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

Explicit model predictive control accuracy analysis

Knyazev, A.; Zhu, P.; Di Cairano, S.

TR2015-149 December 15, 2015

Abstract

Model Predictive Control (MPC) can efficiently control constrained systems in real-time applications. MPC feedback law for a linear system with linear inequality constraints can be explicitly computed off-line, which results in an off-line partition of the state space into non-overlapped convex regions, with affine control laws associated to each region of the partition. An actual implementation of this explicit MPC in low cost micro-controllers requires the data to be "quantized", i.e. represented with a small number of memory bits. An aggressive quantization decreases the number of bits and the controller manufacturing costs, and may increase the speed of the controller, but reduces accuracy of the control input computation. We derive upper bounds for the absolute error in the control depending on the number of quantization bits and system parameters. The bounds can be used to determine how many quantization bits are needed in order to guarantee a specific level of accuracy in the control input.

IEEE Conference on Decision and Control (CDC)

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Explicit model predictive control accuracy analysis

Andrew Knyazev[1], Peizhen Zhu[2] and Stefano Di Cairano[3]

Abstract— **Model Predictive Control (MPC) can efficiently control constrained systems in real-time applications. MPC feedback law for a linear system with linear inequality constraints can be explicitly computed off-line, which results in an off-line partition of the state space into non-overlapped convex regions, with control laws associated to each region of the partition. An actual implementation of this explicit MPC in a target control micro-controller hardware requires the data to be "quantized", i.e. represented with a small number of memory bits. An aggressive quantization decreases the number of bits and the controller manufacturing costs, and may increase the speed of the controller, but reduces accuracy of the control. We derive upper bounds for the absolute error in the control depending on the number of quantization bits and system parameters. The bounds can be used, e.g., to determine how many quantization bits are needed in order to guarantee a specific level of accuracy in the input.**

I. INTRODUCTION

Model predictive control (MPC) [1] is an effective method for control design of multivariable constrained systems in chemical and process control, automotive, aerospace, and factory automation [2]–[4]. Due to the need to solve a constrained optimal control problem in real time, MPC tends to be significantly more computationally expensive than other control methods.

Explicit MPC [5], [6] may reduce the online computational cost and code complexity by pre-computing the MPC feedback law as a state feedback, thus making it viable for fast applications with limited computational capabilities [7]–[10]. In particular, for linear systems subject to linear constraints and cost function based on 1-norm, ∞-norm, or squared 2-norm, the Explicit MPC results in a polyhedral piecewise affine (PWA) feedback law. Thus, during the online execution, the Explicit MPC controller first identifies which polyhedral region contains the current state, and then computes the control action by evaluating the corresponding affine control law. The identification of the polyhedral region is referred to as the point location problem [11], which can be solved by sequential search and binary search tree see, e.g., [12], [13]. Due to the exponential increase of the number of regions with respect to the number of constraints in the MPC problems, in recent years several techniques for reducing complexity of the explicit MPC feedback law while maintaining its most important properties have been proposed, see, e.g., [14]–[16] and references therein.

[1]Mitsubishi Electric Research Laboratories; 201 Broadway Cambridge, MA 02139 knyazev@merl.com

[2]Mitsubishi Electric Research Laboratories; 201 Broadway Cambridge, MA 02139 pzhu@merl.com

[3]Mitsubishi Electric Research Laboratories; 201 Broadway Cambridge, MA 02139 dicairano@merl.com

In practice, the data of explicit MPC have to be typically stored in a micro-controller hardware memory, so that every stored number is represented by a small fixed number of bits for every number in the data. In other words, data cannot be stored exactly and hence a precision loss occurs. We call this reduction of precision "quantization" and the reduced precision data "quantized" data. The method for quantization can be as simple as rounding. Aggressive quantization has the advantage of decreasing memory requirements and increasing the speed of the control input evaluation, at the price of introducing inaccuracy in the computation of the control input. If the quantization precision is too small, the controller can fail to accurately determine the region for the current state of the controlled system, and thus, the control. For example, by quantizing the region data some regions may disappear, or holes between regions, where the control input cannot be determined, may appear. In addition, by quantizing the state measurement/estimate data the quantized state may jump to a different region.

The effect of quantization has been investigated for implicit MPC for instance in [17], [18], but not been fully addressed for Explicit MPC. In this paper, we investigate the resulting accuracy in the control input computation in Explicit MPC as a consequence of different quantization levels, so that we can determine how many bits needs to be used to guarantee a desired level of accuracy in the control input computation.

In Section II, we provide an accuracy analysis divided into different cases, depending on mutual positions of the exact and the quantized system states. A most likely and easy to analyze case is where the quantization does not affect the system state region, so that the same feedback law applies to both the exact and the quantized system states. A difficult case for analysis, leading to a much larger possible controller inaccuracy, is where the quantization makes the system state to jump over a region face to a different region. Here, bounding the accuracy of the control requires taking into account not only quantization precision for the system state, but also quantization effects of the region faces and of the feedback laws in different regions. Upper bounds on the accuracy of the control input computation in Section II use no knowledge of the quantized data, describing the worst case scenario, for that reason called "a-priori" bounds. In Section III, we show how such bounds can be improved by exploiting a rescaling technique that makes the system state space evenly sized in all spacial directions.

After some quantized implementation of the controller is determined, the a-priori bounds of Section II are improved in Section III, using the already known off-line quantized data,

in addition to the original data. Resulting tighter a-posteriori bounds depend primarily on the quantization precision of the current state.

All bounds mentioned above are deterministic, guaranteed under our assumptions. The a-priori bounds for the case, where the state region changes as a result of the quantization, are the most pessimistic, but in this case the system state needs to be in a proximity to a face of the region. If the system state is randomly uniformly distributed over the state space, such a situation has a small probability to appear. We use this and similar observations to derive in Section III probabilistic estimates of the control error bounds, which improve our deterministic bounds, but only hold with certain probability.

We validate the bounds using several numerical tests in Section IV. Finally, Section V summarizes our conclusions.

Preliminary: throughout this paper, $\mathbb{R}$ denotes real number. $\|\cdot\|_1$, $\|\cdot\|_2$, and $\|\cdot\|_\infty$ denote 1-norm, 2-norm, and ∞-norm, respectively. $A^{'}$ denotes the transpose of A. The explicit MPC control law has the form

$$u_i(x) = F_i x + G_i \quad \forall x \in P_i,\, i = 1, \ldots, n_r,$$

where $F_i \in \mathbb{R}^{m\times n}$, $G_i \in \mathbb{R}^m$, $P_i = \{x \in \mathbb{R}^n | H_i x \leq K_i$, $H_i \in \mathbb{R}^{n_c^i \times n}, K_i \in \mathbb{R}^{n_c^i}\}$, n_r denotes the number of regions, and $H_i = \left[H_i^{1^{'}}, \ldots, H_i^{{n_c}^{i^{'}}}\right]^{'}$ and $K_i = [K_i^1, \ldots, K_i^{{n_c}^i}]^{'}$. We denote $u(x)$ such as

$$u(x) = \begin{cases} u_1(x) \\ \vdots \\ u_{n_r}(x) \end{cases}$$

and $u(x)$ is a continuous PWA function. PWA controller determines which region contains a given state x. If the state x is in the region, then the corresponding affine control law to compute the control input is used.

