MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

On Region-Free Explicit Model Predictive Control

Kvasnica, M.; Takacs, B.; Holaza, J.; Di Cairano, S.

TR2015-148 December 2015

Abstract

We show that explicit MPC solutions admit a closed-form solution which does not require the storage of critical regions. Therefore significant amount of memory can be saved. In fact, not even the construction of such regions is required. Instead, all possible optimal active sets are first extensively enumerated. Then, for each optimal, only the analytical expressions of primal and dual variables are stored. Optimality of a particular if checked by verifying primal and dual feasibility conditions, which are unique for all candidate sets. We show that the required memory storage can be further reduced by only storing the factors for the dual variables.

IEEE Conference on Decision and Control (CDC)

On Region-Free Explicit Model Predictive Control

Michal Kvasnica, Bálint Takács, Juraj Holaza, and Stefano Di Cairano

***Abstract*— We show that explicit MPC solutions admit a closed-form solution which does not require the storage of critical regions. Therefore significant amount of memory can be saved. In fact, not even the construction of such regions is required. Instead, all possible optimal active sets are first extensively enumerated. Then, for each optimal , only the analytical expressions of primal and dual variables are stored. Optimality of a particular if checked by verifying primal and dual feasibility conditions, which are unique for all candidate sets. We show that the required memory storage can be further reduced by only storing the factors for the dual variables.**

I. Introduction

Explicit model predictive control (MPC) [2] has garnered a significant attention in the community for three reasons. First, it allows to implement MPC in a division-free fashion using only additions and multiplications, hence simplifying certification for mission-critical applications. Second, it provides an exact worst-case implementation analysis thus allowing to fulfil rigorous real-time guarantees by an appropriate choice of the implementation hardware. Third, since explicit MPC synthesizes an explicit representation of the feedback law for all feasible initial conditions, it allows to rigorously analyze the closed-loop system, e.g., with respect to Lyapunov stability or liveness analysis. These three advantages are achieved by solving a given MPC optimization problem using parametric programming, which results in a piecewise affine (PWA) feedback law defined over polyhedral critical regions.

However, the construction of such regions (which happens off-line) and the memory required to store them for on-line implementation are the main limitations of explicit MPC. In the off-line phase, traditional geometric approaches (e.g., [2, 1]) enumerate the regions by performing numerically sensitive geometric operations, which scale badly with the increasing of the dimensionality of the parametric space. Therefore, from a practical point of view, they are limited to systems with a small number of states. Even if the regions could be constructed, they usually occupy an impractically large amount of memory of the control platform. This is due to two facts. First, the number of critical regions grows exponentially with the number of constraints (i.e., with the prediction horizon and the number of inputs and states). Second, each critical region is stored as a set of half-spaces, which are obtained as affine transformation of constraints of the original MPC problem.

To attack the memory issue, in this note we revisit the idea of [4] and show that explicit MPC solutions in fact require neither the construction, nor the storage of critical regions. Instead, only a (partial) factorization of the Karush-Kuhn-Tucker system for all possible s is retained. Optimality of a particular for a given value of the parameter is checked by verifying primal and dual feasibility conditions, which are unique for all s. We refer to this approach as *region-free* explicit MPC. Although this underlying idea of [4] has been known for 5 years, its significance with respect to the reduction of memory storage appears to have slipped under the radar of the community. Most probably because its practical significance wasn't clear until [6] presented a way how to efficiently enumerate optimal active sets without *creating* the critical regions in the first place. Combining the extensive enumeration of [6] with the region-free approach of [4] it becomes possible to construct explicit MPC solutions even for moderately large parametric spaces.

This paper introduces several novel results. First, we show how to address the main limitation of [4], which is the lack of a closed-form representation of region-free explicit MPC. In particular, we show that the explicit MPC feedback law can be obtained as a PWA function which maps state measurements onto optimal control inputs and does not require the storage of critical regions. Thus, we allow the region-free format to be used for rigorous closed-loop analysis. Second, we show that by pre-computing fewer data, we can save about half of the memory required by [4]. Here, the idea is to only store the factors for the dual variables and compute the primal ones on-the-fly. The implementation is still division-free, but the price to be paid is an increased on-line computation. Finally, we also show how to convert conventional region-based explicit solutions to the region-free format and vice versa.

II. Explicit MPC

We consider MPC setups represented by a constrained finite-time optimal control (CFTOC) problem of the form

$$\min_{u_0,\ldots,u_{N-1}} \ \ell_{\mathrm{N}}(x_N) + \sum_{k=0}^{N-1} \ell(x_k, u_k) \tag{1a}$$

$$\text{s.t. } x_{k+1} = Ax_k + Bu_k, \ k = 0, \ldots, N-1, \tag{1b}$$

$$(x_k, u_k) \in \mathcal{Z}, \ k = 0, \ldots, N-1, \tag{1c}$$

$$x_N \in \mathcal{F}, \tag{1d}$$

where $x_k \in \mathbb{R}^{n_{\mathrm{x}}}$ and $u_k \in \mathbb{R}^{n_{\mathrm{u}}}$ represent, respectively, predictions of the states and inputs at the k-th step of the prediction window, N is the prediction horizon, ℓ_{N} is the

