

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/278033755>

Complexity Reduced Explicit Model Predictive Control by Solving Approximated mp-QP Program

Conference Paper · June 2015

DOI: 10.1109/ASCC.2015.7244434

READS

30

3 authors, including:



Yutao Chen

University of Padova

2 PUBLICATIONS 0 CITATIONS

SEE PROFILE



Shaoyuan Li

Shanghai Jiao Tong University

222 PUBLICATIONS 1,684 CITATIONS

SEE PROFILE

Complexity Reduced Explicit Model Predictive Control by Solving Approximated mp-QP Program

Y Chen, S Li*, N Li

Department of Automation, Shanghai Jiao Tong University, Key Laboratory of System Control and Information Processing, Ministry of Education, Shanghai, 200240, P.R.China
Email: syli@sjtu.edu.cn*

Abstract—In this paper, two methods to reduce the complexity of multi-parametric programming model predictive control are proposed. We show that the standard multi-parametric programming problem can be modified by approximating the quadratic programming constraints. For a certain control sequence, only constraints on the first element is considered, while constraints on future elements are ignored or approximated to a simple saturation function. Both the number of critical regions and the computation time are proven to be reduced. Geometric interpretations is given and complexity analysis is conducted. The result is tested on an illustrating example to show its effectiveness.

I. INTRODUCTION

Constrained Model Predictive Control(MPC) has long been applied in process industries [10], [3]. The main advantage of MPC is its ability to handle multi-variable system with hard constraints [8]. At each sampling time, MPC solves an optimization problem by taking system's future state predictions into consideration. The first control law in the computed control sequence is implemented and the process is repeated at the next sampling time with updated state information. On-line optimization results to an inexplicit control law and a high computation demand. In order to overcome the drawbacks, Bemporad et al. [2] proposed a Multi-parametric Programming(MP) method to deal with MPC problem. The main idea of MP is to consider the state space as a vector parameter of the optimization problem. After the entire state space is explored, the decision variables are explicitly defined by the states, i.e. a piecewise affine function of the states. Thus, the state space is divided into a number of critical regions and the on-line computation includes only evaluation procedure. Many algorithms for MP are proposed [12], [1], extending MPC to a field with fast plants such as motor control, battery management and electrical vehicle control.

Despite its contribution, MP has obvious disadvantages. The number of all combinations of active constraints is at most 2^q , where q is the number of constraints, usually $q = Nn_u$, where n_u, N is the number of inputs and controls respectively. Due to this exponential relationship, the number of active constraints combinations, as well as the number of critical regions grow dramatically as the system dimensions and prediction horizons become large. Optimizations, instead of done on-line, are done off-line, leading to a high off-line computation burden and a complex piecewise affine control law. Attempts to reduce the complexity of MP have been reported in [5], [7], [11]. However, they all require existed piecewise affine function. In other words, these methods are operations on already

calculated solution to reduce its complexity in posterior.

In this paper, we propose two methods for complexity reduction of mp-QP. The idea of approximating QP constraints is proposed by Zheng [13], to reduce the on-line optimization of MPC. We derive corresponding MP solutions to these approximations. In the first method, the number of critical regions can be easily reduced by ignoring some of the unimportant constraints. Hence, the constraint matrix is a subset of the original one. In the second method, the original QP problem is divided into a number of subproblems, each with a single inequality constraint and some equality constraints. These methods are reduced solution of the original mp-QP problem and do not require additional steps for complexity reduction.

II. MULTI-PARAMETRIC PROGRAMMING BASED MPC

A general discrete time linear system is considered for regulating problem:

$$x_{k+1} = Ax_k + Bu_k \quad (1)$$

$$y_k = Cx_k \quad (2)$$

where $x \in \mathbb{R}^{n_x}, u \in \mathbb{R}^{n_u}, y \in \mathbb{R}^{n_y}$ is the state vector, manipulated variable and measured output respectively. At time k , A MPC controller minimizes the following finite horizon cost function subject to polyhedral input constraints

$$\begin{aligned} \min_U J &= x'_{t+N} P x_{t+N} + \sum_{k=0}^{N-1} (x'_{t+k} Q x_{t+k} \\ &+ u'_{t+k} R u_{t+k}) \quad (3) \\ \text{s.t.} \quad &u_{min} \leq u_{t+k} \leq u_{max}, \quad k = 1, \dots, N \end{aligned}$$