We use $hx = k$ to denote a general hyperplane, where $h \in \mathbb{R}^{1\times n}$, $k \in \mathbb{R}$ and $x \in \mathbb{R}^n$. The vector h' (the transpose of h) can be thought of as a vector normal (orthogonal) to the hyperplane $hx = k$ at each point. The hyperplane dose not change, if h and k are multiplied by the same nonzero scalar.

Definition 1.1 (face): [19] Linear inequality $hx \leq k$ is called valid for a polyhedron P if $hx \leq k$ holds for all $x \in P$. A nonempty subset of a polyhedron is called a face of P if it is represented as $P \cap \{x \in \mathbb{R}^n | \, hx = k\}$, for some valid inequality $hx \leq k$. The faces of polyhedron P of dimension $0, 1, (n-2)$ and $(n-1)$ are called vertices, edges, ridges and facets, respectively.

Definition 1.2 (distance): A distance between a point x_0 and a hyperplane $hx = k$ is defined as

$$\mathrm{dist}(x_0, hx = k) = \frac{|hx_0 - k|}{\|h\|_2}.$$

We assume that all data are quantized when storing the data in the control hardware. Let the quantization function be $f(z)$, such that $\hat{z} = f(z) = z + \Delta z$, where $\|\Delta z\|_\infty \leq \epsilon$ and $0 \leq \epsilon \leq 1$. Given a state x, we have $\hat{x} = x + \Delta x$. Similarly, we have $\hat{u}_i(\hat{x}) = \hat{F}_i\hat{x} + \hat{G}_i$, where $\hat{H}_i\hat{x} \leq \hat{K}_i$ for $i = 1, \ldots, n_r$. In general, we use the symbol "hat" to denote data after quantization.

II. Accuracy analysis

In this section, we focus on accuracy analysis of the control input. Since the control input depends on the location of the state, to analyze the accuracy of the control input we have to analyze how the state changes and which region it falls into before and after quantization. For every state x, after quantization the state $\hat{x}$ may fall into one of regions or not. In the following, we analyze the situation case by case.

Case 1: suppose that $\hat{x}$ is in the region $\hat{P}_i$ and x is in the region P_i. In this case, the state x is in the "same" region as the state $\hat{x}$ after quantization. Here, the "same" region means that the quantized region $\hat{P}_i$ and the exact region P_i have the same index i in our database of regions. Therefore, the accuracy of the control input can be measured by the maximum absolute changes between $u_i(x)$ and $\hat{u}_i(\hat{x})$, i.e., $\|\hat{u}_i(\hat{x}) - u_i(x)\|_\infty$.

Case 2: suppose that $\hat{x}$ is in the region $\hat{P}_i$ and x is out of any region P_i for $i = 1, \ldots, n_r$. We see that the state x is not in any original region. But, after quantization the state $\hat{x}$ falls into one of regions. This may happen when the state x is in the neighborhood of the outside of boundary of the union of the regions, which is convex. In this case, the control law corresponding to the state x does not exist. After quantization, the control law corresponding to the state $\hat{x}$ is $\hat{u}_i(\hat{x})$.

Case 3: suppose that $\hat{x}$ is out of any region $\hat{P}_i$ and x is out of any region P_i for $i = 1, \ldots, n_r$. In this case, the state x and the state $\hat{x}$ are out of all regions. Consequently, the control laws corresponding to x and $\hat{x}$ do not exist.

Case 4: suppose that $\hat{x}$ is out of any region $\hat{P}_i$ for $i = 1, \ldots, n_r$, but x is in one of regions, e.g., region P_j. In this case, the state x is in one of regions, but after quantization the state $\hat{x}$ does not fall into any region. It is possible that the state $\hat{x}$ may fall into some holes which mean some holes between regions after quantization. In this case, the control law corresponding to $\hat{x}$ does not exist

Case 5: suppose that $\hat{x}$ is in the region $\hat{P}_i$ and x is in the region P_j, where $i \neq j$, which means the state x is in one of regions, but after quantization the state $\hat{x}$ falls into another region. In addition, the two original regions P_i and P_j may intersect each other or may not, see Figure 1 for one dimension. Under this situation, the accuracy of the control input can be measured by $\|\hat{u}_i(\hat{x}) - u_j(x)\|_\infty$.

From analysis mentioned above, at least one of the control laws corresponding to the states x and $\hat{x}$ does not exist in cases 2, 3 and 4. In the rest of the paper, we are interested to further investigate the accuracy of the control input when both the states x and $\hat{x}$ exist.

As we know, the control input depends which region the state falls into. The region is consisted by hyperplanes. To obtain the bounds of the accuracy of the control input, we

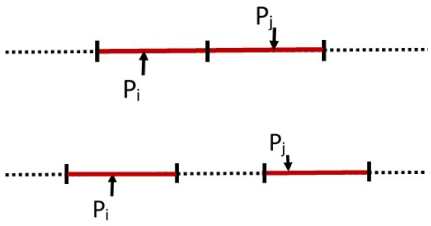

Fig. 1. Left: two neighboring regions; Right: two regions with no intersection (1D).

start with the accuracy of the hyperplane when data are quantized.

Lemma 2.1: Let $hx \leq k$ be a half-space. Let $y = hx - k$ and $\hat{y} = (h + \Delta h)(x + \Delta x) - (k + \Delta k)$ with $\|\Delta h\|_\infty \leq \epsilon$, $\|\Delta x\|_\infty \leq \epsilon$, and $|\Delta k| \leq \epsilon$ for some $\epsilon \geq 0$. We have

$$|\hat{y} - y| \quad \leq \quad \epsilon(\|h\|_1 + \|x\|_1 + n\epsilon + 1). \tag{1}$$

Proof: From direct calculation, we have

$$\begin{aligned} & |\hat{y} - y| \\ = \; & |(h + \Delta h)(x + \Delta x) - (k + \Delta k) - (hx - k)| \\ = \; & |h\Delta x + \Delta h x + \Delta h \Delta x - \Delta k| \\ \leq \; & |h\Delta x| + |\Delta h x| + |\Delta h \Delta x| + |\Delta k| \\ \leq \; & \|h\|_1 \|\Delta x\|_\infty + \|\Delta h\|_\infty \|x\|_1 + \|\Delta h\|_\infty \|\Delta x\|_1 + |\Delta k| \\ \leq \; & \epsilon\|h\|_1 + \epsilon\|x\|_1 + n\epsilon^2 + \epsilon \\ = \; & \epsilon(\|h\|_1 + \|x\|_1 + n\epsilon + 1). \end{aligned}$$

■

Lemma 2.2: Let hyperplane $hx = k$ separate two neighboring regions P_i and P_j, such that $hx \leq k$ for $x \in P_j$ and $hx \geq k$ for $x \in P_i$. Let the state x be in the region P_j. After quantization, $\hat{x}$ falls into the region $\hat{P}_i$. We have

$$-\delta < y \leq 0 \leq \hat{y} < \delta,$$

where δ, y and $\hat{y}$ use the same notation as in Lemma 2.1.

