Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Some Necessary and Sufficient Conditions for Correctness of Linear Machine in Presence of Numerical Errors

Aditi Sharma, **Mani Bhushan** *

* *Department of Chemical Engineering, Indian Institute of Technology Bombay, India (e-mail: aditis0997@gmail.com,mbhushan@iitb.ac.in)*

Abstract: Linear machine (LM) has been recently proposed (Airan et al., 2017) for solving the point location problem which arises in explicit model predictive control (e-MPC). LM associates a linear discriminant function with each critical region identified in the offline phase in e-MPC. The solution to the online point location problem in the LM approach then simply corresponds to the region whose discriminant function attains the largest value amongst all the discriminant functions. LM involves two steps: (i) identification of neighbouring critical regions, and (ii) finding the discriminant functions by writing constraints involving discriminant functions of neighbouring pairs of regions. Both these steps involve solving linear programming (LP) problems. Similar to any other optimization problem, the constraints of the LP are satisfied with some tolerances. Even though theoretically sound, the resulting LM may not accurately identify the critical region due to the numerical errors arising from these tolerances. In the current work, we identify some conditions which can be used as an aid by the user to judge the accuracy of LM results. In particular, we give a necessary condition for step (i) whose violation will yield incorrect misclassification for some point location problems. We also propose a sufficient condition whose satisfaction guarantees the accuracy of linear machine solution despite numerical errors which may have crept in during step (ii) of the LM design. This condition needs to be evaluated for each specified point during the point location phase. We illustrate these ideas on the well known quadruple tank system.

Keywords: point location, explicit model predictive control, multi-parametric programming

1. INTRODUCTION

Over the last few decades, model predictive control (MPC) has emerged as the preferred advanced control tool for many industries such as refineries and power plants (Qin and Badgwell, 2003), and automobiles (Hrovat et al., 2012). MPC involves solving an optimization problem at each sampling instant. Solving the optimization problem in real-time is a challenging activity for large dimensional systems or systems with fast sampling.

Over the years, explicit-MPC (e-MPC) has emerged as an attractive option for mitigating the computational challenges associated with online optimization for linear systems (Bemporad et al., 2000). Based on multi-parametric programming, e-MPC is based on the insight that the structure of optimization problems solved by MPC does not change with time, and only the values of the optimization-parameters such as the current states and input variables change. e-MPC thus has two phases- an offline phase where the optimization problem is analyzed offline to obtain the so-called critical regions in the parametric space and the associated control law for each critical region. The online phase corresponds to the online implementation of e-MPC wherein the pre-computed solution is evaluated at the parameter values which become available online. This in turn, requires solving a point location problem to identify the critical region containing the parameter values realized at the current time instant. Thus, the real-time MPC optimization problem is converted to solving a point location problem. For a problem with large number of critical regions, solving the point location problem in real-time is itself challenging.

The point location problem has garnered significant interest in the research community (Oberdieck et al., 2016) and several approaches have been proposed such as binary search tree (Tøndel et al., 2003), multi-way search tree (Mönnigmann and Kastsian, 2011), hash tables (Bayat et al., 2011), descriptor function (Baotić et al., 2008) and linear machine (Airan et al., 2017).

In the current work, we focus on the linear machine (LM) approach (Airan et al., 2017) which draws inspiration from multi-category pattern classification. A key feature of the LM approach is that the required online computational complexity is constant for all points in the parametric space and is independent of the critical region containing a specific realization of the parameter (Airan et al., 2017). This facilitates appropriate hardware design for e-MPC implementation for a given linear system.

The LM approach associates a linear discriminant function with each critical region which was identified in the offline phase in e-MPC. The solution to the online point location problem in the LM approach then corresponds to the region with the highest discriminant function value.

Generation of the LM involves two steps (Airan et al., 2017): (i) identification of neighbouring critical regions, and (ii) finding the discriminant functions by writing appropriate constraints involving discriminant functions of neighbouring pairs of regions. Both these steps involve solving linear programming (LP) problems. Similar to any other optimization problem, the numerical solution of LP requires user to specify various tolerances. Even though theoretically sound, the resulting LM may not accurately identify the critical region due to the numerical errors arising from these tolerances. In the current work, we identify some conditions which can be used as an aid by the user to judge the accuracy of LM results. In particular, we give a necessary condition for step (i) whose violation will yield misclassification for some point location problems. We also propose a sufficient condition whose satisfaction guarantees the accuracy of LM solution despite numerical errors which may have crept in during step (ii) of its design. This condition needs to be evaluated for each specified point during the point location phase. We demonstrate these ideas on the well- known quadruple tank system.