M. Kvasnica, B. Takács, and J. Holaza are with the Slovak University of Technology in Bratislava, Slovakia, {michal.kvasnica, balint.takacs, juraj.holaza}@stuba.sk. S. Di Cairano is with Mitsubishi Electric Research Laboratories, Cambridge, MA, USA, dicairano@ieee.org.

terminal penalty, $\ell(\cdot,\cdot)$ is the stage cost, and $\mathcal{Z}$, $\mathcal{F}$ are full-dimensional polyhedral sets of appropriate dimensions. CFTOC problems (1) are general enough to capture various setups, e.g., regulation problems, trajectory tracking, slew-rate penalties and constraints, etc. In this paper, we assume that the terminal and stage cost functions are strictly convex quadratic functions, e.g., $\ell_{\mathcal{N}} = x_N^T Q_{\text{N}} x_N$ and $\ell(x_k, u_k) = x_k^T Q_{\text{x}} x_k + u_k^T Q_{\text{u}} u_k$ with $Q_{\text{N}} \succ 0$, $Q_{\text{x}} \succ 0$, $Q_{\text{u}} \succ 0$ for the case of regulation. The objective is to solve (4) for any feasible initial condition x_0, i.e., to determine $U^\star = [u_0^{\star T}, \dots, u_{N-1}^{\star T}]^T$ as a function of the initial conditions. In the case of regulation problems, the initial condition is just x_0. In tracking problems with trajectory preview, the initial condition also includes the prediction of the reference to be tracked. In slew rate setups, it also embodies knowledge of the control action from the previous time step. Therefore, we denote the vector of initial conditions of (1) by θ.

Using straightforward algebraic manipulations, problem (1) can be converted into a parametric quadratic programs (pQP) of the form

$$\min_{z} \ 1/2\, U^T H U + \theta^T F U \tag{2a}$$

$$\text{s.t. } GU \leq w + E\theta, \tag{2b}$$

where $z \in \mathbb{R}^m$ is the vector of optimization variables, $\theta \in \mathbb{R}^n$ is the vector of parameters, and $H \in \mathbb{R}^{m \times m}$, $F \in \mathbb{R}^{n \times m}$, $G \in \mathbb{R}^{p \times m}$, $w \in \mathbb{R}^p$, $E \in \mathbb{R}^{p \times n}$ are problem data. Under the assumption that $\ell_N(\cdot)$ and $\ell(\cdot,\cdot)$ in (1a) are strictly convex quadratic functions, H in (4a) is positive definite.

As shown, e.g., in [3], the parametric solution to (2), i.e., the map from the space of parameters to the space of optimal decision variables, is a piecewise affine (PWA) function of the form $U^\star = \kappa(\theta)$ with

$$\kappa(\theta) = F_i\theta + f_i \text{ if } \theta \in \mathcal{P}_i, \tag{3}$$

where $\mathcal{P}_i = \{\theta \mid A_i\theta \leq b_i\}$, $i = 1, \dots, R$ are polyhedral critical regions of the parametric space, $A_i \in \mathbb{R}^{c_i \times n}$, $b_i \in \mathbb{R}^{c_i}$ are half-space representations of respective critical regions, R denotes the total number of regions, and $F_i \in \mathbb{R}^{m \times n}$, $f_i \in \mathbb{R}^m$ are parameters of locally affine expressions. Moreover, the critical regions satisfy $\text{int}(\mathcal{P}_i) \cap \text{int}(\mathcal{P}_j) = \emptyset$ for all $i \neq j$, and $\bigcup_i \mathcal{P}_i = \Omega$ with $\Omega = \{\theta \mid \exists U \text{ s.t. } GU \leq w + S\theta\}$ being the feasible set of (2). Finally, the function (3) is continuous, i.e., $F_i\theta + f_i = F_j\theta + f_j$ for all $\theta \in \mathcal{P}_i \cap \mathcal{P}_j$ for such combinations of i and j for which the intersection is not empty.

The reason why parametric programming is of interest is that once the properties of the map in (3), i.e., critical regions $\mathcal{P}_i$ and local expressions F_i, f_i, are available, computing the optimal value of $U^\star$ which solves (2) reduces to a mere evaluation of (3). Moreover, such an evaluation is division-free, i.e., only additions and multiplications are required.

III. Region-based parametric quadratic programming

The PWA solution in (3) is constructed by investigating all optimal combinations of constraints that can be active in (2). To simplify the technical exposition, we first rewrite (2) into

$$\min_{z} \ 1/2\, z^T H z \tag{4a}$$

$$\text{s.t. } Gz \leq w + S\theta. \tag{4b}$$

Note that (4) is equivalent to (2) with $z = U + H^{-1}F^T\theta$ and $S = E + GH^{-1}F^T$. Once $z^\star$ is available, $U^\star$ in (2) and (3) is obtained by $U^\star = z^\star - H^{-1}F^T\theta$. Next, rewrite (4) into

$$\min_{z} \ 1/2\, z^T H z \tag{5a}$$

$$\text{s.t. } G_{\mathcal{A}} z = w_{\mathcal{A}} + S_{\mathcal{A}}\theta, \tag{5b}$$

$$G_{\mathcal{N}} z < w_{\mathcal{N}} + S_{\mathcal{N}}\theta, \tag{5c}$$

where $M_{\mathcal{I}}$ is the matrix obtained from matrix M by retaining only rows indexed by $\mathcal{I}$, $\mathcal{A} \subseteq \{1, \dots, p\}$ is the index set of *active* constraints, and $\mathcal{N} \subseteq \{1, \dots, p\}$ is the index set of *inactive* constraints. The index sets $\mathcal{A}$ and $\mathcal{N}$ are disjoint, i.e., $\mathcal{A} \cap \mathcal{N} = \emptyset$, and satisfy $\mathcal{A} \cup \mathcal{N} = \{1, \dots, p\}$.