Proof: According to Lemma 2.1, we have $|\hat{y} - y| \leq \delta$, which implies

$$-\delta + y \leq \hat{y} \leq \delta + y.$$

After quantization, $\hat{x}$ falls into the region $\hat{P}_i$ which means $\hat{x}$ is out of region $\hat{P}_j$. So, $\delta + y > 0$. Therefore, we have

$$-\delta < y \leq 0.$$

Similarly, we can obtain that $0 \leq \hat{y} < \delta$. ■

Corollary 2.3: Let the state x be satisfied $hx \leq k$. After quantization, we have $\hat{h}\hat{x} \geq \hat{k}$. Then,

$$\text{dist}(x, hx = k) \leq \frac{\delta}{\|h\|_2},$$

where $\delta = \epsilon(\|h\|_1 + \|x\|_1 + n\epsilon + 1)$.

Proof: From Definition 1.2 and Lemma 2.2, we can obtain this result directly. ■

Corollary 2.3 illustrates that after quantization the state jumps to the other side of a hyperplane which happens if distance from the state to the hyperplane is less than a constant.

By the auxiliary lemmas and corollary presented above, we are already to provide our main result.

Theorem 2.4: Let a hyperplane $hx = k$ separate two neighboring regions P_i and P_j, such that for $x \in P_j$, we have $hx \leq k$ and for $x \in P_i$, we have $hx \geq k$. Moreover, P_i and P_j share a common facet $\mathcal{Z}$, where $\mathcal{Z} \subset \{x : hx = k\}$. Let the state x be in the region P_j and the orthogonal projection of x on the hyperplane $hx = k$ be in $\mathcal{Z}$. After quantization, $\hat{x}$ falls into the region $\hat{P}_i$. We have

$$\begin{aligned} \|\hat{u}(\hat{x}) - u(x)\|_\infty \quad \leq \quad & \frac{\delta}{\|h\|_2^2}\|(F_i - F_j)h'\|_\infty + \\ & \epsilon(\|F_i\|_\infty + n\|x\|_\infty + n\epsilon + 1), \end{aligned} \tag{2}$$

where $\delta = \epsilon(\|h\|_1 + \|x\|_1 + n\epsilon + 1)$.

Proof: Since x is in the region P_j, but after quantization $\hat{x}$ falls into the region $\hat{P}_i$. According to Lemma 2.1 and 2.2, we get $-\delta < hx - k \leq 0$, where $\delta = \epsilon(\|h\|_1 + \|x\|_1 + n\epsilon + 1)$. To get the accuracy for $u(x)$, based on Minkowski inequality we have

$$\begin{aligned} & \|\hat{u}(\hat{x}) - u(x)\|_\infty \\ = \; & \|\hat{u}_i(\hat{x}) - u_j(x)\|_\infty \\ = \; & \|\hat{F}_i\hat{x} + \hat{G}_i - F_j x - G_j\|_\infty \\ \leq \; & \|F_i x + G_i - F_j x - G_j\|_\infty + \\ & \|F_i\Delta x + \Delta F_i x + \Delta F_i \Delta x + \Delta G_i\|_\infty. \end{aligned}$$

From the above inequality, in order to get the upper bound of difference of $u(x)$, we have to get the upper bound of $\|F_i x + G_i - F_j x - G_j\|_\infty$, when x satisfies $-\delta < hx - k \leq 0$. Let the orthogonal projection of x on the hyperplane $hx = k$ be x_p, where x_p is in $\mathcal{Z}$. We can write

$$x_p = x + th',$$

where h' is the transpose of h and t is a real scalar. We have

$$\begin{aligned} k \quad = \quad & hx_p \\ = \quad & hx + thh' \\ = \quad & hx + t\|h\|_2^2. \end{aligned}$$

Therefore, $|t| = |hx - k|/\|h\|_2^2$. Since $u(x)$ is a linear continuous affine function, we have $F_i x_p + G_i = F_j x_p + G_j$. It follows $F_i(x + th') + G_i = F_j(x + th') + G_j$. Hence,

$$\begin{aligned} \|F_i x + G_i - F_j x - G_j\|_\infty \quad = \quad & \|t(F_i - F_j)h'\|_\infty \\ = \quad & |t|\|(F_i - F_j)h'\|_\infty \\ = \quad & \frac{|hx - k|}{\|h\|_2^2}\|(F_i - F_j)h'\|_\infty \\ \leq \quad & \frac{\delta}{\|h\|_2^2}\|(F_i - F_j)h'\|_\infty. \end{aligned} \tag{3}$$

Moreover,

$$\begin{aligned} & \|F_i\Delta x + \Delta F_i x + \Delta F_i \Delta x + \Delta G_i\|_\infty \\ \leq\ & \|F_i\Delta x\|_\infty + \|\Delta F_i x\|_\infty + \|\Delta F_i \Delta x\|_\infty + \|\Delta G_i\|_\infty \\ \leq\ & \|F_i\|_\infty \|\Delta x\|_\infty + \|\Delta F_i\|_\infty \|x\|_\infty \\ & + \|\Delta F_i\|_\infty \|\Delta x\|_\infty + \|\Delta G_i\|_\infty \\ \leq\ & \epsilon(\|F_i\|_\infty + n\|x\|_\infty + n\epsilon + 1). \end{aligned} \tag{4}$$

Combining inequalities (3) and (4), we can obtain the result as expected. ■

In Figure 2, the state x in the red area of the region P_j satisfies the assumption in Theorem 2.4, such that the orthogonal projection of x on the hyperplane $hx = k$ is in $\mathcal{Z}$.

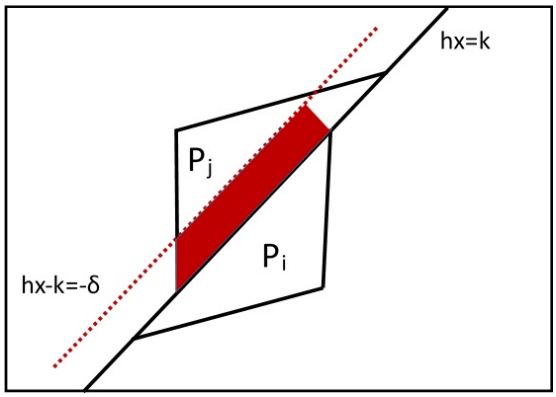

Fig. 2. The state x in the red area of the region P_j satisfies that the orthogonal projection of x on the hyperplane $hx = k$ is in $\mathcal{Z}$.

Remark 2.5: The bound of accuracy of the control input in (2) is called **a priori bound**, since the data after quantization are not explicitly known, and a priori information about data before quantization needs to be used.