The rest of the article is organized as follows: Section 2 briefly discusses relevant background material; Section 3 discusses quadruple tank problem; Section 4 discusses numerical issues in design of linear machine; Section 5 presents our main contribution, namely determination of a few conditions to check the correctness of neighbour identification procedure as well as the obtained linear machine. Finally, Section 6 concludes the paper.

2. RELEVANT BACKGROUND

2.1 Explicit Model Predictive Control

The conventional linear MPC problem can be written as a multi-parametric quadratic programming (mp-QP) problem as (Bemporad et al. (2000)),

$$\min_z \frac{1}{2}(z^T H z) \text{ s.t. } Gz \leq W + Sx \tag{1}$$

where, $z \in \mathbb{R}^p$ is the vector of decision variables, $x \in \tilde{X} \subset \mathbb{R}^d$ is a vector of parameters and H, G, W, S are constant matrices. $X \subset \tilde{X}$ denotes the set of feasible x also known as the parametric space. Bemporad et al. (2000) show that the explicit solution to (1) is given by piecewise affine (PWA) functions of the parameter vector x associated with respective critical regions (polyhedral partitions) of X. The critical regions are defined as:

$$CR^i = \{x \in X | A^i x \leq b^i\}, \quad i = 1, 2, .., n_r \tag{2}$$

where, CR^i refers to the i^{th} critical region, n_r is the number of critical regions, and $A^i \in \mathbb{R}^{n_{f_i} \times d}$ and $b^i \in \mathbb{R}^{n_{f_i}}$ define the inequalities which characterize the i^{th} critical region (Airan et al., 2017). The intersection of CR^i and the hyperplane $A^i_j x = b^i_j$ defining the j^{th} inequality of CR^i is known as the j^{th} facet of CR^i. The CRs have disjoint interiors, $CR^i \cap CR^k = \phi$, $\forall i \neq k$ and CRs encompass the entire parametric space $\cup_{i=1}^{n_r} CR^i = X$ (Spjøtvold et al., 2006). The polyhedral partition $\{CR^i\}_1^{n_r}$ will be represented as $\mathcal{P}$. The PWA explicit solution is:

$$z^*(x) = \Omega^i x + \omega^i, \quad \forall x \in CR^i, i = 1, 2, .., n_r \tag{3}$$

where, $\Omega^i \in \mathbb{R}^{p \times d}$ and $\omega^i \in \mathbb{R}^p$. These PWA functions present the solution to the mP-QP problem and are evaluated to obtain the values of the manipulated variable to be implemented in the process in real-time.

2.2 Linear Machine

In the pattern classification area, linear machine is a classifier for multi-category classification problems (Duda et al., 2001). The point location problem in e-MPC is similar to a pattern classification problem if the critical regions are considered equivalent to classes. The linear machine classifier $\mathcal{L} \triangleq \{g^i(x)\}_{i=1}^{n_r}$ associates a linear function $g^i(x)$, known as linear discriminant function, with each critical region i such that (Airan et al., 2017):

$$g^i(x) > g^k(x), \ \forall x \in CR^i, \ k = 1, 2, \ldots, n_r, \ k \neq i \tag{4}$$

The linear discriminant functions are of the following form:

$$\begin{aligned} g^i(x) &= \alpha^i_1 x_1 + \alpha^i_2 x_2 + \ldots + \alpha^i_d x_d + \alpha^i_0 \\ &= (\alpha^i)^T x + \alpha^i_0, \quad i = 1, 2, \ldots, n_r \end{aligned} \tag{5}$$

Given the CRs (2) in the parametric space, designing LM involves obtaining coefficients $\alpha^i \in \mathbb{R}^d$ and $\alpha^i_0 \in \mathbb{R}$, $i = 1, 2, \ldots, n_r$ such that (4) is satisfied.

To obtain LM, Airan et al. (2013) used the hyperplanes of adjacent regions as natural switching boundaries where the discriminant functions for adjacent regions switch the inequality sign in (4). In particular, Airan et al. (2013) proposed a two-step procedure:
(1) Identifying all adjacent (neighbouring) pairs of critical regions (referred here as adjacency oracle).
(2) Determining linear discriminant functions based on the adjacency oracle provided by step (1).
This is briefly discussed next (Airan et al. (2013, 2017)).