A. One critical region

Consider a particular realization of $\mathcal{A}$ and $\mathcal{N}$. The Karush-Kuhn-Tucker (KKT) conditions for (5) are given by

$$Hz^\star + G_{\mathcal{A}}^T\lambda^\star + G_{\mathcal{N}}^T\mu^\star = 0, \tag{6a}$$

$$G_{\mathcal{A}} z^\star = w_{\mathcal{A}} + S_{\mathcal{A}}\theta, \tag{6b}$$

$$G_{\mathcal{N}} z^\star < w_{\mathcal{N}} + S_{\mathcal{N}}\theta, \tag{6c}$$

$$\lambda^\star \geq 0, \tag{6d}$$

$$\mu^\star \geq 0, \tag{6e}$$

$$\lambda^{\star T}(G_{\mathcal{A}} z^\star - w_{\mathcal{A}} - S_{\mathcal{A}}\theta) = 0, \tag{6f}$$

$$\mu^{\star T}(G_{\mathcal{N}} z^\star - w_{\mathcal{N}} - S_{\mathcal{N}}\theta) = 0. \tag{6g}$$

Since $G_{\mathcal{N}} z^\star - w_{\mathcal{N}} - S_{\mathcal{N}}\theta < 0$ for all inactive constraints, from (6g) we conclude that $\mu^\star = 0$. Then from (6a) we get[1]

$$z^\star = -H^{-1}G_{\mathcal{A}}^T\lambda^\star. \tag{7}$$

Substituting (7) into (6b), and we obtain[2]

$$\lambda^\star = -(G_{\mathcal{A}}H^{-1}G_{\mathcal{A}}^T)^{-1}(w_{\mathcal{A}} + S_{\mathcal{A}}\theta), \tag{8}$$

which can be written as

$$\lambda^\star = Q(\mathcal{A})\theta + q(\mathcal{A}) \tag{9}$$

where

$$Q(\mathcal{A}) = -(G_{\mathcal{A}}H^{-1}G_{\mathcal{A}}^T)^{-1}S_{\mathcal{A}}, \tag{10a}$$

$$q(\mathcal{A}) = -(G_{\mathcal{A}}H^{-1}G_{\mathcal{A}}^T)^{-1}w_{\mathcal{A}}. \tag{10b}$$

Plugging (8) into (7) we finally get

$$z^\star = H^{-1}G_{\mathcal{A}}^T(G_{\mathcal{A}}H^{-1}G_{\mathcal{A}}^T)^{-1}(w_{\mathcal{A}} + S_{\mathcal{A}}\theta), \tag{11}$$

which can be written as

$$z^\star = F(\mathcal{A})\theta + f(\mathcal{A}) \tag{12}$$

[1] Note that $H \succ 0$ in (4a) is assumed.

[2] At this point we assume that the problem is either not degenerate, in which case $G_{\mathcal{A}}$ is of full row rank, or that at most m linearly independent rows are obtained from $G_{\mathcal{A}}$, see [7].

with

$$F(\mathcal{A}) = H^{-1}G_{\mathcal{A}}^T(G_{\mathcal{A}}H^{-1}G_{\mathcal{A}}^T)^{-1}S_{\mathcal{A}}, \tag{13a}$$

$$f(\mathcal{A}) = H^{-1}G_{\mathcal{A}}^T(G_{\mathcal{A}}H^{-1}G_{\mathcal{A}}^T)^{-1}w_{\mathcal{A}}. \tag{13b}$$

The subset of the parametric space where $z^\star$ from (11) and $\lambda^\star$ from (8) satisfy[3] primal feasibility (6c) and dual feasibility (6d) constitutes the critical region

$$\mathcal{P}(\mathcal{A}) = \{\theta \mid G_{\mathcal{N}}z^\star < w_{\mathcal{N}} + S_{\mathcal{N}}\theta,\ \lambda^\star \geq 0\}, \tag{14}$$

which is a polyhedron in half-space representation. In the sequel, we will consider the closure of (14) obtained by replacing strict inequalities by non-strict ones

$$\mathcal{P}(\mathcal{A}) = \{\theta \mid A(\mathcal{A})\theta \leq b(\mathcal{A})\}, \tag{15}$$

with

$$A(\mathcal{A}) = \begin{bmatrix} G_{\mathcal{N}}F(\mathcal{A}) - S_{\mathcal{N}} \\ -Q(\mathcal{A}) \end{bmatrix},\ b(\mathcal{A}) = \begin{bmatrix} w_{\mathcal{N}} - G_{\mathcal{N}}f(\mathcal{A}) \\ q(\mathcal{A}) \end{bmatrix}. \tag{16}$$

B. Generation of all optimal active sets

To construct the full PWA solution as in (3) it is necessary to apply the procedure of Section III-A to all optimal active sets $\mathcal{A}_1, \ldots, \mathcal{A}_R$. Two distinct classes of methods can be applied to obtain the list of optimal active sets:

- geometric approaches of [2] and [1],
- extensive enumeration procedure of [6].

