad \vec{x}_0 = \vec{x}(0) \\ & \underline{x} \le \vec{x}_t \le \bar{x} \\ & \underline{u} \le \vec{u}_t \le \bar{u} \end{aligned} \tag{1}$$

where $\vec{x}_t$ is the state of system at time t, $\vec{u}_t$ is the manipulated action at time t, $Q_i \succ 0$ and $R_i \succ 0$ are weight matrices, P is the terminal weight matrix and is the solution to the discrete time algebraic Ricatti equation, A_d and B_d define the evolution of the state space in discrete time (e.g. the state space matrices after performing system identification), $\bar{x}$ and $\underline{x}$ are the upper and lower bounds for the states of the system respectively, $\bar{u}$ and $\underline{u}$ are the upper and lower bounds for the manipulated actions of the system respectively, and N is the control and output horizon.

The details of the MPC are as follows. The output and control horizon (i.e. N) are 45, the number of states are 50, and the number of manipulated actions are 40. After transforming the MPC formulation to a quadratic program, the QP has 1800 variables and 8100 constraints.

It is important to point out that determining the offline, explicit solution using multiparametric programming is possible for this problem. However, it is not practical because of the explosion in critical regions defining the full multiparametric solution. The maximum possible number of critical regions associated with a multiparametric programming problem is exponentially bounded. In other words, for a small increase in problem size, the possible number of critical regions grows exponentially. This exponential growth associated with the maximum possible number of critical regions is the reason for the potential explosion in critical regions for large problem sizes (Bemporad et al., 2002). While the true number of critical regions may be far fewer than this theoretical upper bound, the number of critical regions still grows with problem size, and has the capacity to approach this theoretical limit.

2.2. Determining the optimal solution

The computational time to determine the optimal solution of the QP was performed on a computer running 4 cores and an Intel i7-4770 CPU at 3.40 GHz with 16 GB of RAM. All calculations were performed using the MATLAB environment.

Using CPLEX, the time to determine the optimal solution required approximately 45 seconds on average for an arbitrary initial state. Therefore, under time constraints, it is expensive to directly utilize the commercial solver CPLEX because of the large computational overhead.

To further emphasize how large a problem of this size is, assume an active set strategy were employed, whereby an active set combination was selected and the associated optimality conditions were checked, and based on these results either (i) changed the active set combination in an effort to satisfy the optimality criteria or (ii) terminated because the optimality conditions are satisfied. This problem maintains a search space of active set combinations which is large, $\binom{8100}{1800}$, and therefore can make such an active set approach to be impractical. If the active set strategy utilizes a warm start procedures, the computational overhead may still 'blow up' if a poor initialization is performed because of the large active set search space. One reason for the increased computational overhead is because a large matrix must be inverted. This large matrix is defined by Eq. (2) and is based on the first order optimality conditions.

$$K = \begin{bmatrix} Q & \hat{A}^T \\ \hat{A} & \mathbf{0} \end{bmatrix} \tag{2}$$

where $\hat{A}$ is the active constraints of the associated quadratic program. The time required to invert this matrix for this motivating problem is approximately 0.1 seconds. Therefore if hundreds of candidate active set combinations are tested, this strategy would perform worse than CPLEX.

3. Preliminaries

For completeness, multiparametric model predictive control (Bemporad et al., 2002), the hot start strategy (Ferreau et al., 2008), and random walks are briefly explained. These topics are the foundation for the novel strategy presented in this work.

3.1. Multiparametric model predictive control

The MPC formulation given by Eq. (1) is exactly reformulated to its multiparametric model predictive control (mpMPC) counterpart seen by Eq. (3). This reformulation is provided to be consistent with the multiparametric literature, and for the reader's benefit.

$$\begin{aligned} \min_{x} \quad & (Q_{mp}x + H_t\theta + c)^T x \\ \text{s.t.} \quad & Ax \le b + F\theta \\ & CR_A\theta \le CR_b \end{aligned} \tag{3}$$

where x is the vector of optimization variables (i.e. manipulated variables), θ is the vector of uncertain parameters (i.e. initial states of the system), and the objective function matrices Q_{mp}, H_t, and c are defined by cost matrices of the original MPC formulation and implementing variable aggregation on the future states of the process (Jerez et al., 2011). The constraint matrices A, b, and F are defined by the propagation through time of the state space system, and the matrices CR_A and CR_b define the bounds of the initial states of the system.