Determination of adjacent critical regions : Critical regions CR^i and CR^k are defined as adjacent if $CR^i \cap CR^k$ is $d-1$ dimensional. To check for adjacency, Airan et al. (2013) first checked if they shared a common hyperplane by comparing the defining inequalities of the two regions under consideration. Regions which did not share a common hyperplane could not be adjacent regions. However, Formulation I (6) was solved for regions which shared a common facet corresponding to hyperplane $H^i_{j^*}$.

Formulation I: Adjacent Region Identification

$$\max_{t,x} \quad t \tag{6a}$$

$$\text{s.t. } x \in (CR^i \cap CR^k \cap H^i_{j^*}) \tag{6b}$$

$$t \leq \frac{-A^i_l x + b^i_l}{||A^i_l||}, \quad l = 1, 2, ..., n_{f_i}, \ l \neq j^* \tag{6c}$$

$$t \leq \frac{-A^k_m x + b^k_m}{||A^k_m||}, \quad m = 1, 2, ..., n_{f_k}, \ m \neq j^* \tag{6d}$$

Formulation I is a linear programming (LP) formulation which identifies if two CRs are adjacent, where A^i_l corresponds to the l^{th} row of the i^{th} region's A matrix and n_{f_i} represents the number of facets in the i^{th} region (similarly for region k). Formulation I attempts to identify a point x on the j^{*th} facet of the two regions which is farthest possible from the other facets of the two regions. The two regions are declared as adjacent i.e. their intersection is $d-1$ dimensional, if the LP is feasible and $t > 0$.

Determination of discriminant functions : This step is executed after the adjacency oracle from the previous step becomes available. The key idea in this step is to write down the relationship between discriminant functions of adjacent regions. To illustrate, let CR^i and CR^k be adjacent regions corresponding to common hyperplane $H^{i,k} = \{x \in X | A^{i,k}x = b^{i,k}\}$. Further, let CR^i be on the negative side ($A^{i,k}x \leq b^{i,k}$) of this hyperplane and CR^k be on the positive side ($A^{i,k}x \geq b^{i,k}$). From the definition of linear machine (4), it can be concluded that:

$$-g^i(x) + g^k(x) = \beta^{i,k}(A^{i,k}x - b^{i,k}),\ \forall (i,k) \in NR \tag{7}$$

where, $\beta^{i,k} > 0$ is a scaling factor and NR is a set containing all the pairs of indices corresponding to adjacent regions. Equation (7) states that discriminant function of CR^i attains values greater than the discriminant function of CR^k on the negative side of the common hyperplane and vice-versa. Further, the two discriminant functions attain equal values for points lying on the common hyperplane. Using (5), (7) can be written as:

$$\begin{aligned}(-\alpha^i + \alpha^k)^T x - \alpha_0^i + \alpha_0^k = \beta^{i,k}(A^{i,k}x - b^{i,k}),\\ \beta^{i,k} > 0,\ \ \forall (i,k) \in NR\end{aligned} \tag{8}$$

Equation (8), written for all n_f pairs of adjacent regions determined by Formulation I, can be expressed as:

$$Gy = 0 \tag{9}$$

where $G \in \mathbb{R}^{(n_f(d+1))\times(n_r(d+1)+n_f)}$ is a known matrix and the vector $y \in \mathbb{R}^{(n_r(d+1)+n_f)}$ of unknowns is:

$$y \triangleq [(\alpha^1)^T\ \alpha_0^1\ (\alpha^2)^T\ \alpha_0^2\ ..(\alpha^{n_r})^T\ \alpha_0^{n_r}\ \beta_1\ \beta_2\ ..\beta_{n_f}]^T \tag{10}$$

with $\beta_1, \beta_2, \ldots, \beta_{n_f}$ being the n_f scale factors. Equation (9) may have multiple solutions since any $y \in$ Null Space(G) will satisfy the equation. However, the solution y should be such that it not only satisfies (9), but also corresponds to positive scale factors (8). Additionally, it should ensure well-separated discriminant functions. Considering these requirements, Formulation II was posed (Airan et al., 2013) as presented next:

Formulation II: Generation of Linear Machine

$$\max_{s,y} \quad s \tag{11a}$$

$$\text{s.t.,} \quad s \leq \beta_l, \quad l = 1, 2, \ldots, n_f \tag{11b}$$

$$Gy = 0 \tag{11c}$$

$$-1 \leq \alpha_j^i \leq 1, \quad i = 1, 2, ..., n_r,\ j = 0, 1, ..., d \tag{11d}$$

$$0 < \beta_l \leq 1, \quad l = 1, 2, ..., n_f \tag{11e}$$

Formulation II is an LP with objective function (to be maximized) being the minimum of all positive scaling factors. A large value of objective function indicates well separated discriminant functions. In Formulation II, without loss of generality, the magnitudes of all the coefficients of the discriminant functions as well as the scale factors have been bounded by one.

Airan et al. (2017) have presented a formal proof of the correctness of Formulation II namely, if Formulation II is feasible and the scaling factors are positive then linear discriminant function will be the greatest for the region to which the point belongs. A key assumption in this proof is that we have correct adjacency identification from step (1) and the proof fails to hold if there is incorrect identification. Airan et al. (2017) also proved that if Formulation II is infeasible, LM does not exist.

2.3 Linear Machine Tree

For a given polyhedral partition of the parametric space, a LM may not always exist. Airan et al. (2017) showed the violation of facet-to-facet properly as one condition under which LM will not exist. The polyhedral partition $\mathcal{P}$ is said to possess the facet-to-facet property if $F^{i,k} = CR^i \cap CR^k$ is a facet of both CR^i and CR^k for all $d-1$ dimensional intersections $F^{i,k}$. Airan et al. (2017) proposed construction of LM tree when the facet-to-facet property is violated. The LM tree is generated by modifying the polyhedral partition by recursively splitting the partition into two halves along the facet involved in maximum number of violations until it reaches a half region which no longer violates the facet-to-facet property. Such partitions when encountered are called child nodes and LMs are developed for them. During online implementation, the child node containing the point is first identified by checking the signs of hyperplanes used to split the original partition. Subsequently, LM developed on the child node is used to find the region to which the point belongs within the node. The identified region in the child node is then mapped to the CR in the original partition to solve the point location problem (Airan et al., 2017).

3. ILLUSTRATIVE SYSTEM

In this work, e-MPC is applied on a quadruple tank setup to illustrate the ideas proposed ahead. The model equations are described in Johansson (2000). The system parameters correspond to an experimental quadruple tank setup available at IIT Bombay (Thosar et al., 2020)

A conventional quadratic cost MPC formulation was converted to a mp-QP formulation using MPT (Herceg et al., 2013) which generated the e-MPC solution. The state and input variables were constrained within ± 10 cm and ± 10 Volts respectively. The cost function has weighting matrices for state, input and terminal state which are specified as $I_{4\times4}$, $I_{2\times2}$, and $0_{4\times4}$ respectively. The prediction and control horizons were taken to be equal and were varied from $3-8$ to generate different e-MPC problems. The following nomenclature will be used to refer to a polyhedral partition generated from the system and parameters described above: $\mathcal{P}$-horizon("horizon number").

It was found that the selection of parameters across the prediction horizons indicated above yielded polyhedral partitions which violated the facet-to-facet property. For such cases LM trees were constructed and LMs generated for the child nodes. The nomenclature to refer to a child node generated from the system and parameters described above is:$\mathcal{P}$-horizon("horizon number")-CN ("child node number"). Hence, if the first child node in the LM tree is generated from the specified mp-QP parameters with horizon length 3, we call it $\mathcal{P}$-horizon3-CN1.

4. NUMERICAL ISSUES

Formulations I and II are both LPs which invoke numerical optimization engines. The LP solutions depend on the optimization solver and the underlying algorithm used. Additionally, tolerances/thresholds on constraint violations used in the numerical procedure can also affect the solution. For instance, strict positivity of objective function is critical for both LPs and is ensured by thresholds. Various thresholds should be tuned to ensure robustness and accuracy. For example, very tightly tuned parameters (low thresholds) might not yield a LM via Formulation II even when one exists. Loosely tuned parameters on the other hand may incorrectly identified adjacent pairs and incorrect linear machine via Formulation II. Here, we investigated the effect of these tolerances and found that the results are sensitive to the chosen values. Using various instances of the e-MPC problem corresponding to different horizon lengths for the quadruple tank problem, (Section 3), the thresholds are tuned and acceptable range of thresholds is presented in Appendix A. However, the obtained ranges are for the specific quadruple tank problem considered here and may not work for other systems/parameter settings.