where $U = [u_t, u_{t+1}, \dots, u_{t+N-1}]'$, $Q = Q' \geq 0, R = R' \geq 0, P \geq 0$. N is the prediction horizon and P is a terminal penalty matrix used to guarantee the stability of the system and always obtained by solving the discrete Lyapunov equation

$$P = A' P A + Q \quad (4)$$

The optimal solution U^* is obtained by solving Problem (3) and only the first component u_t is implemented into the system. The procedure is repeated at each sampling time by adopting the updated x_t , known as receding or moving horizon strategy.

Substituting (1) into (3), we obtain a compact expression of the optimization problem.

$$\begin{aligned} \min_U J &= \frac{1}{2} U' H U + x'_t F U + \frac{1}{2} x'_t Y x_t \quad (5) \\ \text{s.t.} \quad &G U \leq W + E x_t \end{aligned}$$

where H, F, Y can be easily obtained by matrix operations from A, B, P . Traditionally Problem (5) is solved on-line and will cost a large amount of computations when the system is complex, i.e. n_x, N are large. In addition, the optimal control law is implicit as a result of numerical optimization. An effective way to handle these drawbacks is to solve the above problem by Multi-parametric Quadratic Programming (mp-QP), which solves the QP problem off-line by taking the system state space as a vector parameter. The optimal control law U^* is given as a piecewise linear affine function of the state x_t . Thus, the on-line computation degenerates to an evaluation problem, which definitely saves most of the computations. The mp-QP formulation is easily obtained by transforming (5) and we define this standard mp-QP as the following

Definition 1: Controller #0

$$\begin{aligned} \min_U J &= \frac{1}{2} z' H z & (6) \\ \text{s.t.} \quad G z &\leq W + S x_t \end{aligned}$$

where $z = U + H^{-1} F' x_t$, $S = E + G H^{-1} F'$. z is proven to be an affine function of x by using the first-order KKT conditions for Problem (5)

$$H z + G' \lambda = 0 \quad (7)$$

$$\lambda_i (G_i z - W_i - S_i x) = 0 \quad (8)$$

$$\lambda \geq 0 \quad (9)$$

Finally, the Lagrange multipliers λ and z can be explicitly expressed by x

$$\lambda = -(\tilde{G} H^{-1} \tilde{G}')^{-1} (\tilde{W} + \tilde{S} x) \quad (10)$$

$$z = H^{-1} \tilde{G}' (\tilde{G} H^{-1} \tilde{G}')^{-1} (\tilde{W} + \tilde{S} x) \quad (11)$$

where $\tilde{G}, \tilde{W}, \tilde{S}$ correspond to the set of active constraints. Effective algorithms have already been proposed to solve the mp-QP problem [12], [9]. The state space is partitioned into a number of critical regions, in which a corresponding optimal control law is decided. During on-line operation, once the updated measurement x_t is obtained, the critical region where x_t belongs to is identified. The optimal control law $U(x_t) = z(x_t) - H^{-1} F' x_t$ is immediately obtained and the first component of U is implemented. Therefore, no optimizer but only an evaluator is called on-line. A toolbox MPT Toolbox based on Matlab is friendly to users to develop mp-QP controllers and it is used throughout this paper to obtain complexity reduced solutions[6].

As stated above, the mp-QP algorithm increases the off-line computation for region partition as well as data storage, since the number of critical regions is in exponential relation with the system decision variables and constraints. Several complexity reduction methods are already available to reduce the critical regions. Kvasnica [5] proposed a clipping-based method in which regions of a PWA feedback where the control action is saturated are completely eliminated and replaced by extensions of unsaturated regions. He also gives a separation-based method based in which regions in which the optimal control action is saturated are completely removed [7]. Takacs et al. [11] gives an optimal PWA fitting scheme. For given PWA feedback law, it is approximated by a different PWA function of lower complexity such that the approximation error is minimized. Despite their success on complexity reduction,

prerequisites for existed continuous or discontinuous PWA feedback law is required. Thus, the methods just mentioned are additional steps in posteriori. In the following, we propose two methods for complexity reduction where an mp-QP problem is solved by approximating QP constraints without any extra steps for complexity handling.