The multiparametric solution to Eq. (3) provides the offline, explicit solution that relates the optimal manipulated variables to initial states of the system. The explicit solution is a set of critical regions that occupy a subset of the parameter space, and are uniquely identified by an active set combination. Within each critical region is an associated affine relationship between the optimization variables and the uncertain parameters. Eq. (4) defines the multiparametric solution.

$$x^*(\theta) = G_i\theta^* + e_i, \;\; \theta^* \in CR_i \tag{4}$$

where θ^* is the parameter realization (i.e. the measured initial state of the system), x^* is the optimal solution at θ^*, CR_i is the critical region where θ^* resides, and the matrix G_i and vector e_i define the control law for the *ith* critical region. Each critical region is a closed and bounded convex polytope in the form of $P_A x \le P_b$ (Oberdieck et al., 2017).

Each critical region is associated with a unique active set combination. However the facets of the critical region are defined by the (i) nonredundant inactive constraints and/or (ii) the lagrange multipliers associated with the active constraints.

During online implementation, when a parameter realization is made, (i.e. the states of the plant are measured) the optimal solution is identified by finding (i) which critical region contains the parameter realization and (ii) applying the associated control law defined by the associated critical region.

3.1.1. Relevant critical region

In multiparametric programming, a critical region is uniquely defined by an active set combination. The active set combination defines the explicit relationship between optimization variables and uncertain parameters. However, the facets of the critical region are defined by (i) the inactive constraints and (ii) the restriction that the lagrange multipliers must be positive for the associated active constraints. In this work, the concept of a *relevant* critical region is defined to be a critical region that occupies more volume compared to other critical regions. Therefore, larger critical regions in a multiparametric solution are considered relevant. Thus, the larger a critical region is the more relevant it is considered. In other words, if all critical regions were ranked by their respected volumes, where the critical region occupying the most volume would be ranked first, critical regions closer to the first rank are considered large.

Work presented in Katz (2020) demonstrates the importance of these relevant critical regions for large multiparametric problem sizes. The work presented large, randomly generated multiparametric problems and showed a significant portion of the total volume of the feasible uncertain parameter space is defined by only a select few critical regions.

3.2. Hot start strategy

The hot start strategy utilizes the previous parameter realization, θ_{-1}, the corresponding critical region such that $\theta_{-1} \in CR_i$, and the current parameter realization θ_i, to identify the critical region such that $\theta_i \in CR_j$ (Ferreau et al., 2008). Once the critical region (or active set combination) is determined such that $\theta_i \in CR_j$, then the associated control law is determined $x(\theta) = G_j\theta + e_j$ to identify the optimal solution. The procedure is summarized as follows.

- Initialization-Given an initial measurement, θ^*, the optimal solution at this point requires an initial active set combination. If the mpQP is defined from an MPC formulation that contains the origin,[1] then the critical region associated with no active constraints exists (Bemporad and Filippi, 2003). The measurement θ^* is checked to exist in the unconstrained critical region, otherwise a point is selected in the unconstrained critical region, where the algorithm proceeds to the update step. Another strategy to determine an initial active set combination is to solve a deterministic optimization problem at θ^*. The deterministic optimization problem is a result of fixing θ to θ^*, therefore the multiparametric quadratic programming problem is reduced to a standard quadratic programming problem. The solution to the deterministic optimization problem provides the active set combination that is used to construct the initial critical region, CR_0, and control law, i.e. the affine function relating the optimization variables to the uncertain parameters, $x(\theta) = G\theta + e$. The determined control law is utilized until a new parameter realization, θ_i^*, does not belong to the initial critical region, $\theta_i^* \notin CR_0$. If $\theta_i^* \notin CR_0$, proceed to the update step.
- Update-The update corresponds to determining the new critical region, CR_i, and associated control law where the current parameter realization exists, $\theta_i^* \in CR_i$. First, a direction vector, $\vec{d}$, that points from the previous parameter realization, $\theta^* \in CR_0$, to the new parameter realization, θ_i^*, is determined. Then, given the current critical region, CR_0, the previous parameter realization, θ^*, and the direction vector, $\vec{d}$, the intersection of the facet of CR_0 and the line segment joining θ^* and θ_i^* is identified. The active set corresponding to the critical region that is adjacent to CR_0 along the identified facet is then determined. This procedure is run iteratively until the critical region identified is found such that $\theta_i^* \in CR_i$. Performing the update strategy in this way allows for a fast computational online performance, as demonstrated in Ferreau et al. (2008).