It is thus desirable to develop numerical accuracy tests or conditions whose satisfaction can confirm the correctness of the LM obtained after solving Formulations I and II with user specified tolerances, and are proposed next.

5. SOME NECESSARY & SUFFICIENT CONDITIONS

In this section we propose some conditions to evaluate accuracy of results of the numerical optimization approaches involved in obtaining LM solution to the point location problem. We first present a necessary condition for correct identification of adjacent pairs of regions.

5.1 A Necessary Condition for Accuracy of Formulation I - Connected Adjacency Graph Test

A correct neighbour identification, would neither leave out any neighbouring pair, nor will it identify two non-neighbours as neighbours. Once Formulation I has been executed for all the critical regions and a list of neighbouring pairs obtained, we propose to construct an undirected graph G(V,E) where V is the set of vertices and E is the set of undirected edges. The vertices correspond to the critical regions in e-MPC. Vertices i and k are joined by an edge if CR^i and CR^k have been identified as neighbours using Formulation I. Then the following can be stated:

Theorem 1. Necessary condition for adjacent region identification: A necessary condition for the correctness of adjacent regions identification procedure is that the graph G(V,E) should be a connected graph.

Proof. An undirected graph G(V,E) is said to be connected if we can reach any vertex from any other vertex by traveling along the edges (Deo, 2016). The proof of the theorem follows from the fact that the critical regions generated in the offline multi-parametric programming step are such that $\cup_{i=1}^{n_r} CR^i = X$ (Spjøtvold et al., 2006). Thus, the CRs or groups of CRs cannot be isolated.

In case the adjacency graph is disconnected, it indicates that at least one pair of adjacent regions has not been identified by Formulation I. This may lead to an incorrect LM despite Formulation II being feasible with a positive objective function since a constraint of the form (7) on the discriminant functions of the missing pair of adjacent regions will be absent in Formulation II.

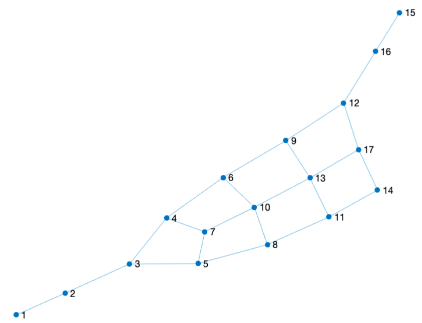

Fig. 1. Adjacency graph for $\mathcal{P}$-horizon5-CN1

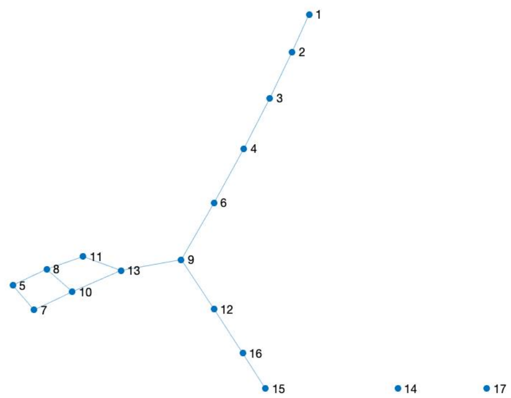

Fig. 2. Adjacency graph for $\mathcal{P}$-horizon5-CN1 with deviation from tolerance Formulation I-II=10^{-7}

Note that the connectedness of the graph is just a necessary condition and not a sufficient condition for the accuracy of neighbour identification step. It is possible that a non-neighbouring pair of regions has been identified as a neighbouring pair by Formulation I due to the choice of thresholds. This cannot be detected from adjacency graph.

Remark 1. Depth first search can be used to check connectedness of an undirected graph (Deo, 2016).

Figures 1 and 2 illustrate two types of adjacency graphs obtained after Formulation I for various instances of the e-MPC problem for the quadruple tank setup (Section 3):

(1) Figure 1 shows a case where the graph is connected i.e. the necessary condition is satisfied. The linear machine obtained for this case resulted in accurate point location. The tolerances used for Formulation I in this case are the same as presented in Appendix A.
(2) Figure 2 shows a case where the graph is disconnected i.e. the necessary condition is violated. Critical regions 14 and 17 at a child node in the LM tree are isolated and have no neighbours and hence no constraints on the discriminant functions of regions 14 and 17 get imposed in Formulation II.