Remark 2.6: Determination of neighboring regions is a non-trivial problem, since regions share a common hyperplane may not necessarily share a common facet, see [20]. To find the common facet, we first find the common hyperplane $hx = k$ between two regions P_i and P_j, then to decide whether the hyperplane $hx = k$ is a common facet of regions P_i and P_j.

Remark 2.7: If the state x is in the region P_j and after quantization $\hat{x}$ is still in region P_j, from Theorem 2.4 the first term in (2) disappears. Therefore, $\|\hat{u}(\hat{x}) - u(x)\|_\infty \leq \epsilon(\|F_j\|_\infty + n\|x\|_\infty + n\epsilon + 1)$.

Remark 2.8: We mention that to get the bound of the absolute error in $u(x)$, i.e., $\max \|\hat{u}(\hat{x}) - u(x)\|_\infty$, equivalently, it is to solve

$$\begin{aligned} \max_x\ & \|\hat{F}_i\hat{x} + \hat{G}_i - F_j x - G_j\|_\infty \\ s.t.\quad & H_j x \leq K_j \\ & \hat{H}_i\hat{x} \leq \hat{K}_i \\ & \hat{x} = f(x). \end{aligned}$$

In particular, for the case $m = 1$ and $n \leq 2$ if we relax the condition for the state x whose orthogonal protection on the hyperplane $hx = k$ is not necessary in $\mathcal{Z}$, we can get the different bounds for the absolute error in the control input. See the following lemma and see appendix for the proof.

Lemma 2.9: Let $m = 1$ and $n \leq 2$. Let a hyperplane $hx = k$ separate two neighboring regions P_i and P_j, such that for $x \in P_j$, we have $hx \leq k$ and for $x \in P_i$, we have $hx \geq k$. Moreover, P_i and P_j share a common facet $\mathcal{Z}$, where $\mathcal{Z} \subset \{x : hx = k\}$. Let the state x be in the region P_j. After quantization, $\hat{x}$ falls into the region $\hat{P}_i$. We have

$$\begin{aligned} |\hat{u}(\hat{x}) - u(x)| \ &\leq\ \frac{\delta}{|k|}|G_i - G_j| \\ &+\ \epsilon(\|F_i\|_\infty + n\|x\|_\infty + n\epsilon + 1), \end{aligned}$$

where $\delta = \epsilon(\|h\|_1 + \|x\|_1 + n\epsilon + 1)$.

Remark 2.10: Comparing these two bounds presented in Lemma 2.9 and in Theorem 2.4 for the case $m = 1$ and $n \leq 2$, the only difference are the first term in bounds, i.e., $\delta|G_i - G_j|/|k|$ and $\delta\|(F_i - F_j)h'\|_\infty/\|h\|_2^2$. Furthermore, if $n = 1$, then these two bounds are equivalent.

III. Extension of error bound estimation

From the above analysis, we can see that the a priori bound for the accuracy of the control input depends on ϵ, δ, and the dimension of regions. When the dimension of regions n gets very large, the bound estimation may be pessimistic. In this section, we update the a priori bound by rescaling data, improve the bound by using quantization errors, and refine it by impacting of the probabilistic relaxation.

A. Rescale

The accuracy of $u(x)$ depends on how small ϵ and δ are. To make ϵ and δ small, one approach is to rescale the regions, h, and k, such that

$$\frac{hD^{-1}Dx}{\max(\|hD^{-1}\|_1, |k|)} \leq \frac{k}{\max(\|hD^{-1}\|_1, |k|)},$$

where D is a diagonal matrix with $\|Dx\|_\infty \leq 1$. Let Dx, $hD^{-1}/\max\left(\|hD^{-1}\|_1, |k|\right)$, and $k/\max\left(\|hD^{-1}\|_1, |k|\right)$ be our new x, h, and k, respectively. We have $\|x\|_\infty \leq 1$, $|hx| \leq 1$, and $|k| \leq 1$. Our new δ is less than $\epsilon(n + 2 + n\epsilon)$. The control law has the form

$$u_i(x) = F_i D^{-1}x + G_i \quad \forall x \in P_i,\ i = 1, \ldots, n_r.$$

Consequently,

$$\begin{aligned} \|\hat{u}(\hat{x}) - u(x)\|_\infty \leq\ & \frac{\delta}{\|h\|_2^2}\|(F_i - F_j)D^{-1}h'\|_\infty + \\ & (\epsilon\|F_i D^{-1}\|_\infty + n\epsilon_1\|x\|_\infty + n\epsilon\epsilon_1 + \epsilon_1), \end{aligned}$$

where ϵ is taken after rescaling for regions and ϵ_1 is taken after rescaling for the input control.

Remark 3.1: In particular, let $m = 1$ and $n \leq 2$. Combining Lemma 2.9, Theorem 2.4, and the rescaling skill stated above, we obtain

$$\begin{aligned} |\hat{u}(\hat{x}) - u(x)| \leq\ & \min\left\{\frac{\delta}{\|h\|_2^2}|(F_i - F_j)D^{-1}h'|, \frac{\delta}{|k|}|G_i - G_j|\right\} \\ & + (\epsilon\|F_i D^{-1}\|_\infty + n\epsilon_1\|x\|_\infty + n\epsilon\epsilon_1 + \epsilon_1). \end{aligned}$$

B. A posteriori bound

In practice, we use a fixed point number format which has a specific number of bits reserved for the integer part and a specific number of bits reserved for the fractional part. We use MATLAB function fi with a-bit total word length, 1-bit for sign and b-bit fraction length. Given a state x, we have

$$\hat{x} = \text{fi}(x, 1, a, b) = x + \Delta x,$$

where $\|\Delta x\|_\infty \leq \epsilon = 2^{-b}$. Similarly, we have $\hat{u}_i(\hat{x}) = \hat{F}_i\hat{x} + \hat{G}_i$, where $\hat{H}_i\hat{x} \leq \hat{K}_i$ for $i = 1, \ldots, n_r$. The quantization errors ΔH_i, ΔK_i, ΔF_i, and ΔG_i are known, given the number of bits for the fractional part. Every component in ΔH_i, ΔK_i, ΔF_i, and ΔG_i is bounded by 2^{-b}.

Since the quantization errors are known, all results are presented in Section II and Subsection III-A can be updated. Under assumptions of Theorem 2.4, we have

$$\begin{aligned} \|\hat{u}(\hat{x}) - u(x)\|_\infty &\leq \frac{\delta}{\|h\|_2^2}\|(F_i - F_j)h'\|_\infty \\ &+ \|\Delta F_i x + \Delta G_i\|_\infty + \|\hat{F}_i\|_\infty\epsilon, \end{aligned} \tag{5}$$

where $\delta = |\Delta h x - \Delta k| + \|\hat{h}\|_1\epsilon$, since the term $\|F_i\Delta x + \Delta F_i x + \Delta F_i\Delta x + \Delta G_i\|_\infty$ in the proof of Theorem 2.4 can be written as

$$\begin{aligned} & \|F_i\Delta x + \Delta F_i x + \Delta F_i\Delta x + \Delta G_i\|_\infty \\ = \; & \|(\Delta F_i x + \Delta G_i) + (F_i\Delta x + \Delta F_i\Delta x)\|_\infty \\ \leq \; & \|\Delta F_i x + \Delta G_i\|_\infty + \|\hat{F}_i\|_\infty\|\Delta x\|_\infty \\ \leq \; & \|\Delta F_i x + \Delta G_i\|_\infty + \|\hat{F}_i\|_\infty\epsilon. \end{aligned}$$

Since the bound in (5) is based on the data obtained before and after quantization, therefore it is called **a posteriori bound**.