Fig. 1 provides a visualization of the proposed algorithm in the parameter space. The white arrows indicate the direction vector, the numbers indicate the parameter realizations and the order in which they were revealed, and the white 'x' indicates where the direction vector passed through a critical region. The parameter space is defined in two dimensions by $x_{0,1}$ and $x_{0,2}$.

[1] An MPC formulation is said to contain the origin if the initial state vector, x_0, leads to a feasible and optimal solution.

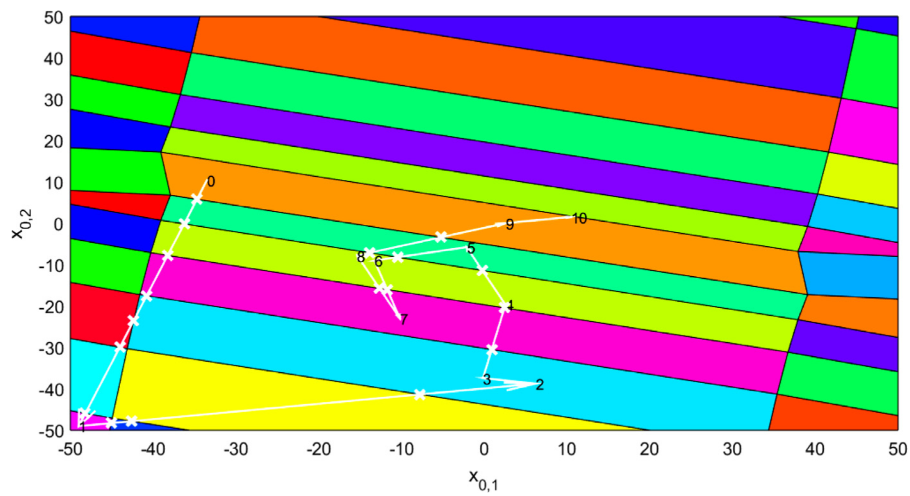

Fig. 1. Visualization of the hot start strategy on the associated multiparametric solution, adapted from Katz (2020).

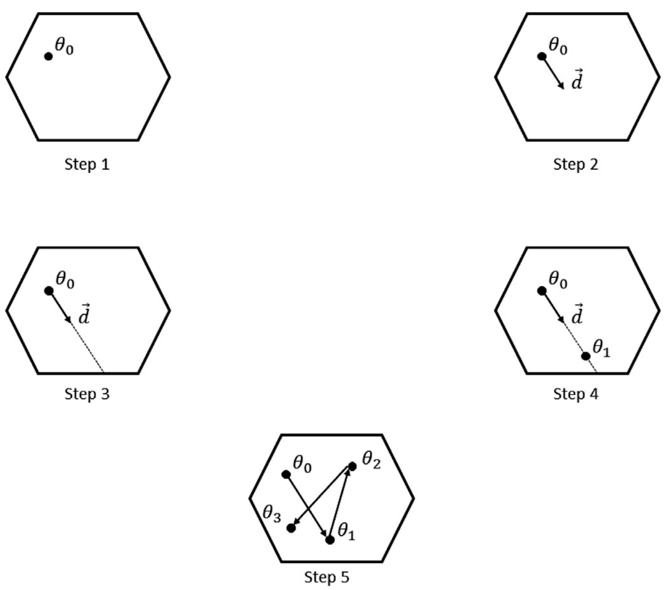

Fig. 2. Adapted from Katz (2020), Step 1) Start with an interior point. Step 2) Randomly choose a direction. Step 3) The distance from the point to the edge of the space is determined. Step 4) Randomly select a random point between the point and the distance determined along the chosen direction, this is the updated point. Step 5) Repeat until termination.