5.2 A Sufficient Condition for Accuracy of Formulation II- Online Verification Test

Once the adjacency oracle is obtained by step I (Section 2.2) of the LM generation approach, Formulation II is solved to obtain the discriminant functions and scaling factors. As discussed in Section 2.2 tolerances for constraints are involved in Formulation II as well which may affect the correctness of the resulting LM. We now propose a sufficient condition whose satisfaction by a given parametric realization will guarantee the correctness of the solution of point location problem given by the LM for that realization. The proposed condition needs to be checked during the online phase and is termed the "Online Verification Test".

For a pair of adjacent regions CR^i and CR^k, we define,

$$\mathcal{G}^{i,k}(x) = -g^i(x) + g^k(x),\ \forall\ (i,k) \in NR \tag{12}$$

where NR is the set of all pairs of adjacent regions identified in Step I of the Linear Machine approach. Thus, $\mathcal{G}^{i,k}(x)$ is the difference of discriminant functions of adjacent critical regions k and i. Further, let

$$P^{i,k}(x) = A^{i,k}x - b^{i,k},\ \forall\ (i,k) \in NR \tag{13}$$

where $A^{i,k}x = b^{i,k}$ is the equation of the hyperplane defining the common facet between critical regions i and k. Let $HS_A^{i,k} = \{x \in X | A^{i,k}x \le b^{i,k}\}$ be the negative half-space of the hyperplane and $HS_B^{i,k} = \{x \in X | A^{i,k}x \ge b^{i,k}\}$ the positive half-space. Without loss of generality, we assume that CR^i and CR^k are in the negative and positive half-spaces of the hyperplane, respectively. We now state the sufficient condition for correctness of LM results for a given parametric realization $x = x^*$.

Theorem 2. Given an accurate adjacency oracle and all scaling factors $\beta^{i,k} > 0$, for any $x^* \in X$, if $\mathcal{G}^{i,k}(x^*)$ and $P^{i,k}(x^*)$ have the same sign for all pairs $(i,k) \in NR$, then the linear machine identification for x^* is accurate.

Proof. The proof is on similar lines as the proof of correctness of LM formulation in Airan et al. (2017). From (7, 12 and 13) we know that theoretically,

$$-g^i(x) + g^k(x) = \beta^{i,k}(A^{i,k}x - b^{i,k}),\ \forall x \in X,\ \forall (i,k) \in NR$$

$$\text{or, } \mathcal{G}^{i,k}(x) = \beta^{i,k}P^{i,k}(x),\ \forall x \in X,\ \forall (i,k) \in NR \tag{14}$$

But since optimization solver solutions have numerical inaccuracies, (14) does not hold exactly and we can write,

$$\mathcal{G}^{i,k}x = \beta^{i,k}P^{i,k}(x) + \epsilon^{i,k}(x),\ \forall x \in X,\ \forall (i,k) \in NR \tag{15}$$

Without loss of generality, assume that the given parametric realization $x^* \in \text{int}(CR^i)$. We need to prove that $g^i(x^*)$ has the largest value amongst all discriminant functions, even if the errors $\epsilon^{i,k}(x^*)$ are non-zero, but assuming that $\mathcal{G}^{i,k}(x^*)$ and $P^{i,k}(x^*)$ have same signs $\forall (i,k) \in NR$. We know $P^{i,k}(x^*) < 0$ since $x^* \in \text{in}(CR^i)$. With $P^{i,k}(x^*)$ and $\mathcal{G}^{i,k}(x^*)$ having the same signs, (14) yields $g^i(x^*) > g^k(x^*)$. This relation can be similarly shown to hold for all regions adjacent to CR^i. Thus, $g^i(x^*)$ is largest amongst all CR^i neighbours. It remains to show that $g^i(x^*)$ is largest amongst non-neighbouring regions as well, i.e.