The geometric approach is based on constructing an initial critical region by picking an arbitrary $\theta_1 \in \Omega$. For this value of the parameter, the pQP (4) is solved as a QP which yields the information about the $\mathcal{A}_1$ optimal for a given θ. Given $\mathcal{A}_1$, the expression for $z^\star = F_1\theta + f_1$ is then obtained from (11) with $F_1 = F(\mathcal{A}_1)$ and $f_1 = f(\mathcal{A}_1)$ via (13). Then, the critical region $\mathcal{P}_1$ is formed by (15). Subsequently, a new θ satisfying $\theta \notin \mathcal{P}_1$ is selected. In [2] this is achieved by performing set difference operations, while [1] selects the new point by stepping over facets of the critical region. In either case, a new $\mathcal{A}_2$ is obtained by solving the QP for the new parameter, a new local optimizer $z^\star = F_2\theta + f_2$ is formed, and the associated critical region $\mathcal{P}_2$ is created. The procedure is then repeated recursively until the whole search space, i.e., Ω, is covered. It is important to note that critical regions per (15) play an essential role in this type of algorithms, since new regions can only be constructed based on existing ones.

The extensive enumeration approach, on the other hand, can generate optimal active sets without having to construct the critical regions. The procedure first enumerates all possible combinations of active constraints and organizes them in a tree in the order of increasing cardinality. Since there are p constraints in (4) and m decision variables, at least no constraint would be active (which corresponds to the root node with $\mathcal{A} = \emptyset$), and at most m constraints could be active if no degeneracy is assumed due to LICQ conditions [7]. At the k-th level of the tree (which corresponds to k constraints being active), the tree contains $\binom{p}{k}$ candidate s. In total, the number of candidate s is

$$R_{\text{max}} = \sum_{k=0}^{m} \frac{p!}{k!(p-k)!}. \tag{17}$$

Clearly, (17) indicates that the number of regions can quickly become impractically large as p and/or m increase. However, not all candidates need to be considered. If, say, we determine that the 3rd and the 5th constraint cannot be simultaneously active (which is a case e.g. when these two constraints represent, respectively, the lower and the upper bound of a decision variable), then all subsets containing the 3rd and the 5th constraints will be infeasible as well. This allows to prune the tree of candidate s to a certain extent. Moreover, not all feasible s will be optimal. To determine optimality of a particular candidate $\mathcal{A}$, the authors in [6] propose to solve the linear program in the decision variables u, θ, λ, and t,

$$\max_{t,z,\theta,\lambda}\ t \tag{18a}$$

$$\text{s.t. } Hz + G_{\mathcal{A}}^T\lambda = 0, \tag{18b}$$

$$G_{\mathcal{A}}z = w_{\mathcal{A}} + S_{\mathcal{A}}\theta, \tag{18c}$$

$$t \leq w_{\mathcal{N}} + S_{\mathcal{N}}\theta - G_{\mathcal{N}}z, \tag{18d}$$

$$\lambda \geq t, \tag{18e}$$

$$t \geq 0, \tag{18f}$$

with $\mathcal{N} = \{1, \ldots, p\} \setminus \mathcal{A}$. If (18) is feasible with $t^\star > 0$, the candidate $\mathcal{A}$ is a feasible optimal which yields a full-dimensional critical region. If the LP is infeasible, a new LP is solved by dropping the stationarity constraint (18b). If this modified LP is infeasible, the candidate and, more importantly, all other candidates that are a superset of $\mathcal{A}$ are infeasible and can be removed from consideration. The authors in [6] propose to construct the PWA solution in (3) in two steps:

1) Enumerate all optimal active sets by exploring the tree of all possible combinations, using the LP (18) as a pruning criterion.
2) Once all optimal active sets, i.e., $\mathcal{A}_1, \ldots, \mathcal{A}_R$ are enumerated, construct the local optimizers per (11) and critical regions via (14).

The common feature of approaches [2] and [6] is that they construct critical regions at certain stage. The geometric approach [2] requires the regions to be constructed at each intermediate step and uses them to generate new s. The enumeration approach of [6], on the other hand, only constructs the regions at the very end.

The memory required to store the PWA function (3) is proportional to the number of critical region. For each such region, we need to store:

- The local affine optimizer in (12), i.e., the matrix $F_i \in \mathbb{R}^{m\times n}$ and the vector $f_i \in \mathbb{R}^m$. This requires $m\times(n+1)$ real numbers.