C. Probabilistic bound

However, the bounds may be large in some cases, since it is unlike to take all states x in regions to reach the upper a posterior bound. In this subsection, we present the new probabilistic bounds for the case $m = 1$.

From the proof of Lemma 2.1 and Theorem 2.4, we can write

$$\begin{aligned} |\hat{u}(\hat{x}) - u(x)| \leq Z = \frac{|(F_i - F_j)h'|}{\|h\|_2^2}(|\Delta h x - \Delta k| + |\hat{h}\Delta x|) \\ +(|\Delta F_i x + \Delta G_i| + |\hat{F}_i\Delta x|), \end{aligned}$$

where Δx can be write as $2^{-b}[\epsilon_1, \ldots, \epsilon_n]'$ with $\epsilon_i \in [-1, 1]$ for $i = 1, \ldots, n$. Moreover, we can rewrite

$$Z = c_0 + \left|\sum_{i=1}^n c_{1i}\epsilon_i\right| + \left|\sum_{i=1}^n c_{2i}\epsilon_i\right|,$$

where $c_0 = \frac{|(F_i - F_j)h'|}{\|h\|_2^2}|\Delta h x - \Delta k| + |\Delta F_i x + \Delta G_i|$, $\hat{h} = [c_{11}, \ldots, c_{1n}]$, and $\hat{F}_i = [c_{21}, \ldots, c_{2n}]$. $\epsilon_1, \ldots, \epsilon_n$ are independently distributed. Let ϵ_i be distributed normally, such as $\epsilon_i \sim \mathcal{N}(0, 1)$ for $i = 1, \ldots, n$. Let $X_1 = \sum_{i=1}^n c_{1i}\epsilon_i$ and $X_2 = \sum_{i=1}^n c_{1i}\epsilon_i$. Then,

$$X_1 \sim \mathcal{N}(0, \sigma_1^2) \quad \text{and} \quad X_2 \sim \mathcal{N}(0, \sigma_2^2),$$

where σ_1^2 and σ_2^2 are variance of X_1 and X_2, respectively. Therefore, $Y_1 = |X_1|$ and $Y_2 = |X_2|$ follow half-normal distributions with probability density functions

$$f_{Y_1}(y_1; \sigma_1) = \frac{\sqrt{2}}{\sigma_1\sqrt{\pi}}\exp\left(-\frac{y_1^2}{2\sigma_1^2}\right)$$

and

$$f_{Y_2}(y_2; \sigma_2) = \frac{\sqrt{2}}{\sigma_2\sqrt{\pi}}\exp\left(-\frac{y_2^2}{2\sigma_2^2}\right),$$

respectively. Since Y_1 and Y_2 are independent, the probability density function of Z can be obtained by

$$f_Z(z) = \int_0^{+\infty} f_{Y_1}(z - y_2 - c_0)f_{Y_2}(y_2)dy_2,$$

where $z \in [c_0, +\infty)$. Furthermore,

$$\begin{aligned} f_Z(z) = \frac{\sqrt{2}}{\sqrt{\pi(\sigma_1^2 + \sigma_2^2)}}\exp\left(-\frac{(z - c_0)^2}{2(\sigma_1^2 + \sigma_2^2)}\right) \\ \left[\text{erf}\left(\frac{\sigma_1(z - c_0)}{\sigma_2\sqrt{2(\sigma_1^2 + \sigma_2^2)}}\right)\right. \\ \left.+\text{erf}\left(\frac{\sigma_2(z - c_0)}{\sigma_1\sqrt{2(\sigma_1^2 + \sigma_2^2)}}\right)\right], \end{aligned}$$

where $\text{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt$. From the probability distribution, we develop a refinement of our a posteriori bound with a specified high probability p. We denote the new bounds Z_p and define it by

$$P(Z \leq Z_p) \leq p.$$

To calculate Z_p for a given p, we use the inverse cumulative density function of Z.

IV. Test Results

In this section, we present several tests regarding to the accuracy of $u(x)$. The system to be controlled is a double integrator

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k), \end{aligned}$$

where

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \text{and } D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Moreover, the state x satisfies $\begin{bmatrix} -15 \\ -15 \end{bmatrix} \leq x \leq \begin{bmatrix} 15 \\ 15 \end{bmatrix}$ and the input $u(k)$ satisfies the constraints, such that $-1 \leq u(k) \leq 1$.

In our first test, we take a quantization function as $f(z) = z + \Delta z$, where Δz is a random vector and satisfying $\|\Delta z\|_\infty \leq 0.01$. Let the state

$$x = \begin{bmatrix} -1.872300978007571 \\ -2.005527198961941 \end{bmatrix}$$

and after quantization

$$\hat{x} = \begin{bmatrix} -1.8624195 \\ -2.0045545 \end{bmatrix}.$$

The state x is in the region P_9 and after quantization $\hat{x}$ jumps into the region $\hat{P}_6$. For the detail, let us see the data sets for regions in the following.

$$H_9 = \begin{bmatrix} -0.090909090909091 & -0.181818181818182 \\ -0.051779109470760 & -0.352859834980392 \\ 0.110805433262676 & 0.533053615854606 \\ 0.074084535025187 & 0.430780401254739 \\ 1 & 1 \\ -1 & -1 \\ 1 & 0 \end{bmatrix}$$

$$K_9 = \begin{bmatrix} 1.000001 \\ 1.000001 \\ -0.999999 \\ -0.999999 \\ 10.000000999999999 \\ 10.000000999999999 \\ 15 \end{bmatrix}$$

$$H_6 = \begin{bmatrix} -0.074084535025187 & -0.430780401254739 \\ 0.152799695981699 & 0.650013942240116 \\ 0.115162535317429 & 0.554474699453481 \\ -1 & -1 \end{bmatrix}$$

$$K_6 = \begin{bmatrix} 1.000001 \\ -0.999999 \\ -0.999999 \\ 10.000000999999999 \end{bmatrix}.$$

The corresponding input controls for both regions are

$$u_9(x) = \begin{bmatrix} -0.277555756156289 \times 10^{-16} \\ 0 \end{bmatrix}' x + 1$$

$$u_6(x) = \begin{bmatrix} -0.027755575615629 \times 10^{-15} \\ -0.111022302462516 \times 10^{-15} \end{bmatrix}' x + 1.$$