III. MPC WITH REDUCED MP-QP

The original QP problem consists of $n_u \times N$ decision variables. The QP approximations manage to account for the constraints on the first control component while ignoring or approximating constraints on others. Two controllers are defined in this way:

Definition 2: Controller #1

$$\begin{aligned} \min_U J &= x'_{t+1} P x_{t+1} + u'_t R u_t + x'_t Q x_t & (12) \\ \text{s.t.} \quad u_{min} &\leq u_t \leq u_{max} \end{aligned}$$

In Controller #1, only the first control law u_t is penalized and only its constraints are included. Thus, Controller #1 softens the cost demands and constraints. The overall number of decision variables have been reduced to n_u . Specifically, the dimension of matrix \hat{H}, \hat{F} is $n_u \times n_u, n_x \times n_u$. Reform (12) as (5), we have

$$H = \begin{bmatrix} \hat{H} & O_{n_u \times (N-1)n_u} \\ O_{(N-1)n_u \times n_u} & O_{(N-1)n_u \times (N-1)n_u} \end{bmatrix} \quad (13)$$

$$F = \begin{bmatrix} \hat{F} & O_{n_x \times (N-1)n_u} \end{bmatrix} \quad (14)$$

where \hat{H} is of dimension $n_u \times n_u$, \hat{F} is of dimension $n_x \times n_u$ and $O_{a \times b}$ is a zero matrix with a rows and b columns. Notice that $\det(H) = 0$, resulting to an infeasible operation on H^{-1} as well as z . Thus, in order to solve this problem by mp-QP, we have to modify the expression of z .

Assuming $z = U + M x_t$.

$$\begin{aligned} \frac{1}{2} z' H z &= \frac{1}{2} (x'_t M' H U + U' H U \\ &\quad + U' H M x_t + x'_t M' M x_t) \end{aligned} \quad (15)$$

Comparing (15) with (5), we have

$$M' H = F \quad (16)$$

Take (13) into account,

$$M = [\hat{H}^{-1} \hat{F} \quad O_{n_u \times (N-1)n_u}]' \quad (17)$$

The matrix S in constraints becomes

$$S = E + G M \quad (18)$$

After the modification, Problem (12) can be solved by existed mp-QP algorithms. As indicated in [13], Controller #1 will possibly lead to an approximation error when $u_{t+k}, k = 1, 2, \dots, N-1$, violates constraints while u_t remains in the feasible region. Although not perfect, Controller #1 gives acceptable performance with nearly invisible difference from the original MPC.

Definition 3: Controller #2

$$\min_U J = \frac{1}{2}U'HU + x_t'FU + \frac{1}{2}x_t'Yx_t \quad (19)$$

s.t. $u_{min} \leq u_t \leq u_{max}$
 $u_{t+k} = sat(Kx_t(t+k)) \quad k = 1, 2, \dots, N-1$

where $K = -H^{-1}F'$ is the stabilized unconstrained feedback gain of $U = Kx_t$ which can be computed off-line, and $sat(\cdot)$ is the saturation function. The purpose of Controller #2 is to take constraints on future inputs into account in an explicit function instead of in an optimization process. For each unconstrained future input $u_{t+k} = Kx_{t+k}$, the function $sat(\cdot)$ gives it three status, which are

- 1) $Kx_{t+k} \geq u_{max}$
- 2) $Kx_{t+k} \leq u_{min}$
- 3) $u_{min} \leq Kx_{t+k} \leq u_{max}$

Totally, 3^{N-1} subproblems are obtained by exploring all the combinations of the status. For each combination, there exists an inequality constraint $u_{min} \leq u_t \leq u_{max}$ and $N-1$ corresponding equality constraints, i.e. $u_{t+k} = u_{max}$ or $u_{t+k} = u_{min}$ or $u_{t+k} = Kx_{t+k}$, $k = 1, 2, \dots, N-1$. Thus, solving Problem (19) by mp-QP resulting to solving 3^{N-1} subproblems iteratively (with single processor) or simultaneously (with multi-core processor). Finally critical regions from the subproblems are combined to form an overall partition of the state space. The partition can guarantee a complete exploration of the state space, since all possible combinations of the control move are considered. Controller #2 is a better approximation of the original QP problem, with nearly the same performance and control laws as the original mp-QP MPC. Both on-line and off-line computations can be reduced, for detailed result see Section IV. A general rule for solving Problem (19) is as follows:

Algorithm 1 mp-Qp algorithm for Controller #2

- 1: Define three status for each future control move u_{t+k} .
 - 2: Enumerate all 3^{N-1} combinations of future control moves.
 - 3: Solve (19) for each combination defined in step 2.
 - 4: Combine 3^{N-1} subsets and obtain a unified state partition.
-

IV. GEOMETRIC INTERPRETATIONS AND COMPLEXITY ANALYSIS

We have argued that the two controllers approximate the original mp-QP problem with reduced complexity. In this section, geometric interpretations are given. Also, a quantitative complexity analysis is derived.

A. Geometric Interpretations

The geometric interpretation of QP was proposed by Goodwin et al. [4]. A typical two-dimension u -space of QP is illustrated in Fig.1. QP finds the point in the parallelogram with least Euclidean distance to the unconstrained optimal point u_k^0 . A feedback gain can be derived in each part of the u -space and the result is the same as mp-QP.

If constraints on only the first control move u_t are considered, as in Controller #1, the u -space will definitely be divided

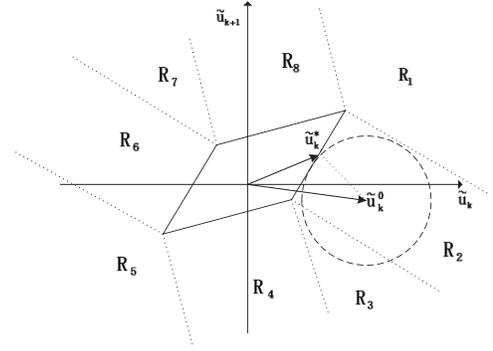


Fig. 1: A general geometric interpretation of QP in \tilde{u} -space ($N = 2$)

into three parts, as shown in Fig. 2. The tangent point \tilde{u}_t^* always locates at the constraint boundary, leading to

$$U^* = u_t^* = sat(Kx_t) \quad (20)$$

To this extent, Controller #1 approximates the original mp-QP

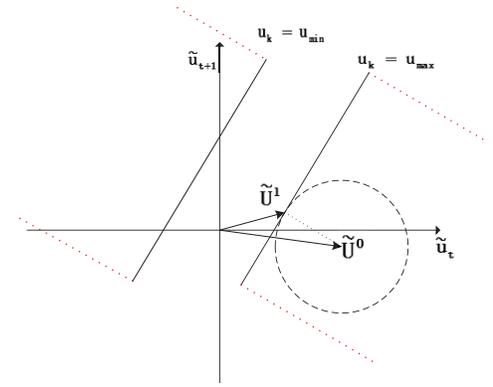


Fig. 2: Partition for Controller 1

problem with least critical regions, because the function $sat(\cdot)$ is the simplest way of handling constraint.

The three controllers Controller #0, #1 and #2 are compared in Fig.(3). $\tilde{U}_0, \tilde{U}_1, \tilde{U}_2$ are transformed control vectors for the three controllers. u_t^0, u_t^1, u_t^2 are the real control variable to be applied respectively. A situation in which the unconstrained u_t obeys the constraint and u_{t+1} violates the constraint is shown. Controller #0 finds the tangent point and for Controller #1, the upper and lower bound of the parallelogram do not exist, resulting to u_t^1 the same as the unconstrained Kx_t . Controller #2 considers the constraint on u_{t+1} by using the $sat(\cdot)$ function. Thus, its control vector \tilde{U}^2 “stops” when reaching the boundary of u_{t+1} . Therefore, $u_{t+1}^2 = u_{t+1}^0 = u_{max}$ but $u_t^2 \neq u_t^0$.

B. Complexity Analysis

For polyhedral boundary, the number of possible combination of active constraints of a QP is at most 2^q , where q is the number of constraints. The way of generating regions can be associated with a search tree with a maximum depth of 2^q .

