NONLINEAR MODEL PREDICTIVE CONTROL WITH REGULABLE COMPUTATIONAL COST

Y. Q. He and J. D. Han

ABSTRACT

Nonlinear model predictive control (NMPC) suffers from problems of closed loop instability and huge computational burden, which greatly limit its applications in real plants. In this paper, a new NMPC algorithm, whose stability is robust with respect to regulable computational cost, is presented. First, a new generalized pointwise min-norm (GPMN) control, as well as its analytic form considering a super-ball type input constraint, is given. Second, the GPMN controller is integrated into a normal NMPC algorithm as a structure of control input profile to be optimized, called GPMN enhanced NMPC (GPMN-ENMPC). Finally, a numerical example is presented and simulation results exhibit the advantage of the GPMN-ENMPC algorithm: computational cost can be regulated according to the computational resources with guaranteed stability.

Key Words: Control Lyapunov functions, nonlinear model predictive control, input constraints.

I. INTRODUCTION

Model based predictive control (MPC) has been extensively researched and applied in practice since it was first proposed in the 1970s. From the 1970s to 1980s, MPC was mainly used to realize the sub-optimal control of linear plants [1]. Recently, with rapid advances in computer science, nonlinear MPC (NMPC) algorithms have also been paid attention to [2–4].

However, nominal NMPC strategies result often in high computational cost and closed loop instability. This makes the gap between theory development of NMPC and its applications in reality larger and larger, especially when the system has fast time-varying dynamics [3]. On the one hand, in order to guarantee the closed loop stability of NMPC, some extra strategies must be considered, such as, the length of the horizon [4]; terminal state constraints [3, 5, 6]; control Lyapunov function (CLF) [3] and [7]. However, each approach above will make the computational cost increase greatly because the number of either constraints or optimizing variables will be augmented. On the other hand, the high computational burden of NMPC, coming from the process of solving the optimal control problem at each time step, can often be reduced by decreasing the number of optimized variables [8], which, unfortunately, will deteriorate the closed loop stability due to the inaccurate solving of the optimal control problem. Therefore, it can be concluded that the stability and computational cost of NMPC algorithms are deteriorated by each other. Also how to design a fast and stable NMPC algorithm has been a difficult problem that many researchers are pursuing.

Control Lyapunov function is a newly introduced concept to design a nonlinear controller. However, due to the difficulties in constructing CLFs and the inflexibility of the existing CLF-based controller design strategy, its practicability has been disputed for many years. Thus, in preceding work [9], the authors have designed a new CLF-based nonlinear control design method, called the generalized pointwise min-norm controller (GPMN), where the design of the CLF-based...
controller is continuous assuringly and more flexible than the original version.

Recently, the concept of CLF has become more attractive due to its use in NMPC, and there appear some CLF-based NMPC algorithms [3] where CLF is used not only to ensure the stability but also to improve the optimality of the closed loop with a pre-defined Lyapunov function. Unfortunately, the huge computational burden is still a problem in these algorithms. Thus in this paper we attempt to design a new stable NMPC to partially solve the conflict between the computational burden and the closed performance.

II. THE FUNDAMENTALS

2.1 Normal NMPC

Consider the following nonlinear system,

\[ \dot{x} = f(x) + g(x)u \]

\[ u \in U \subseteq \mathbb{R}^m \]

where \( x \in \mathbb{R}^n \) is state vector, \( u \in \mathbb{R}^m \) is input vector, \( f(\cdot) \) and \( g(\cdot) \) are nonlinear smooth functions with \( f(0) = 0 \), and \( U \) is the control constraint.

A typical continuous version NMPC algorithm of system (1) can be denoted as follows,

\[ u^* = \arg\min_{u \in U} J(x, u) \]

\[ J(x, u) = \int_0^{t+T} I(x(t), u(t)) \, dt + \varphi(x(t+T)) \]

s.t. \( \dot{x} = f(x) + g(x)u, \ u \in U, \ \forall t \in [t, t+T] \)

where \( I(\cdot, \cdot) \) is a continuous and positive definite function with \( I(0, 0) = 0 \), \( t \) and \( T \) are respectively the current time and predictive horizon, \( \varphi(x(t+T)) \) is the terminal cost function, \( X_f \) is the terminal constraint set.