After quantization,

$$\left[\hat{H}_9\, \hat{K}_9\right] = \begin{bmatrix} -0.09375 & -0.1796875 & 1 \\ -0.0546875 & -0.3515625 & 1 \\ 0.109375 & 0.53125 & -1 \\ 0.0703125 & 0.4296875 & -1 \\ 1 & 1 & 10 \\ -1 & -1 & 10 \\ 1 & 0 & 15 \end{bmatrix}$$

$$\left[\hat{H}_6\, \hat{K}_6\right] = \begin{bmatrix} -0.0703125 & -0.4296875 & 1 \\ 0.15625 & 0.6484375 & -1 \\ 0.1171875 & 0.5546875 & -1 \\ -1 & -1 & 10 \end{bmatrix}.$$

The corresponding input control for region $\hat{P}_6$ after quantization is

$$\hat{u}_6(x) = \begin{bmatrix} -0.0065625 \\ -0.00777875 \end{bmatrix}' \hat{x} + 1.01.$$

From the above data sets, we can see that the common hyperplane between regions P_9 and P_6 is $hx = k$. where $h = \begin{bmatrix} 0.074084535025187 & 0.430780401254739 \end{bmatrix}$ and $k = -0.999999$. We can obtain $\delta = 0.054026931132494$. By calculation, our bound for the absolute error of $u(x)$ in Theorem 2.4 is

$$\begin{aligned} &\frac{\delta}{\|h\|_2^2}\|(F_i - F_j)h'\|_\infty + \epsilon(\|F_i\|_\infty + 2\|x\|_\infty + 2\epsilon + 1) \\ &= 1.352402835933399 \times 10^{-17} + 0.050310543979239 \\ &= 0.050310543979239. \end{aligned}$$

The absolute error of $u(x)$ from the real computation by sequential search is $|u(x) - \hat{u}(x)| = 0.037815056285625$.

From this example, we can see that the bound is sharp. From other hand, we also can see that the term $\|x\|_\infty = 2.005527198961941$ is a dominant term in our bound. Rescaling regions may help to get a bound closer to the absolute error of $u(x)$ from the real computation. We use the rescale technique as Subsection III-A described. We take $D = \mathrm{diag}(1/15, 1/15)$. Our new x is

$$x = \begin{bmatrix} -0.124820065200505 \\ -0.133701813264129 \end{bmatrix}$$

and our new H, K and $u(x)$ are

$$H_9 = \begin{bmatrix} -0.333333333333333 & -0.666666666666667 \\ -0.127963732064873 & -0.872036267935127 \\ 0.172095792416971 & 0.827904207583029 \\ 0.146741295941595 & 0.853258704058405 \\ 0.5 & 0.5 \\ -0.5 & -0.5 \\ 1 & 0 \end{bmatrix}$$

$$K_9 = \begin{bmatrix} 0.244444688888889 \\ 0.164756097374066 \\ -0.103542227280021 \\ -0.132048386032173 \\ 0.333333366666667 \\ 0.333333366666667 \\ 1 \end{bmatrix}$$

$$H_6 = \begin{bmatrix} -0.146741295941595 & -0.853258704058405 \\ 0.190330219501678 & 0.809669780498322 \\ 0.171977496676730 & 0.828022503323270 \\ -0.5 & -0.5 \end{bmatrix}$$

$$K_6 = \begin{bmatrix} 0.132048650129209 \\ -0.083041190166703 \\ -0.099556291882137 \\ 0.333333366666667 \end{bmatrix}$$

$$u_9(x) = \begin{bmatrix} -0.416333634234434 \times 10^{-15} \\ 0 \end{bmatrix}' x + 1$$

$$u_6(x) = \begin{bmatrix} 0.041633363423443 \times 10^{-15} \\ -0.166533453693773 \times 10^{-15} \end{bmatrix}' x + 1.$$

We still use the quantization function $f(z) = z + \Delta z$, where Δz is a random vector with $\|\Delta z\|_\infty \leq 0.01$. Let

$\hat{x} = \begin{bmatrix} -0.1241613 \\ -0.133637 \end{bmatrix}$. After quantization, data for regions and the input control are as follows:

$$\left[\hat{H}_9\ \hat{K}_9\right] = \begin{bmatrix} -0.3359375 & -0.6640625 & 0.2421875 \\ -0.125 & -0.875 & 0.1640625 \\ 0.171875 & 0.828125 & -0.1015625 \\ 0.1484375 & 0.8515625 & -0.1328125 \\ 0.5 & 0.5 & 0.3359375 \\ -0.5 & -0.5 & 0.3359375 \\ 1 & 0 & 1 \end{bmatrix}$$

$$\left[\hat{H}_6\ \hat{K}_6\right] = \begin{bmatrix} -0.1484375 & -0.8515625 & 0.1328125 \\ 0.1875 & 0.8125 & -0.0859375 \\ 0.171875 & 0.828125 & -0.1015625 \\ -0.5 & -0.5 & 0.3359375 \end{bmatrix}.$$

The corresponding input control for region $\hat{P}_6$ after quantization is

$$\hat{u}_6(x) = \begin{bmatrix} -0.0065625 \\ -0.00777875 \end{bmatrix}' \hat{x} + 1.01.$$

The state x jumps from the region P_9 to the region $\hat{P}_6$ after quantization. Therefore, we get the bound of absolute error of $u(x)$ is 0.012874036265283 and the absolute error of $u(x)$ obtained from real computation by sequential search is 0.011854337345.

From this example, we obtain our bounds by taking random quantization errors. In reality, data are stored in low available memory with fixed bits. In next a series of tests we still take the same example, but quantize data using a-bit total word length, 1-bit for sign, and b-bit for fraction length in fi function.

We take a random state x and it is in the region P_j, but after quantization it jumps to the region $\hat{P}_i$, where P_i and P_j are neighbors. In Figure 3, we plot the a priori bounds of absolute errors of $u(x)$ as described in Theorem 2.4 and Lemma 2.9, the a posteriori bounds of absolute errors of $u(x)$ as described in Subsection III-B, and plot the absolute errors of $u(x)$ by using sequential search in real computation.

From Figure3, the maximum of difference between our a priori bounds and real errors is about 0.9 and the maximum of difference between our a posteriori bounds and real errors is about 0.1 for $a = 12$ and $b = 5$ (upper). The maximum of difference between a priori bounds and real errors is about 0.1 and the maximum of difference between our a posteriori bounds and real errors is about 0.01 for $a = 16$ and $b = 9$ (bottom). From this test, we can see that the a posteriori bounds are about 10 times sharper than the a priori bounds if taking the same a and b for bits. Moreover, when we take more bits for the total word length and the fractional part, both a priori bounds and a posteriori bounds become sharper and sharper.