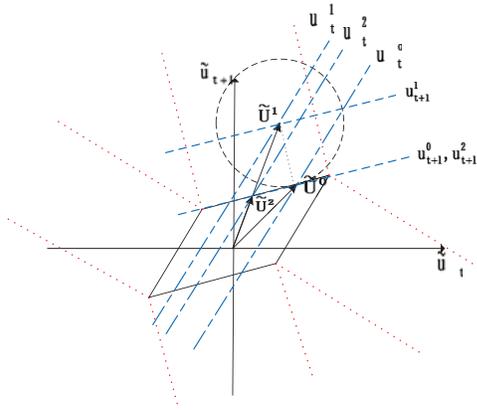


Fig. 3: Comparison of three controllers($N = 2$)

Therefore, the number of critical regions is at most [9]

$$Nr \leq \sum_{k=0}^{2^q-1} k!q^k \quad (21)$$

In this paper, the number of constraints equals to the prediction horizon, i.e. $q = N$, since only input constraint is considered. The number of critical regions for Controller #1 and Controller #2 at the worst situation can be obtained. For Controller #1, according to (21), take $q = N = 1$, we have

$$Nr_1 \leq \sum_{k=0}^1 k! = 3 \quad (22)$$

For Controller 2, according to Algorithm 1, Problem (19) is divided into 3^{N-1} parts. In each part, the number of constraint $q_s = 1$. Thus, the total number of critical regions is

$$\begin{aligned} Nr_2 &\leq 3^{N-1} \sum_{k=0}^{2^{q_s}-1} k! \\ &\leq 3^{N-1} Nr_1 \\ &\leq 3^N \end{aligned} \quad (23)$$

The difference between (23) and (21) becomes significant when N is large. In addition, Algorithm 1 allows parallel computing in each subproblem. The original mp-QP problem is solved by exploring the state space sequentially, started from an initial point at a certain region, defining a critical region at each searching step [12]. Then the rest of the critical regions are defined, where the next step of searching is initiated. Note that at each iteration, the critical region is defined by solving a quadratic problem for optimal control law and a linear program for redundant constraints removal. However, Algorithm 1 defines 3^{N-1} subregions in step 1 immediately without any optimization. The exploration of the subspace can be initiated simultaneously, each solving a reduced mp-QP problem with a single inequality constraint. From (23), the number of critical regions in each subregion is identical to Controller #1. As a result, the number of iteration for optimization is also reduced. Therefore, Algorithm 1 reduced not only the number of critical regions to be explored but also the computation time for iteration, especially under parallel computing.

TABLE I: The Number of Critical Regions(CRs) and the Computation Time(CT). Simulation done by computer with Intel i3 3.0 GHz, 2GB RAM, Matlab R2014a

	Controller	CRs	CT[s](Total)	CT[s](Average)
$N = 2$	#0	9	0.1590	
	#1	3	0.0470	
	#2	9	0.1450	0.0483
$N = 3$	#0	19	0.4310	
	#1	3	0.0570	
	#2	17	0.3880	0.1293
$N = 4$	#0	29	0.5320	
	#1	3	0.0490	
	#2	15	0.2590	0.0863
$N = 5$	#0	31	0.6280	
	#1	3	0.0510	
	#2	17	0.3240	0.1080

V. ILLUSTRATING EXAMPLE

The following linear discrete time system is considered [9]:

$$x_{t+1} = \begin{bmatrix} 0.7326 & -0.0861 \\ 0.1722 & 0.9909 \end{bmatrix} x_t + \begin{bmatrix} 0.0609 \\ 0.0064 \end{bmatrix} u_t \quad (24)$$

$$y_t = [0 \quad 1.4142] x_t \quad (25)$$

The constraints on input are

$$-2 \leq U \leq 2 \quad (26)$$

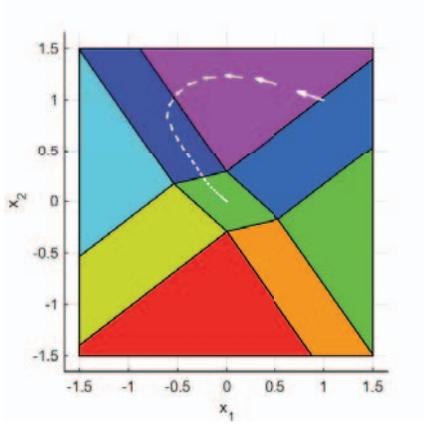
Parameters for MPC controller are chosen as