3.3. Random walks

Random walks provide an efficient strategy (i.e. in polynomial time) for a random distribution of samples to be produced that exist in the interior of a polytope defined by its halfspace representation (i.e. $Ax \leq b$). The computational complexity for a random walk to produce a distribution of random samples scales with the dimensionality and the number of facets defining the polytope and is termed the *mixing* time. Many random walks exist in the literature with varying mixing times (Chen et al., 2017), however the hit and run sampling algorithm is a simple and effective strategy (Smith, 1996). A schematic representation of hit and run sampling is presented in Fig. 2 (Yao and Kane, 2017).

4. Methodology

As demonstrated in Section 2, identifying the optimal solution using a commercial solver such as CPLEX for large scale MPC formulations is computationally demanding. In addition, an active set strategy such as the hot start strategy is computationally demanding if a poor initialization is made during real time implementation. To improve the hot start strategy, a two phase approach is performed. The first phase is an offline phase involving (i) hit and run sampling on the feasible domain of the parameter space for the multiparametric model predictive controller, (ii) defining a partial multiparametric solution based on the sampled points, and (iii) identifying a feasible point within each of the defined critical regions. The second phase of the procedure is during online implementation, where an improved initialization procedure is performed. This initialization procedure provides a strategy to minimize the possibility of a poor initialization of the hot start strategy.

4.1. Phase I

Phase I is performed once and offline. During this phase, a partial multiparametric solution is developed using hit and run sampling. Note that it is possible to develop the full multiparametric solution, but for large problem sizes such as the ones considered in this work, even after 30 minutes the full solution may not be identified (Katz, 2020).

To begin, the model predictive control formulation is recast in its multiparametric counterpart, Eq. (3). This recasting is exact and the associated manipulated actions of the MPC formulation are defined by the vector x, and the associated state measurements are defined by the uncertain parameter vector θ. The constraints in Eq. (3) define a high dimensional polytope in the optimization and uncertain parameter space (i.e. $\mathbb{R}^{n_x+n_\theta}$). Hit and run sampling is performed on this polytope to obtain a random distribution of points. From this distribution of points, a random subset is chosen to formulate the partial multiparametric solution. Note, it is possible that many of the chosen points belong to the same critical region because of the random sampling. However, this is not an issue, and is actually preferred because it provides a good indication that the developed critical region is volumetrically significant. By increasing the number of randomly selected points, it is possible to increase the number of critical regions defining the partial multiparametric solution. However, as more critical regions are stored, the inherent issues of a large scale multiparametric solution become apparent. Therefore a balance is needed between selecting all sample points and selecting a single sample point. In addition, a feasible point is determined for each critical region (i.e. the Chebyshev center) to be used during Phase II.

4.1.1. Hit and run sampling

The hit and run sampling strategy is implemented on the convex polytope defined by the feasible region of the multiparametric programming problem, Eq. (5).

$$\begin{bmatrix} A & -F \\ \mathbf{0} & CR_A \end{bmatrix} \begin{bmatrix} x \\ \theta \end{bmatrix} \leq \begin{bmatrix} b \\ CR_b \end{bmatrix} \tag{5}$$

where the polytope is defined in the optimization space x, and the uncertain parameter space θ. To perform the hit and run sampling, first a feasible point is chosen $p_0 = [x_0^T, \theta_0^T]^T$. Then, a random direction is chosen, $\vec{d}$. The maximum distance from the initial point p_0 to the boundary of the polytope is determined. In other words, find the smallest positive t satisfying $[A, -F](p_0 + \vec{d}t) = b$ and $[\mathbf{0}, CR_A](p_0 + \vec{d}t) = CR_b$. The next point is sampled from the polytope by selecting a random number between (0,t] and moving in the direction $\vec{d}$ by this amount. This procedure is repeated until enough sample points have been collected. The amount of sample points needed depends on the dimensionality of the polytope, but scales favorably (i.e. polynomially).