In the next test, we compare the a posteriori bounds without rescaling data with those bounds after rescaling data by take same random states. The upper of Figure 4 presents the a posteriori bounds of absolute errors of $u(x)$ for $a = 12$ and $b = 5$ without rescaling data, while the bottom of Figure 4 presents the bounds after rescaling, in which we take

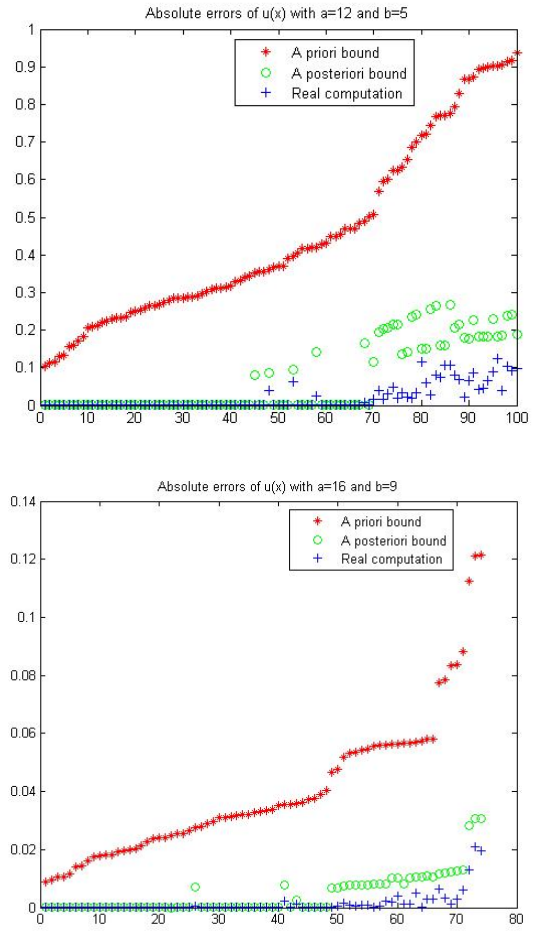

Fig. 3. Testing data with a-bit word total length and b-bit for fraction length. Upper: a priori bounds, a posteriori bounds and real computation absolute errors with $a = 12$ and $b = 5$. Bottom: both bounds of absolute errors of $u(x)$ and absolute errors in real computation with $a = 16$ and $b = 9$.

$a = 12$ and $b = 10$ for regions and states, and take $a = 17$ and $b = 10$ for control inputs. From Figure 4, we can see that the a posteriori bounds without rescaling data are between 0 and 0.35, and the a posteriori bounds for data after rescaling is less than 0.035. Though the bounds become smaller after rescaling, the real error of the absolute difference become smaller as well after rescaling.

In previous tests, we notice that for some states the posteriori bounds are almost near zero. Since we take random states for random regions, the posteriori bounds for some common hyperplane may be very small, but for some common hyperplane may be not. Next, we would like to test the states near the common hyperplanes between two neighbor regions to see in which regions the corresponding control inputs of the states are stable or not. We take the random states around the common hyperplane, such that the states are obtained as $x = \bar{x} + 0.5\mathrm{rand}(2, 1)$ where $\bar{x}$ is on the hyperplane in regions and $\mathrm{rand}(2, 1)$ is an uniformly distributed random vector.

We summarize our test in Table I. For example, the bounds

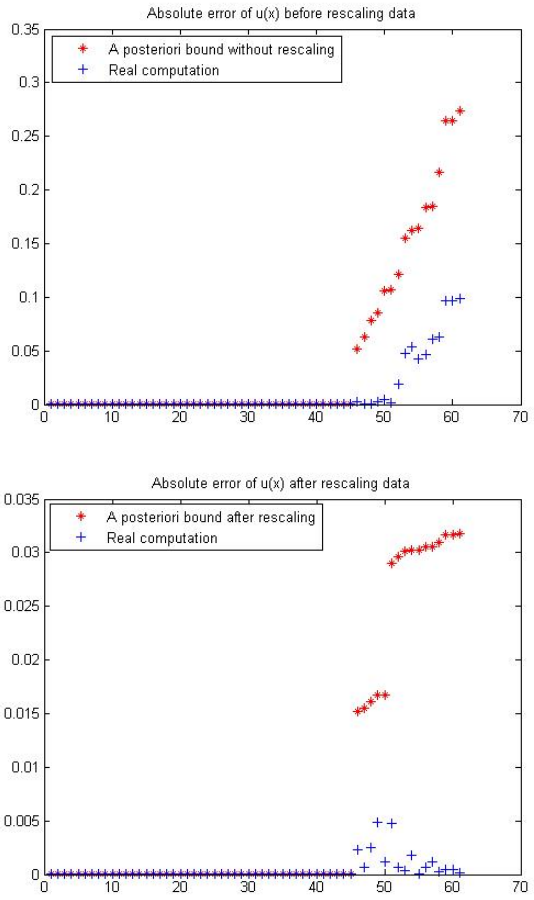

Fig. 4. Upper: A posteriori bounds and real computation absolute errors with $a = 12$ and $b = 5$. Bottom: A posteriori bounds and real computation absolute errors after rescaling with $a = 12$ and $b = 10$ for regions and with $a = 17$ and $b = 10$ for the control input.

Neighboring regions	Maximum and minimum (posteriori)	Corresponding real errors
1 and 9	less than 10^{-4}	less than 10^{-4}
2 and 8	less than 10^{-4}	less than 10^{-4}
3 and 4	0.270, 0.121	0.071, 0.019
3 and 5	0.191, 0.058	0.036, 0.022
3 and 12	0.245, 0.184	0.102, 0.069
4 and 7	less than 10^{-4}	less than 10^{-4}
5 and 6	less than 10^{-4}	less than 10^{-4}
6 and 9	less than 10^{-4}	less than 10^{-4}
7 and 8	less than 10^{-4}	less than 10^{-4}
7 and 13	0.181, 0.171	0.104, 0.090
10 and 13	0.248, 0.225	0.118, 0.111
11 and 12	0.245,0.228	0.097, 0.963

TABLE I

A POSTERIORI BOUNDS AND REAL ERRORS FOR THE STATES NEAR A SINGLE COMMON HYPERPLANE WITH $a = 12$ AND $b = 5$

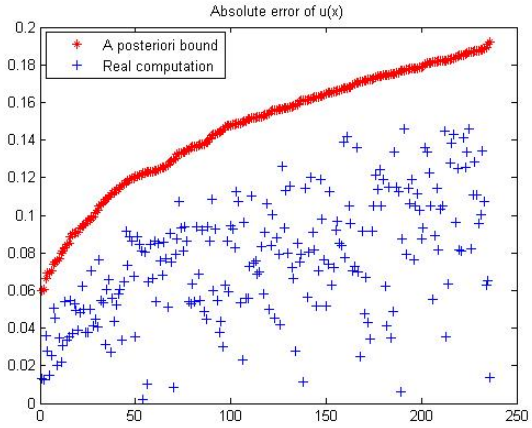

Fig. 5. A posteriori bounds and real errors with $a = 12$ and $b = 5$ for the states near the common hyperplane between regions 3 and 5.

and the real errors are less than 10^{-4}, if the state is in region P_1 or P_9 and after quantization the state jumps into region $\hat{P}_9$ or $\hat{P}_1$. The maximum and minimum bounds are 0.191 and 0.058, respectively, and the corresponding real errors are 0.036 and 0.022, if the state is in region P_3 or P_5 and after quantization the state jumps into region $\hat{P}_3$ or $\hat{P}_5$. To see detail for every common hyperplane, we take the common hyperplane between region 3 and 5 as an example. Figure 5 presents the a posteriori bounds and real computation errors.