[3] The complementary slackness condition (6f) is trivially satisfied since $G_{\mathcal{A}}z^\star - w_{\mathcal{A}} - S_{\mathcal{A}}\theta = 0$ holds for active constraints. Moreover, (6e) is always satisfied (hence redundant) since $\mu^\star = 0$ is the only feasible choice due to (6g) and inactivity of constraints indexed by $\mathcal{N}$.

- The half-space representation of the region per (15), i.e., the matrix $A_i \in \mathbb{R}^{c_i \times n}$ and the vector $b \in \mathbb{R}^{c_i}$, where c_i is the number of non-redundant defining half-spaces. Therefore $c_i \times (n+1)$ numbers are required.

In total, the PWA function (3) with R critical region consumes

$$\mathcal{M}_{\text{RB}} = \sum_{i=1}^{R} (m + c_i)(n+1) \tag{19}$$

real numbers (here, "RB" stands for "region-based"). This expression can be simplified by considering c_{avg} as the average number of half-spaces of all critical regions. Then the exact memory footprint of the PWA representation of $z^\star = \kappa(\theta)$ in (3) is

$$\mathcal{M}_{\text{RB}} = R(m + c_{\text{avg}})(n+1). \tag{20}$$

IV. Region-free approach of [4]

In [4] the authors have shown how to compute $z^\star$ which solves (4) using an algorithm that does not require storage of the critical regions. The underlying idea is to pre-compute, off-line, the factors in (8) and (11) for all possible optimal active sets. Let $\widetilde{w}(\theta) = w + S\theta$ and let $\widetilde{w}_{\mathcal{A}}(\theta)$ be the rows indexed by $\mathcal{A}$. Then, (7) can be written as

$$z^\star = H^{-1} G_{\mathcal{A}}^T (G_{\mathcal{A}} H^{-1} G_{\mathcal{A}}^T)^{-1} \widetilde{w}_{\mathcal{A}}(\theta) = \widetilde{F}(\mathcal{A}) \widetilde{w}_{\mathcal{A}}(\theta). \tag{21}$$

Similarly, (8) becomes

$$\lambda^\star = -(G_{\mathcal{A}} H^{-1} G_{\mathcal{A}}^T)^{-1} \widetilde{w}_{\mathcal{A}}(\theta) = \widetilde{Q}(\mathcal{A}) \widetilde{w}_{\mathcal{A}}(\theta). \tag{22}$$

With $\widetilde{F}(\mathcal{A}_i)$ and $\widetilde{Q}(\mathcal{A}_i)$ computed, off-line, for all optimal active sets $\mathcal{A}_1, \ldots, \mathcal{A}_R$, the task to be performed on-line then becomes to identify which is optimal for a given θ. In [4] this is achieved by Algorithm 1, which operates according to conventional methods, see, e.g., [5]. The main difference to numerical algorithms is that Alg. 1 uses prefactored expressions for $z^\star$ and $\lambda^\star$ (see (21) and (22)) instead of computing them on-line by inverting the KKT matrix.

Algorithm 1 Region-free method of [4]

```
INPUT: Factors $\widetilde{F}(\mathcal{A}_i)$, $\widetilde{Q}(\mathcal{A}_i)$, $i = 1, \ldots, R$, pQP data $G$,
    $w$, $S$ from (4), initial $\mathcal{A} \neq \emptyset$, query parameter $\theta$.
OUTPUT: $z^\star$ solving (4) for given $\theta$.
 1: $\widetilde{w}(\theta) \leftarrow w + S\theta$
 2: $i^\star \leftarrow \arg\min_i \widetilde{Q}_i(\mathcal{A}) \widetilde{w}_{\mathcal{A}}(\theta)$
 3: if $\widetilde{Q}_{i^\star}(\mathcal{A}) \widetilde{w}_{\mathcal{A}}(\theta) < 0$ then
 4:     $\mathcal{A} \leftarrow \mathcal{A} \setminus \mathcal{A}(i^\star)$
 5: else
 6:     $z \leftarrow \widetilde{F}(\mathcal{A}) \widetilde{w}_{\mathcal{A}}(\theta)$
 7:     $j^\star \leftarrow \arg\min_j (\widetilde{w}_j(\theta) - G_j z)$
 8:     if $G_{j^\star} z > \widetilde{w}_{j^\star}(\theta)$ then
 9:         $\mathcal{A} \leftarrow \mathcal{A} \cup j^\star$
10:     else
11:         return $z^\star \leftarrow z$
12:     end if
13: end if
14: goto 2
```

Notice that Alg. 1 does not require storing the critical regions. Instead, it iteratively builds the optimal by removing (cf. Step 4) or adding (cf. Step 9) constraints one at a time. Then, optimality of the current iterate of the active set is checked by verifying dual (Step 3) and primal feasibility conditions (Step 8).