$$g^i(x^*) > g^k(x^*) \tag{16}$$

where CR^k is not a neighbour of CR^i. Select a point $x^k \in \text{int}(CR^k)$ such that the line joining x^k and x^* intersects only $d-1$ dimensional facets. Let w be the number of regions encountered while traversing from x^* to x^k, with the sequence of regions being $\{CR^{l_1}, CR^{l_2}, \ldots, CR^{l_w}\}$ where $l_1 = i$ and $l_w = k$ are fixed. Let a pair of successive regions $CR^{l_u}, CR^{l_{u+1}}$ in this sequence be adjacent regions with common hyperplane $H^{l_u,l_{u+1}}$ with

$$CR^{l_u} \subseteq HS_A^{l_u,l_{u+1}},\ \text{ and } CR^{l_{u+1}} \subseteq HS_B^{l_u,l_{u+1}}. \tag{17}$$

$$\therefore\ x^* \in HS_A^{l_u,l_{u+1}} \tag{18}$$

Since $H^{l_u,l_{u+1}}(x^*)$ and $\mathcal{G}^{l_u,l_{u+1}}(x^*)$ have same signs,

$$g^{l_u}(x^*) > g^{l_{u+1}}(x^*) \tag{19}$$

It is important to note that we did not use the exact equality (7) which was part of the LM design formulation to obtain (19). We instead used the given statement of same signs for both $H^{l_u,l_{u+1}}(x^*)$ and $\mathcal{G}^{l_u,l_{u+1}}(x^*)$ and the positivity of all scaling factors. Applying (19) to all the CRs encountered while traversing along the line joining x^* to x^k, we obtain the desired result:

$$g^i(x^*) = g_1^l(x^*) > g_2^l(x^*) > \ldots > g^{l_w}(x^*) = g^k(x^*) \tag{20}$$

$$\implies g^i(x^*) > g^k(x^*) \tag{21}$$

It should be noted that this test is a sufficient condition for guaranteeing accuracy of classification of the given parametric realization x^*. Hence, if this test is violated, accuracy of the region identified by the LM is not guaranteed. In such cases, sequential/binary search (Tøndel et al., 2003) may be used to solve the point location problem.

5.3 Demonstration on Quadruple Tank System

We now demonstrate the utility of the online-verification test by performing point location for the quadruple tank system (Section 3). The parameters tabulated in Appendix A (Table A.1) are used on the polyhedral partition $\mathcal{P}$-horizon8. It results in a total of 895 CRs, but as the facet-to-facet property is violated, LM does not exist. The LM tree modifies the polyhedral partition and results in 41 nodes with 21 child nodes out of which LM exists only for 6 child nodes. For the remaining 15 child nodes, sequential search is used to solve the point location problem. Point location is carried out as indicated in Section 2.3 and in Airan et al. (2017). After identifying the region within child node containing the point in contention, the online verification test is used to validate the accuracy of the finding. If the test fails, the LM finding is dropped and a sequential search is used within the child node. The identified region within the child node is mapped to the original partition and renders the location of x^*. We generate $N = 500$ random points in each of the 895 critical regions and their identified location via the LM approach is then compared with the true region. The percentage of misclassified points is used as a metric to check the accuracy of the approach.

Table 1. $\mathcal{P}$-horizon8 LM classification results

# Child node	# Regions	% Misclassification (no online verification test)	% Misclassification (with online verification test)	% False alarm
1	77	29.5	0	65. 9
2	55	5.5	0	14.8
3	7	0	0	0
4	77	7.8	0	47.4
5	56	35.7	0	26.8
6	8	0	0	0

A detailed analysis for each child node of the LM tree associated with $\mathcal{P}$-horizon8 is presented in Table 1. It is seen that for several child nodes, instances misclassified without the online verification test are rectified after online verification test is performed. Whenever the online verification test is not passed, an alternate point location method (sequential search) is employed. The online verification test is a sufficient condition only, and hence there are instances where the online verification test fails and yet the linear machine classification is correct. The percentage of such instances is presented in the last column of Table 1 under the heading "False Alarm".

6. CONCLUSIONS

In this work, we focused on addressing the issue of correctness of LM in presence of numerical errors in the LM generation procedure. These numerical errors are inevitable in any numerical optimization procedure. We proposed a necessary condition (to be checked offline) for checking the correctness of adjacent regions identification. we also proposed a sufficient condition (to be deployed online) for guaranteeing the correctness of the LM result for a given parametric realization. LM based e-MPC solution to the quadruple tank system was investigated to highlight the utility of the proposed conditions. The issue of correctness in presence of numerical errors is a generic issue potentially affecting not only various other point location methods but also the key step of generation of critical regions in the parametric space. There is need to investigate these issues in more detail to enable robust and accurate e-MPC implementations.