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, R = 0.01 \quad (27)$$

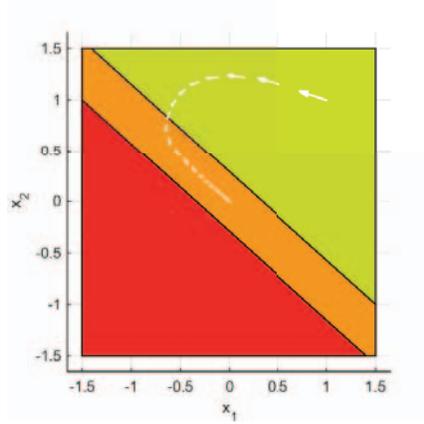
and the terminal penalty P is obtained by (4). Three controllers-Controller #0, #1 and #2 are tested on the system with different prediction horizon N . Fig. 4a to 4c shows the state space partition for the three controllers when $N = 2$. The state space partition of the three controllers for $N = 3, 5$ are shown in Fig. 5 and 6.

In Fig. 4a, 5a and 6a, the optimal partition for $N = 2, 3, 5$ are obtained. These are standard solution of mp-QP problems. Notice that Fig. 4b, 5b and 6b are almost the same. The reason is that for Problem (12), the prediction horizon N has no effect on the cost function as well as the constraints. Therefore, the state space will always be divided into a fixed number of regions. However, the unconstrained region in Fig. 6b is tighter than in Fig. 4b and Fig. 5b, since more free control variables $u_{t+k}, k = 2, 3, 4$ are introduced. In Fig. 4c, 5c and 6c, it can be obviously observed that the state space is divided into a number of parts at first by some group of parallel lines, and then subparts are generated in each part. Every subpart is convex. Table I shows the number of critical regions and the computational time required for the three controllers with prediction horizon $N = 2, 3, 4, 5$.

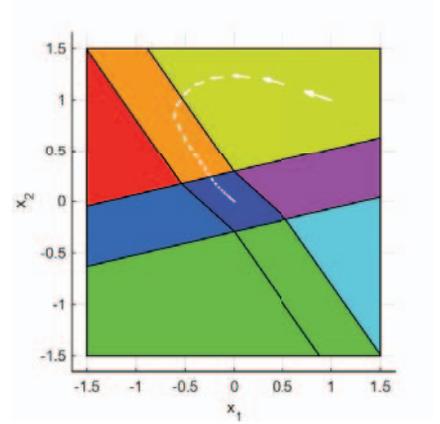
By applying Controller #1 and Controller #2, the number of critical regions are significantly reduced, especially as N grows large. In addition, both the total and average computation time are reduced. Notice that for Controller #0 and Controller #1, the partition is obtained only by iteratively exploring the state space. For Controller #2, it is feasible to conduct parallel



(a) Controller #0

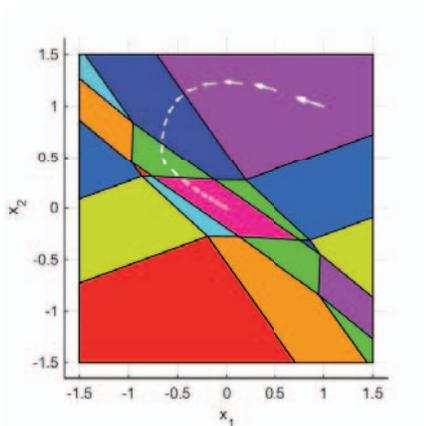


(b) Controller #1

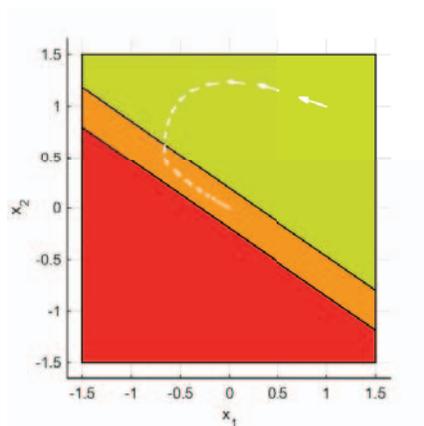


(c) Controller #2

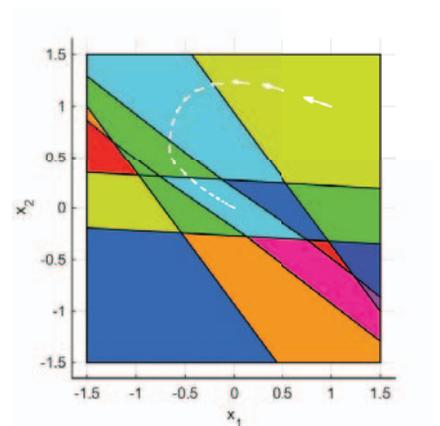
Fig. 4: State Space Partition for $N = 2$