The memory required to run Alg. 1 on-line is determined by the storage of the factors $\widetilde{F}(\mathcal{A}_i)$, $\widetilde{Q}(\mathcal{A}_i)$ for $i = 1, \ldots, R$ and by the pQP data G, w, and S. Here, $\widetilde{F}(\mathcal{A}_i) \in \mathbb{R}^{m \times a_i}$ and $\widetilde{Q}(\mathcal{A}_i) \in \mathbb{R}^{a_i \times a_i}$ with $a_i = |\mathcal{A}_i|$ being the cardinality of the corresponding active set. The pQP data $G \in \mathbb{R}^{p \times m}$, $w \in \mathbb{R}^p$, and $S \in \mathbb{R}^{p \times n}$, on the other hand, have fixed size, but are stored just once and are shared among all active sets. The memory footprint of the input data is thus

$$\mathcal{M}_{\text{RF}} = p(m + n + 1) + \sum_{i=1}^{R} (m|\mathcal{A}_i| + |\mathcal{A}_i|^2), \tag{23}$$

where "RF" stands for "region-free". Let a_{avg} be the average cardinality of $\mathcal{A}_1, \ldots, \mathcal{A}_R$. Then, (23) becomes

$$\mathcal{M}_{\text{RF}} = p(m + n + 1) + R(m a_{\text{avg}} + a_{\text{avg}}^2). \tag{24}$$

V. Novel Results

Although Algorithm 1 does not require storage of the critical regions and thus requires a smaller memory footprint, it is an iterative procedure which does not admin a closed-form solution. In what follows we first show that such a closed-form solution exists. Subsequently, in Section V-C we show that the memory footprint of Alg. 1 can be further reduced at the expense of performing more on-line calculations. Also this reduced representation admits a closed-form representation. Moreover, for the two types of closed-form solutions we show how they can be recovered from the region-based PWA function (3) and vice versa.

All results of this section assume that all optimal active sets for the pQP (4), i.e., $\mathcal{A}_1, \ldots, \mathcal{A}_R$, were obtained by the extensive enumeration approach of [6]. This is done as follows:

1) Enumerate all possible combinations of active constraints with cardinality $0, \ldots, m$.
2) For each candidate active set solve the LP (18).
    a) If the LP is feasible, add the candidate to the list of optimal active sets.
    b) If the LP is infeasible, drop (18b). If the augmented LP is infeasible, discard the candidate, as well as all other candidates which are its supersets.

Note that the generation of the list of optimal active sets does *not* require construction of critical regions.

A. Region-Free Closed-Form Solution

Given $\mathcal{A}_1, \ldots, \mathcal{A}_R$ as the list of optimal active sets, compute $F_i = F(\mathcal{A}_i)$, $f_i = f(\mathcal{A}_i)$ per (12), along with $Q_i = Q(\mathcal{A}_i)$, $q_i = q(\mathcal{A}_i)$ via (10).

Theorem 5.1: The function $z^\star = \kappa(\theta)$ with

$$\kappa(\theta) = F_i\theta + f_i \text{ if } Q_i\theta + q_i \geq 0 \;\wedge\; G(F_i\theta + f_i) \leq w + S\theta \tag{25}$$

is the optimal solution to (4) for any $\theta \in \Omega$. ■

Corollary 5.2: (25) is a PWA function over polyhedra. ■

The consequence of Theorem 5.1 is that (25) provides a closed-form solution to (4) which does not require storage of the critical regions. Instead, the i-th rule is deemed optimal by checking primal and dual feasibility. Since the pQP data G, w, and S are stored just once and shared among all IF-THEN rules, it follows that (25) offers a smaller memory footprint compared to (3). Specifically, for each $i = 1, \ldots, R$ we have $F_i \in \mathbb{R}^{m \times n}$, $f_i \in \mathbb{R}^m$, $Q_i \in \mathbb{R}^{a_i \times n}$, and $q_i \in \mathbb{R}^{a_i}$ where $a_i = |\mathcal{A}_i|$. Therefore the footprint of (25) is

$$\mathcal{M}_{\text{RF}\lambda} = p(m+n+1) + Rm(n+1) + \sum_{i=1}^{R} a_i(n+1), \tag{26}$$

where the first term accounts for G, w, and S, the second term is the storage space of F_i, f_i, and the last one represents memory occupied by Q_i and q_i. Using a_{avg} as the average cardinality of all active sets, (26) becomes

$$\mathcal{M}_{\text{RF}\lambda} = p(m+n+1) + R(m + a_{\text{avg}})(n+1). \tag{27}$$

Comparing (27) to (20), we see that (25) requires less memory storage than (3) if

$$\frac{pm}{R(n+1)} + a_{\text{avg}} < c_{\text{avg}}. \tag{28}$$

We remark that a_{avg} is upper bounded by m, and $c_{\text{avg}} \leq p$, in general. If $R \gg pm$, then (28) simplifies to $m < p$, which is always satisfied if all decision variables are lower/upper bounded in (4b).

B. *Analogy between* (3) *and* (25)

The region-based PWA function in (3) can be converted into the region-free format (25) as follows.

1) For each critical region of (3):
   a) Pick a $\theta \in \mathcal{P}_i$, e.g., as the center of the largest inscribed ball[4].
   b) Compute $z^\star = F_i\theta + f_i$.
   c) Plug $z^\star$ and θ into (4b) and obtain the index set $\mathcal{A}_i$ of constraints active in the i-th region.
   d) Construct $Q_i = Q(\mathcal{A}_i)$, $q_i = q(\mathcal{A}_i)$ per (10) and $F_i = F(\mathcal{A}_i)$, $f_i = f(\mathcal{A}_i)$ via (12).
2) Construct (25) using F_i, f_i, Q_i, q_i and the pQP data from (4b).