ACKNOWLEDGEMENTS

Funding in the form of the APJ Abdul Kalam Memorial International Travel Award to the first author, from the Automatic Control and Dynamic Optimization Society (ACDOS), India is gratefully acknowledged.

REFERENCES

Airan, A., Bhartiya, S., and Bhushan, M. (2013). Linear machine: A novel approach to point location problem. In *IFAC Proceedings Volume 10, International Conference on Dynamics and Control of Process Systems (DYCOPS)*, 445–450. Mumbai.

Airan, A., Bhartiya, S., and Bhushan, M. (2017). Linear machine solution to point location problem. *IEEE Transactions on Automatic Control*, 62(3), 1403–1410.

Baotić, M., Borrelli, F., Bemporad, A., and Morari, M. (2008). Efficient on-line computation of constrained optimal control. *SIAM J. Control Optim.*, 47(5), 2470–2489.

Bayat, F., Johansen, T.A., and Jalali, A.A. (2011). Using hash tables to manage the time-storage complexity in a point location problem: Application to explicit model predictive control. *Automatica*, 47(3), 571–577.

Bemporad, A., Borrelli, F., and Morari, M. (2000). The explicit solution of constrained LP-based receding horizon control. In *Proceedings of the 39th IEEE Conference on Decision and Control*, 632–637.

Deo, N. (2016). *Graph Theory with applications to engineering and computer science.* PHI Learning Private Limited, Delhi.

Duda, R.O., Hart, P.E., and Stork, D.G. (2001). *Pattern Classification.* John Wiley & Sons, New York, NY, 2nd edition.

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

Hrovat, D., Cairano, S.D., Tseng, H.E., and Kolmanovsky, I.V. (2012). The development of model predictive control in automotive industry: A survey. In *IEEE International Conference on Control Applications (CCA)*, 295–302.

Johansson, K.H. (2000). The quadruple-tank process: A multivariable laboratory process with an adjustable zero. *IEEE Trans. Autom. Control*, 8(3), 456–465.

Mönnigmann, M. and Kastsian, M. (2011). Fast explicit MPC with multiway trees. In *IFAC Proceedings Volume 18, World Congress*, 1356–1361. Milano.

Oberdieck, R., Diangelakis, N., Nascu, I., Papathanasiou, M., Sun, M., Avraamidou, S., and Pistikopoulos, E. (2016). On multi-parametric programming and its applications in process systems engineering. *Chemical Engineering Research and Design*, 116, 61–82.

Qin, J. and Badgwell, T. (2003). A survey of industrial model predictive control technology. *Control Engineering practice*, 11, 733–764.

Spjøtvold, J., Kerrigan, E.C., Jones, C.N., Tøndel, P., and Johansen, T.A. (2006). On the facet-to-facet property of solutions to convex parametric quadratic programs. *Automatica*, 42, 2209–2214.

Thosar, D., Mukherjee, T., Gilbile, P., and Bhushan, M. (2020). Conventional and explicit approaches for simultaneous moving horizon estimation and model predictive control: A comparative evaluation. In *6th IFAC Conference on Advances in Control and Optimization of Dynamical Systems (ACODS).* Chennai.

Tøndel, P., Johansen, T.A., and Bemporad, A. (2003). Evaluation of piecewise affine control via binary search tree. *Automatica*, 39(5), 945–950.

Appendix A. FORMULATION I AND II NUMERICAL PARAMETERS

The optimization engine used is IBM ILOG CPLEX Optimization (IBM). An indicative range of permissible constraint tolerances generated by testing on variations of the quadruple tank system (Section 3) listed (Table A.1) and were established by post processing the results obtained from the CPLEX LP implementation.

Table A.1. Constraint tols.: Indicative ranges

Constraint tolerance name	Corresponding constraint	Tolerance range identified
Potential neighbour	Equations to match hyperplanes	$10^{-5} - 10^{-4}$
I-1	(6a) ($t > 0$)	$10^{-9} - 5\times10^{-9}$
I-2	(6c & 6d)	$10^{-6} - 10^{-5}$
II-1	(11a) ($s > 0$)	$10^{-10} - 10^{-8}$
II-2	(11c)	$10^{-9} - 10^{-8}$
II-3	LHS of (11e)	$10^{-10} - 10^{-8}$
II-4	RHS of (11e & 11d)	$10^{-8} - 10^{-5}$

As can be seen in Table A.1, most tolerances have narrow ranges and hence may not be valid for other systems.