(a) Controller #0

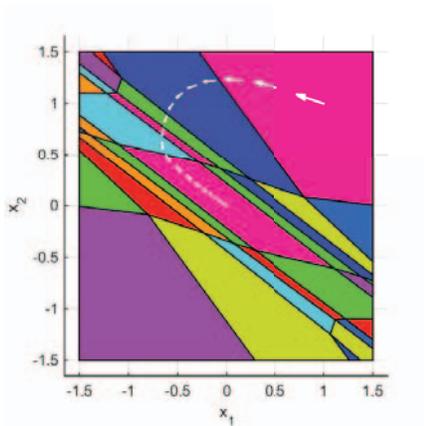


(b) Controller #1

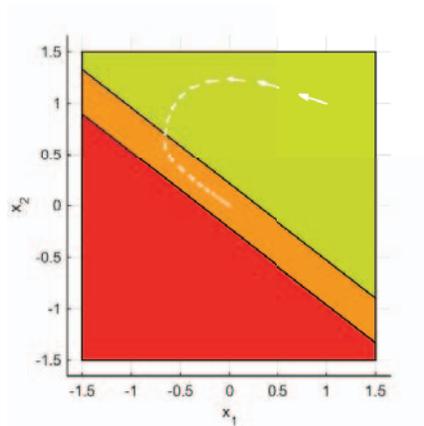


(c) Controller #2

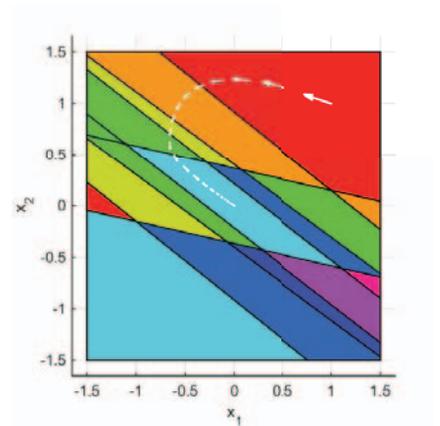
Fig. 5: State Space Partition for $N = 3$



(a) Controller #0



(b) Controller #1



(c) Controller #2

Fig. 6: State Space Partition for $N = 5$

computing. As indicate in Table I, if 3 cores are used, the partition result can be obtained within the average time length.

A comparative simulation for the example are shown in Fig. 7 and 8. Although the state space partition vary a lot for different controllers and different choices of prediction horizon, the system performances as well as the control laws are similar. Specifically, Controller #2 approximates Controller #0 very well, resulting to almost identical system performance. Controller #1, however, maintain a longer time of controller saturation because it is “short seeing”-no predictions and no future constraints. This result is tested by Zheng [13] in on-line optimization MPC and here we successfully transfer it to off-line mp-QP. Therefore, the two methods we proposed for mp-QP are effective for complexity reduction with minimal performance loss. It extends the result of Zheng to off-line computation and enables a faster state space partition.

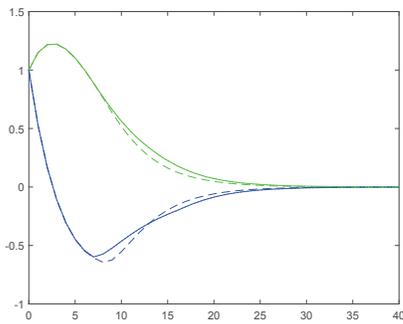


Fig. 7: System States: $x_0 = [1; 1]$. Controller #0(solid line), Controller #1(dashed line), Controller #2(dotted line). ($N = 2$)