Since the PWA function in (25) typically uses less memory than (3) (cf. (28)), this procedure can be viewed as a memory compression of (3).

The region-based function in (3) can be recovered from (25) as follows:

1) For each $i = 1, \ldots, R$:
   a) Construct the half-space representation of the i-th critical region in (15) via (16) by using $Q(\mathcal{A}_i) = Q_i$, $q(\mathcal{A}_i) = q_i$, $F(\mathcal{A}_i) = F_i$, and $f(\mathcal{A}_i) = f_i$.
   b) Optionally remove redundant half-spaces from (15).
2) Recover (3) from $\mathcal{P}_i$, F_i, and f_i.

Therefore, it is possible to employ the region-free function (25) in algorithms that require the region-based format (3).

C. *Region-Free Solution with Reduced Memory Footprint*

In this section we show how the amount of data required to run Alg. 1 can be further reduced by performing more computations on-line. The idea is based on replacing the calculation of the primal optimizer candidate in Step 6 of Alg. 1 by (7). Specifically, Step 6 computes the candidate z by $z = \widetilde{F}(\mathcal{A})\widetilde{w}_{\mathcal{A}}(\theta)$ using the pre-factored expressions of $\widetilde{F}(\mathcal{A})$ obtained from (21). Using (7) and (22) we can equivalently compute the primal optimizer candidate by

$$z = -H^{-1}G_{\mathcal{A}}^T\widetilde{Q}(\mathcal{A})\widetilde{w}_{\mathcal{A}}(\theta). \tag{29}$$

Here, instead of storing $\widetilde{F}(\mathcal{A}_i)$ for each $i = 1, \ldots, R$ as in (21), we only need to store the inverted Hessian H^{-1}, which is unique for each active set. Therefore Step 6 of Alg. 1 can be altered to $z \leftarrow -H^{-1}G_{\mathcal{A}}^T\widetilde{Q}(\mathcal{A})\widetilde{w}_{\mathcal{A}}(\theta)$. Note that the product $\widetilde{Q}(\mathcal{A})\widetilde{w}_{\mathcal{A}}(\theta)$ was already computed at Step 2. Under such a modification, storing $\widetilde{F}(\mathcal{A}_i)$ in Alg. 1 is no longer required.

By removing $\mathcal{F}(\mathcal{A}_i) \in \mathbb{R}^{m \times a_i}$, $a_i = |\mathcal{A}_i|$, $i = 1, \ldots, R$ as the input of Alg. 1, we can save the space needed to store $m\sum_{i=1}^{R} a_i$ real numbers. However, we need to store the inverted Hessian, which requires m^2 real numbers. Thus the total memory footprint of Alg. 1 with Step 6 modified as described above is

$$\mathcal{M}_{\text{RF}\lambda} = p(m+n+1) + m^2 + Ra_{\text{avg}}^2. \tag{30}$$

Therefore the total memory reduction compared to (24) is $Rma_{\text{avg}} - m^2$ real numbers.

Next we show that the memory-reduced version of Alg. 1 also admits a closed-form solution.

Theorem 5.3: Consider the function $z^\star = \kappa(\theta)$ with

$$\begin{aligned} \kappa(\theta) = &- H^{-1}G_{\mathcal{A}_i}^T(Q_i\theta + q_i) \text{ if } (Q_i\theta + q_i) \geq 0 \,\wedge \\ &- GH^{-1}G_{\mathcal{A}_i}^T(Q_i\theta + q_i) \leq w + S\theta. \end{aligned} \tag{31}$$

Then, (31) is the parametric solution to (4). ■

Corollary 5.4: (31) is a PWA function over polyhedra. ■

The evaluation of $z^\star = \kappa(\theta)$ can be performed by Algorithm 2. It goes through all modified active set candidates in a sequential order and first computes the Lagrange multipliers using the explicit relation (9) in Step 2. Then, it validates the dual feasibility condition (6c) for the candidate active set $\mathcal{A}_i$ in Step 3. If dual feasibility holds, the algorithm subsequently computes the primal optimizer in Step 4 using the information of which rows of G should be active. Afterwards, primal feasibility of inactive constraints is checked in Steps 5 and 6. If both dual and primal feasibility conditions are satisfied, the candidate $\mathcal{A}_i$ is the optimal active set for the current θ, and the procedure returns the optimal decision in Step 7.

[4] This can be done by solving one LP.