The presented hit and run sampling procedure is the standard approach for sampling from a convex polytope. However, because the objective is to determine a random sampling of points in the parameter space, it is not necessary to consider the full dimensional space of the polytope. By performing the random walk only in the parameter space, the offline computational time is reduced. To perform the hit and run sampling in the parameter space, a slight adjustment is needed for the original strategy. The major components of the strategy remain the same except the step of determining the largest distance that can be traversed while remaining within the feasible space. The reason the largest distance cannot be readily determined is because the feasible parameter space is explicitly defined by the constraint set $CR_A \leq CR_b$ but implicitly defined by the constraint set $Ax \leq b + F\theta$. Therefore, to identify the largest distance that can be traversed (i.e. t), the following optimization problem is solved, Eq. (6), where the initial point is now defined as θ_0.

$$\begin{aligned} \max_t \quad & t \\ \text{s.t.} \quad & Ax \leq b + F(\theta_0 + \vec{d}t) \\ & CR_A(\theta_0 + \vec{d}t) \leq CR_b \\ & t > 0 \end{aligned} \tag{6}$$

With this adjustment to the hit and run strategy, the main steps become as follows. Identify an initial point $p_0 = \theta_0$ to initialize the algorithm. Then randomly select a direction $\vec{d}$. Determine the maximum distance, t, p_0 can move along the direction $\vec{d}$ while remaining feasible in the parameter space. Select a random point between (0, t], and move from p_0 along the direction vector $\vec{d}$ by this amount. Repeat until enough sample points have been collected (i.e. the mixing time).

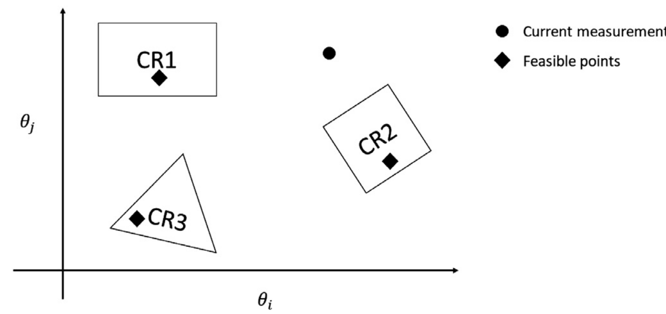

Fig. 3. The steps of Phase II using a partial solution of three critical regions, Step 1) Identify the distance from the current measurement to all feasible points. Step 2) The feasible point that is closest to the current measurement defines the starting critical region to use. Step 3) From the starting critical region, proceed to implement the hot start strategy using the determined critical region as the starting point.

4.2. Phase II

Phase II is performed during online implementation. During this phase, the developed critical regions and associated feasible points are used in real time to initialize the hot start procedure.

Given the parameter realization (i.e. a measurement is performed) the initialization of the hot start procedure is the first step. The developed critical regions during Phase I are utilized for this initialization step. Ideally, the closest critical region to the current parameter realization would be identified (i.e. solving the nearest polytope problem). An exact calculation to determine the closest critical region during online implementation may be too computationally demanding. Therefore, the feasible points associated with each critical region are used to represent their associated critical regions. The distance from each feasible point to the parameter realization is calculated by taking the L_2 norm. The critical region associated with point that is the shortest distance to the parameter realization is used as the initial critical region to be used for the hot start strategy discussed in Section 3.2. The pseudocode for the determination of which critical region to initialize with is provided in Algorithm 1. A visual representation of Phase II is presented in Fig. 3.

It is important to note that the selection of feasible points belonging to each critical region determined during Phase I is critical. Furthermore, different strategies to determine the feasible points belonging to each critical region will lead to a change in computational performance during Phase II.

5. Results

The efficacy of the proposed approach is validated using of randomly generated MPC problems, and is compared against the hot start strategy presented in Section 3.2. The generated MPC problems are based on Eq. (1). The output and control horizon is 45 for all problem sizes, and the underlying state space model maintains 50 states for all problems sizes (i.e. uncertain parameters). A single disturbance is included in the state space model that im-

Require: θ ▷ Parameter realization
Require: CR_{List} ▷ Volumetrically significant critical regions
Require: P ▷ Feasible point for each $CR \in CR_{List}$
1: **for** $i = 1 \rightarrow |CR_{List}|$ **do**
2: $D(i) = \text{GETDISTANCE}(\theta, P(i))$
3: **end for**
4: $index = argmin(D)$
5: $CR = CR_{List}(index)$

Algorithm 1. Determining critical region (Katz, 2020).

Table 1
Comparison of a standard hot-start strategy against the proposed approach, adapted from Katz (2020).