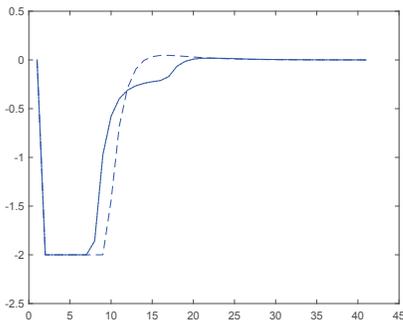


Fig. 8: Optimal Control Law: $x_0 = [1; 1]$. Controller #0(solid line), Controller #1(dashed line), Controller #2(dotted line). ($N = 2$)

VI. CONCLUSION

We have shown that the complexity of a multi-parametric programming model predictive control problem can be reduced by approximating the constraints during optimization. We propose two modified controllers compared with the standard mp-QP Controller #0. Given that only the first element of the control sequence is implemented at each sampling time, constraints on future elements are ignored in Controller #1 and are

approximated to a simple saturation function in Controller #2. The geometric interpretations show the difference of optimal control law among the three controllers. According to the complexity analysis, the number of critical regions is significantly reduced, especially for a large prediction horizon N . Also, Algorithm 1 enables parallel computing, which extends mp-QP problem to be solved using multi-core processors.

Compared with other complexity reduction methods, our methods try to obtain a complexity reduced solution without any extra steps for complexity handling. The simulation result shows that the two modified controllers can effectively reduce the number of critical regions while maintain the system performance.

ACKNOWLEDGMENT

This work was supported by the National Nature Science Foundation of China (61233004, 61221003, 61374109, 61104091, 61304078, 61473184), the National Basic Research Program of China (973 Program-2013CB035500), and partly sponsored by the International Cooperation Program of Shanghai Science and Technology Commission (12230709600), the Higher Education Research Fund for the Doctoral Program of China (20120073130006 & 20110073110018), and the China Postdoctoral Science Foundation (2013M540364).

REFERENCES

- [1] Alessandro Alessio and Alberto Bemporad. A survey on explicit model predictive control. In *Nonlinear model predictive control*, pages 345–369. Springer, 2009.
- [2] Alberto Bemporad, Manfred Morari, Vivek Dua, and Efstratios N Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.
- [3] Carlos E Garcia, David M Prett, and Manfred Morari. Model predictive control: theory and practice survey. *Automatica*, 25(3):335–348, 1989.
- [4] Graham C Goodwin, María M Seron, and José A De Doná. *Constrained control and estimation: an optimisation approach*. Springer, 2006.
- [5] Michal Kvasnica and Miroslav Fikar. Clipping-based complexity reduction in explicit mpc. *Automatic Control, IEEE Transactions on*, 57(7):1878–1883, 2012.
- [6] Michal Kvasnica, Pascal Grieder, Mato Baotić, and Manfred Morari. Multi-parametric toolbox (mpt). In *Hybrid systems: computation and control*, pages 448–462. Springer, 2004.
- [7] Michal Kvasnica, Juraj Hledík, Ivana Rauová, and Miroslav Fikar. Complexity reduction of explicit model predictive control via separation. *Automatica*, 49(6):1776–1781, 2013.
- [8] Jan Marian Maciejowski. *Predictive control: with constraints*. Pearson education, 2002.
- [9] Efstratios Pistikopoulos, Amparo Galindo, Vivek Dua, Eustathios S Kikkinides, Lazaros Papageorgiou, Klaus-Werner Benz, Wolfgang Neumann, Michael Kohler, Wolfgang Fritzsche, Walter Brockmann, et al. Multi-parametric model-based control: Theory and applications, volume 2. 2007.
- [10] S Joe Qin and Thomas A Badgwell. A survey of industrial model predictive control technology. *Control engineering practice*, 11(7):733–764, 2003.
- [11] Bálint Takács, Juraj Holaza, Michal Kvasnica, and Stefano Di Cairano. Nearly-optimal simple explicit mpc regulators with recursive feasibility guarantees. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 7089–7094. IEEE, 2013.
- [12] Petter Tøndel, Tor Arne Johansen, and Alberto Bemporad. An algorithm for multi-parametric quadratic programming and explicit mpc solutions. *Automatica*, 39(3):489–497, 2003.
- [13] Alex Zheng. Reducing on-line computational demands in model predictive control by approximating qp constraints. *Journal of process control*, 9(4):279–290, 1999.