Algorithm 2 Region-free sequential search

```
INPUT: Matrices $Q_i$, $q_i$, $i = 1, \dots, R$ from (10), list of
    optimal active sets $\{\mathcal{A}_1, \dots, \mathcal{A}_R\}$, pQP data $H^{-1}$, $G$, $w$
    $S$ from (4), query parameter $\theta$.
OUTPUT: $z^\star$ solving (4) for given $\theta$.
for $i = 1, \dots, R$ do
    $\lambda \leftarrow Q_i\theta + q_i$
    if $\lambda \geq 0$ then
        $z \leftarrow -H^{-1}G_{\mathcal{A}_i}^T\lambda$
        $\mathcal{N}_i \leftarrow \{1, \dots, p\} \setminus \mathcal{A}_i$
        if $G_{\mathcal{N}_i}z < w_{\mathcal{N}_i} + S_{\mathcal{N}_i}\theta$ then
            return $z^\star \leftarrow z$
        end if
    end if
end for
return $z^\star \leftarrow \emptyset$
```

VI. Complexity Comparison

In this section we asses how the proposed region-free approach of Section V-C compares to the conventional region-based procedure represented by (3) and the region-free approach of [4] in terms of memory complexity. To perform such a comparison, we have assumed that the pQP was generated from the MPC problem formulated for the prediction model $1/(s+1)^n$, discretized with sampling time of 1 second and converted to a state-space form. Here, n represents the number of states of the prediction model. The states of the discretized system were constrained by $-10 \leq x_i \leq 10$, $i = 1, \dots, n$. Input constraints $-1 \leq u \leq 1$ were considered as well. The number of decision variables, i.e., m, was controlled via the prediction horizon, i.e., $m = N$.

Table I shows the results of the exact memory complexity for varying dimensionality of the pQP in (4). Columns of the table represent, respectively:

- n: the parametric dimension which, in our case, corresponds to the number of states of the controlled system, m: the number of optimization variable, equal to the prediction horizon, p: the number of constraints of the pQP (4);
- R: the number of critical regions in (3) and optimal active sets in (31);
- $\mathcal{M}_{\text{RB}}$: the number of real numbers required to store the region-based PWA function (3);
- $\mathcal{M}_{\text{RF}}$: the number of real numbers required for the active set Algorithm 1, computed by (24);
- $\mathcal{M}_{\text{RF}\lambda}$: the number of real numbers required for the memory-reduced active set algorithm of Section V-C, computed by (30);
- $\Delta = \mathcal{M}_{\text{RF}}/\mathcal{M}_{\text{RF}\lambda}$: the memory reduction factor.

In the table, memory consumption is reported in megabytes assuming that each real number is stored in double precision arithmetics using 8 bytes.

VII. Conclusions

In this paper we have revisited the idea of region-free explicit MPC, which was originally suggested in [4]. We have shown that by avoiding the computation and the storage of the factors for the primal optimizer the amount of data required to compute optimal control actions can be reduced by a factor of two, on average. Furthermore, we derived a closed-form representation of the region-free description of the feedback law. During its construction, generation of critical regions is avoided and is replaced by enumeration of optimal active sets. Such a closed-form solution is not only simpler compared to the conventional region-based form, but it allows to rigorously analyze the closed-loop system. Finally, we have shown that the memory-reduced version of the region-free description also admits a closed-form solution.

$n/m/p$	R	$\mathcal{M}_{\text{RB}}$	$\mathcal{M}_{\text{RF}}$	$\mathcal{M}_{\text{RF}\lambda}$	Δ
10/2/52	274	0.661	0.020	0.013	1.6
10/3/66	1 867	3.696	0.233	0.115	2.0
10/4/80	7 134	12.066	1.452	0.685	2.1
10/5/94	19 582	32.386	5.769	2.637	2.2
15/2/84	577	2.773	0.045	0.028	1.6
15/3/106	6 618	28.172	0.837	0.413	2.0
15/4/128	46 135	183.775	9.715	4.623	2.1
20/2/114	883	7.753	0.071	0.045	1.6
20/3/146	11 932	84.494	1.509	0.743	2.0
20/4/176	93 002	652.818	19.278	9.121	2.1
30/2/172	1 279	27.373	0.118	0.080	1.5
30/3/224	27 879	489.625	3.625	1.800	2.0
40/2/230	1 455	52.911	0.160	0.117	1.4
40/3/298	27 544	949.723	3.609	1.810	2.0

TABLE I

Memory occupancy in megabytes for various problem sizes.

Acknowledgments

M. Kvasnica, B. Takács, and J. Holaza gratefully acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under grant 1/0403/15, and the financial support of the Slovak Research and Development Agency under the project APVV 0551-11. This research was supported by Mitsubishi Electric Research Laboratories, under a Collaborative Research Agreement.

References

[1] M. Baotić. *Optimal Control of Piecewise Affine Systems – a Multi-parametric Approach*. Dr. sc. thesis, ETH Zurich, Zurich, Switzerland, March 2005.

[2] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, January 2002.

[3] F. Borrelli. *Constrained Optimal Control Of Linear And Hybrid Systems*, volume 290 of *Lecture Notes in Control and Information Sciences*. Springer, 2003.

[4] F. Borrelli, M. Baotić, J. Pekar, and G. Stewart. On the computation of linear model predictive control laws. *Automatica*, 46(6):1035–1041, 2010.

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

[6] A. Gupta, S. Bhartiya, and P. Nataraj. A novel approach to multiparametric quadratic programming. *Automatica*, 47(9):2112–2117, 2011.

[7] J. Spjøtvold, P. Tøndel, and T. A. Johansen. A Method for Obtaining Continuous Solutions to Multiparametric Linear Programs. In *IFAC World Congress*, Prague, Czech Republic, 2005.