Optimization formulation (variables, constraints)	Manipulation actions	Percent improvement
900, 6300	20	17%
1350, 7200	30	21%
1800, 8100	40	39%

pacts all of the associated states. Note, the addition of the disturbance has no impact on the presented methodology because the uncertain parameter vector is adjusted to go from $\theta = x_0$ to $\theta = [x_0, d]$. Table 1 provides a summary of the problem sizes, and results. Note, it is possible to develop the full multiparametric solution for these problems. However, the offline computational cost would be significant compared to the proposed strategy of determining a partial solution and associated feasible points.

The computational results are developed by starting from an arbitrary state vector, x_0, and applying the optimal solution determined from the MPC formulation to the state space model. This procedure is performed 5 times and the average time to determine the optimal solution for each strategy is determined. This closed loop simulation is rerun for 100 iterations. The average of the 100 trials are used as the basis to identify the percent improvement between the proposed improved hot start strategy and the standard hot start strategy. For the largest problem size tested, the proposed hot start strategy demonstrates a 38% improvement in the average time to determine the optimal solution in real time. For this problem size, the average time to determine the solution for the standard hot start strategy was approximately 14 s, and the proposed hot start strategy is approximately 8.7 s.

6. Conclusion

The use of large scale model predictive control formulations for advanced processes requires the solution under time considerations. However, with increasing problem sizes, the computational burden rapidly increases making it challenging to determine the optimal solution of a large scale quadratic program impractical. Recently a hot start strategy was developed with the ability to improve online performance significantly. This hot start strategy relied on multiparametric concepts, but not in identifying a multiparametric solution. In this work, a partial multiparametric solution was used to improve the initialization procedure for the hot start strategy. By utilizing random walk concepts, the partial multiparametric solution was developed offline and incorporated in an online hot start strategy to improve the online computational performance. A computational study incorporating large randomly generated model predictive control formulation was used as the basis for comparison.

The demonstrated benefit of a partial multiparametric solution makes extending this work to nonlinear MPC formulations a high priority. For example, it was demonstrated in Ziogou et al. (2013) that by merely updating the bounds of a nonlinear MPC formulation, a significant computational speed up can be found. Incorporating the proposed strategy into a nonlinear MPC framework has its challenges however. The first hurdle that must be surpassed is managing the nonconvex critical regions that are known to exist in nonlinear MPC formulations (Pappas et al., 2020; Diangelakis et al., 2018), which will play a significant role during the update step of the presented methodology.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Justin Katz: Methodology, Software, Validation, Formal analysis, Writing - original draft. **Efstratios N. Pistikopoulos:** Conceptualization, Supervision, Project administration, Funding acquisition, Writing - review & editing.

Acknowledgments

Financial support from the NSF SusChEM (Grant no. 1705423) and Texas A&M Energy Institute is gratefully acknowledged. Iosif Pappas is also gratefully acknowledged for his helpful comments.

References

Ahmadi-Moshkenani, P., Johansen, T.A., Olaru, S., 2018. Combinatorial approach toward multiparametric quadratic programming based on characterizing adjacent critical regions. IEEE Trans. Autom. Control 63 (10), 3221–3231. doi:10.1109/TAC.2018.2791479.

Albalawi, F., Durand, H., Christofides, P.D., 2018. Process operational safety via model predictive control: recent results and future research directions. Comput. Chem. Eng. 114, 171–190. doi:10.1016/j.compchemeng.2017.10.006.

Bemporad, A., Filippi, C., 2003. Suboptimal explicit receding horizon control via approximate multiparametric quadratic programming. J. Optim. Theory Appl. 117 (1), 9–38. doi:10.1023/A:1023696221899.

Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E.N., 2002. The explicit linear quadratic regulator for constrained systems. Automatica 38, 3–20.

Bindlish, R., 2018. Scheduling, optimization and control of power for industrial cogeneration plants. Comput. Chem. Eng. 114, 221–224. doi:10.1016/j.compchemeng.2017.10.014.

Burnak, B., Diangelakis, N.A., Katz, J., Pistikopoulos, E.N., 2019. Integrated process design, scheduling, and control using multiparametric programming. Comput. Chem. Eng. 125, 164–184. doi:10.1016/j.compchemeng.2019.03.004.