V. CONCLUSIONS

In this paper we have derived a priori bounds and a posteriori bounds of accuracy analysis of control inputs when data are quantized and stored in low available memory control hardware. Based on those bounds, one can decide the bits and accuracy in the control input.

APPENDIX

The proof of Lemma 2.9 is provided in the following.

Proof: Suppose $hx \leq k$ for $x \in P_j$ and $hx \geq k$ for $x \in P_i$. To get the accuracy for $u(x)$, we have

$$\begin{aligned} |\hat{u}(\hat{x}) - u(x)| &\leq |F_i x + G_i - F_j x - G_j| \\ &+ |F_i \Delta x + \Delta F_i x + \Delta F_i \Delta x + \Delta G_i|. \end{aligned}$$

From the above inequality, in order to get the upper bound of difference of $u(x)$, we have to get the upper bound of $|F_i x + G_i - F_j x - G_j|$. Since the state x after quantization falls to the region P_i, by Lemma 2.2, we have $-\delta < hx - k \leq 0$. For $\forall x_0 \in \mathcal{Z}$, we have $hx = k$. Since $u(x)$ is a linear continuous affine function, we have

$$F_i x_0 + G_i = F_j x_0 + G_j, \tag{6}$$

for $\forall x_0 \in \mathcal{Z}$. We note that for $n = 1$, it is obvious that x_0 is on $hx = k$. For $n = 2$, $F_i x + G_i = F_j x + G_j$ is a line and $hx = k$ also is a line. So these two lines are the same.

Suppose that the maximum value of $|F_i x + G_i - F_j x - G_j|$ occurs at some $x \in P_j$, when it is in the hyperplane $hx - k = -\beta$, where $0 < \beta \leq \delta$. Since $hx_0 = k$, we have $h(x - x_0) = -\beta$. Furthermore, $h\left(\frac{k}{-\beta}(x - x_0)\right) = k$ which

implies that $\frac{k}{-\beta}(x - x_0)$ is on the hyperplane $hx = k$. So,

$$F_i\left(\frac{k}{-\beta}(x - x_0)\right) + G_i = F_j\left(\frac{k}{-\beta}(x - x_0)\right) + G_j. \quad (7)$$

According to equalities (6) and (7), we get

$$\begin{aligned} |F_i x + G_i - F_j x - G_j| &= \frac{\beta}{|k|}|G_i - G_j| \\ &\leq \frac{\delta}{|k|}|G_i - G_j|. \quad (8) \end{aligned}$$

Consequently, we obtain the result as expected.

■

REFERENCES

[1] J. B. Rawlings and D. Q. Mayne, *Model predictive control: Theory and design*. Nob Hill Pub., 2009.

[2] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control engineering practice*, vol. 11, no. 7, pp. 733–764, 2003.

[3] S. Di Cairano, "An industry perspective on mpc in large volumes applications: Potential benefits and open challenges," in *Proc. 4th IFAC Nonlinear Model Predictive Control Conference*. Citeseer, 2012, pp. 52–59.

[4] D. Hrovat, S. Di Cairano, H. E. Tseng, and I. V. Kolmanovsky, "The development of model predictive control in automotive industry: A survey," in *IEEE Conf. Control Applications*. IEEE, 2012, pp. 295–302.

[5] A. Bemporad, F. Borrelli, and M. Morari, "Model predictive control based on linear programming - the explicit solution," *Automatic Control, IEEE Transactions on*, vol. 47, no. 12, pp. 1974–1985, Dec 2002.

[6] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, Jan. 2002.

[7] S. Di Cairano, D. Yanakiev, A. Bemporad, I. Kolmanovsky, and D. Hrovat, "Model predictive idle speed control: Design, analysis, and experimental evaluation," vol. 20, no. 1, pp. 84 –97, 2012.

[8] S. Di Cairano, J. Doering, I. Kolmanovsky, and D. Hrovat, "MPC-based control of engine deceleration with open torque converter," Maui, HI, 2012, pp. 3753–3758.

[9] S. Di Cairano, H. Tseng, D. Bernardini, and A. Bemporad, "Vehicle yaw stability control by coordinated active front steering and differential braking in the tire sideslip angles domain," vol. 21, no. 4, pp. 1236–1248, 2013.

[10] S. Di Cairano, H. Park, and I. Kolmanovsky, "Model predictive control approach for guidance of spacecraft rendezvous and proximity maneuvering," *Int. J. Robust and Nonlinear Control*, vol. 22, no. 12, pp. 1398–1427, 2012.

[11] J. E. Goodman and J. O'Rourke, Eds., *Handbook of Discrete and Computational Geometry*. Boca Raton, FL, USA: CRC Press, Inc., 1997.

[12] P. Tøndel, T. A. Johansen, and A. Bemporad, "Evaluation of piecewise affine control via binary search tree," *Automatica*, vol. 39, no. 5, pp. 945–950, 2003.

[13] F. Bayat, T. A. Johansen, and A. A. Jalali, "Using hash tables to manage the time-storage complexity in a point location problem: Application to explicit model predictive control," *Automatica*, vol. 47, no. 3, pp. 571–577, 2011.

[14] T. Geyer, F. D. Torrisi, and M. Morari, "Optimal complexity reduction of piecewise affine models based on hyperplane arrangements," in *American Control Conference, 2004. Proceedings of the 2004*, vol. 2. IEEE, 2004, pp. 1190–1195.

[15] M. Kvasnica, J. Löfberg, and M. Fikar, "Stabilizing polynomial approximation of explicit mpc," *Automatica*, vol. 47, no. 10, pp. 2292–2297, 2011.

[16] M. Kvasnica and M. Fikar, "Clipping-based complexity reduction in explicit mpc," *Automatic Control, IEEE Transactions on*, vol. 57, no. 7, pp. 1878–1883, 2012.

[17] E. C. Kerrigan, J. L. Jerez, S. Longo, and G. A. Constantinides, "Number representation in predictive control," in *Proc. 4th IFAC Nonlinear Model Predictive Control Conference*. Citeseer, 2012, pp. 60–67.

[18] S. Longo, E. C. Kerrigan, and G. A. Constantinides, "Constrained lqr for low-precision data representation," *Automatica*, vol. 50, no. 1, pp. 162–168, 2014.

[19] M. Kvasnica, P. Grieder, M. Baotic, and M. Morari, "Multi-parametric toolbox (mpt)," in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, R. Alur and G. Pappas, Eds. Springer Berlin Heidelberg, 2004, vol. 2993, pp. 448–462.

[20] A. Airan, B. Bhartiya, and M. Bhushan, "Linear machine: A novel approach to point location problem," Preprints of the 10th IFAC international symposium on dynamics and control of process systems, 2013.