Burnak, B., Katz, J., Diangelakis, N.A., Pistikopoulos, E.N., 2018. Simultaneous process scheduling and control: amultiparametric programming-based approach. Ind. Eng. Chem. Res. 57 (11), 3963–3976. doi:10.1021/acs.iecr.7b04457.

Charitopoulos, V.M., Papageorgiou, L.G., Dua, V., 2019. Closed-loop integration of planning, scheduling and multi-parametric nonlinear control. Comput. Chem. Eng. 122, 172–192. doi:10.1016/j.compchemeng.2018.06.021.

Chen, S. W., Wang, T., Atanasov, N., Kumar, V., Morari, M., 2019. Large scale model predictive control with neural networks and primal active sets1910.10835.

Chen, Y., Dwivedi, R., Wainwright, M. J., Yu, B., 2017. Fast MCMC sampling algorithms on polytopes. arXiv preprint arXiv:1710.08165

Chu, Y., You, F., 2015. Model-based integration of control and operations: overview, challenges, advances, and opportunities. Comput. Chem. Eng. 83, 2–20. doi:10.1016/j.compchemeng.2015.04.011.

Diangelakis, N.A., Burnak, B., Katz, J., 2017. Process design and control optimization: asimultaneous approach by multi-parametric programming. Am. Inst. Chem. Eng. doi:10.1002/aic.15825.

Diangelakis, N.A., Pappas, I.S., Pistikopoulos, E.N., 2018. On multiparametric/explicit NMPC for quadratically constrained problems. IFAC-PapersOnLine 51 (20), 400–405. doi:10.1016/j.ifacol.2018.11.066.

Dua, P., Kouramas, K., Dua, V., Pistikopoulos, E., 2008. MPC on a chip-recent advances on the application of multi-parametric model-based control. Comput. Chem. Eng. 32 (4), 754–765. doi:10.1016/j.compchemeng.2007.03.008.

Ferreau, H.J., Bock, H.G., Diehl, M., 2008. An online active set strategy to overcome the limitations of explicit MPC. Int. J. Robust Nonlinear Control 18 (8), 816–830. doi:10.1002/rnc.1251.

Gupta, A., Bhartiya, S., Nataraj, P., 2011. A novel approach to multiparametric quadratic programming. Automatica 47 (9), 2112–2117. doi:10.1016/j.automatica.2011.06.019.

Hartwich, A., Stockmann, K., Terboven, C., Feuerriegel, S., Marquardt, W., 2011. Parallel sensitivity analysis for efficient large-scale dynamic optimization. Optim. Eng. 12, 489–508. doi:10.1007/s11081-010-9104-4.

Jerez, J., Kerrigan, E., Constantinides, G., 2011. A condensed and sparse QP formulation for predictive control. pp. 5217–5222. 10.1109/CDC.2011.6160293.

Katz, J., 2020. Advancing multiparametric programming for model predictive control. Texas A&M University Ph.D. thesis.

Kvasnica, M., Takács, B., Holaza, J., Cairano, S.D., 2015. On region-free explicit model predictive control. In: 2015 54th IEEE Conference on Decision and Control (CDC), pp. 3669–3674.

Li, H., Swartz, C.L., 2019. Dynamic real-time optimization of distributed MPC systems using rigorous closed-loop prediction. Comput. Chem. Eng. 122, 356–371. doi:10.1016/j.compchemeng.2018.08.028.

Maxeiner, L.S., Engell, S., 2020. An accelerated dual method based on analytical extrapolation for distributed quadratic optimization of large-scale production complexes. Comput. Chem. Eng. 135, 106728. doi:10.1016/j.compchemeng.2020.106728.

Narasingam, A., Kwon, J.S.-I., 2017. Development of local dynamic mode decomposition with control: application to model predictive control of hydraulic fracturing. Comput. Chem. Eng. 106, 501–511. doi:10.1016/j.compchemeng.2017.07.002.

Narasingam, A., Kwon, J.S.-I., 2018. Data-driven identification of interpretable reduced-order models using sparse regression. Comput. Chem. Eng. doi:10.1016/j.compchemeng.2018.08.010.

Ning, C., You, F., 2019. Optimization under uncertainty in the era of big data and deep learning: when machine learning meets mathematical programming. Comput. Chem. Eng. 125, 434–448.

Oberdieck, R., Diangelakis, N.A., Papathanasiou, M.M., Nascu, I., Pistikopoulos, E.N., 2016. POP - parametric optimization toolbox. Ind. Eng. Chem. Res. 55, 8979–8991.

Oberdieck, R., Diangelakis, N.A., Pistikopoulos, E.N., 2017. Explicit model predictive control: a connected-graph approach. Automatica 76, 103–112. doi:10.1016/j.automatica.2016.10.005.

Pannocchia, G., Wright, S.J., Rawlings, J.B., 2011. Partial enumeration MPC: robust stability results and application to an unstable CSTR. J. Process. Control 21 (10), 1459–1466. doi:10.1016/j.jprocont.2011.06.010.

Pappas, I., Diangelakis, N.A., Pistikopoulos, E.N., 2020. A strategy for the exact solution of multiparametric/explicit quadratically constrained NMPC problems. 21st IFAC World Congress 2020. Accepted Manuscript

Sakizlis, V., Perkin, J., Pistikopoulos, E.N., 2005. Explicit solutions to optimal control problems for constrained continuous-time linear systems. IEE Proc.

Saletović, E., 2014. APM (simple MPC) vs. PID - detailed comparison. Int. J. Adv. Comput. Res. 2249–7277 EBSCO Indexed 4 (1), 26–31.

Schultz, E.S., Hannemann-Tamás, R., Mitsos, A., 2020. Polynomial approximation of inequality path constraints in dynamic optimization. Comput. Chem. Eng. 135, 106732. doi:10.1016/j.compchemeng.2020.106732.

Schäfer, P., Caspari, A., Mhamdi, A., Mitsos, A., 2019. Economic nonlinear model predictive control using hybrid mechanistic data-driven models for optimal operation in real-time electricity markets: in-silico application to air separation processes. J. Process Control 84, 171–181. doi:10.1016/j.jprocont.2019.10.008.

Smith, R.L., 1996. The hit-and-run sampler: a globally reaching Markov chain sampler for generating arbitrary multivariate distributions. In: Proceedings of the 28th Conference on Winter Simulation. IEEE Computer Society, pp. 260—264. doi:10.1145/256562.256619.

Spjøtvold, J., Kerrigan, E.C., Jones, C.N., Tøndel, P., Johansen, T.A., 2006. On the facet-to-facet property of solutions to convex parametric quadratic programs. Automatica 42 (12), 2209–2214. doi:10.1016/j.automatica.2006.06.026.

Tøndel, P., Johansen, T.A., Bemporad, A., 2003. An algorithm for multi-parametric quadratic programming and explicit MPC solutions. Automatica 39 (3), 489–497. doi:10.1016/S0005-1098(02)00250-9.

Wittmann-Hohlbein, M., Pistikopoulos, E.N., 2013. On the global solution of multiparametric mixed integer linear programming problems. J. Global Optim. 57 (1), 51–73. doi:10.1007/s10898-012-9895-2.

Xi, Y.-G., Li, D.-W., Lin, S., 2013. Model predictive control - status and challenges. Acta Autom. Sin. 39 (3), 222–236. doi:10.1016/S1874-1029(13)60024-5.

Xiu, X., Zhang, J., 2018. Grid K-D tree approach for point location in polyhedral data sets - application to explicit MPC. Int. J. Control 1–9. doi:10.1080/00207179.2018.1493536.

Yang, X., Biegler, L.T., 2013. Advanced-multi-step nonlinear model predictive control. J. Process Control 23 (8), 1116–1128. doi:10.1016/j.jprocont.2013.06.011.

Yao, A., Kane, D., 2017. Walkr: MCMC sampling from non-negative convex polytopes. J. Open Source Softw. 2. doi:10.21105/joss.00061.

Ziogou, C., Pistikopoulos, E.N., Georgiadis, M.C., Voutetakis, S., Papadopoulou, S., 2013. Empowering the performance of advanced NMPC by multiparametric programming–an application to a PEM fuel cell system. Ind. Eng. Chem. Res. 52 (13), 4863–4873. doi:10.1021/ie303477h.