Generally, for a nonlinear system like (1), algorithm (2) will result in large computational burden that may be far beyond the computational ability of current CPU for real time implementation. Therefore, our main object in the following sections is to design a real time NMPC algorithm through modifying algorithm (2).

2.2 Concept of CLF and corresponding results

Definition 1. For system (1), if there exists a \( C_1 \) function \( V(x) : \mathbb{R}^n \rightarrow \mathbb{R}^+ \) such that

i. \( V(0) = 0, \ V(x) > 0 \) if \( x \neq 0 \);

ii. \( a_1(\|x\|) < V(x) < a_2(\|x\|) \);

iii. \( \inf_{u \in U \subseteq \mathbb{R}^m} [V(x) f(x) + V(x) g(x) u] < 0, \ \forall x \in \mathbb{R}^n \),

where \( a_1(\cdot) \) and \( a_2(\cdot) \) are class \( K_\infty \) functions [10], \( W \) is a subset of \( \mathbb{R}^n \) including zero point, then \( V(x) \) is called a CLF of system (1). Moreover, if \( W \) can be chosen as \( \mathbb{R}^m \) and \( V(x) \) satisfies:

\[ V(x) \to \infty \Rightarrow \|x\| \to \infty \]  \hspace{1cm} (3)

\( V(x) \) is called global CLF of system (1).

Definition 1 indicates that if one can obtain a CLF of system (1), a ‘permitted’ control input set can always be found at every state, and the control action inside the set can guarantee the closed loop stability of the system. Subsequently, in order to design a stable controller, what one needs to do is only to find an approach to select a sequence of control actions from the ‘permitted control set’, see Fig. 1.

Several universal formulas to construct continuous stable controllers based on known CLF have been proposed [11, 12]. Wherein Freeman’s controller, also called the pointwise min-norm (PMN) controller can be described by

\[ \min_{u \in U} \|u\| \]

s.t. \( V_x[f(x) + g(x)u] \leq -\sigma(x) \) \hspace{1cm} (4)

where \( \sigma(x) \) is a pre-selected positive definitely function.

III. GPMN CONTROLLER

In Freeman’s controller, \( \sigma(x) \) is the only parameter to be designed. The selection of \( \sigma(x) \) is, however, very difficult since it is greatly related to the stability region of the closed loop system. Thus, in this section, a guide function will be proposed as design parameters of a new controller, shown in the following proposition,

Proposition 1. If \( V(x) \) is a local CLF of system (1) in \( \Omega_c \),

\[ \Omega_c = \{x \in \mathbb{R}^n | V(x) \leq c, c > 0\} \]

and \( \xi(x) : \mathbb{R}^n \rightarrow \mathbb{R}^m \) is a continuous guide function such that \( \xi(0) = 0 \), then controller (5) can stabilize system (1).

\[ u_{\xi(x)}(x) = \arg\min_{u \in K_V(x)} \{\|u - \xi(x)\|\} \]

\[ K_V(x) = \{y | V_x f(x) + V_x g(x) y \leq -\sigma(x), \ y \in U\} \]

where \( \sigma(x) \) is a positive definitely function.
The proof of Proposition 1 can be easily done through taking $V(x)$ as a Lyapunov function candidate.

Controller (5) is the so-called GPMN controller. The difference between GPMN and the ordinary PMN of (4) is illustrated in Fig. 1, for the ordinary PMN algorithm, the control input has the minimum ‘permitted’ norm (as close to the state-axis as possible), while for GPMN algorithm, the control input has the nearest distance from the guide function $\xi(x)$.

In order to use GPMN control according to (5) in reality, especially in the NMPC algorithm, analytical form of (5) is necessary to be studied. Thus, the analytical form of GPMN controller will be given with respect to two different cases in the following.

### 3.1 Without input constraints

If there are no constraints on inputs, analytical expression of controller (5) is as

$$u_{\xi(x)}(x) = \begin{cases} 
  -\left[ V_x f + \sigma(V_x g \xi(x)) \right] g^T V_x^T + \xi(x) \\
  V_x f + \sigma + V_x g \xi(x) > 0 \\
  \xi(x) \quad V_x f + \sigma + V_x g \xi(x) \leq 0 
\end{cases} \quad (6)$$

where $(u_1, \ldots, u_m)$ is the input vector, $r$ is the radius of the super-ball. Indeed, for most $U$, we can find a maximally inscribed super sphere $B$ of $U$, and then use $B$ to replace $U$ before continuing the following steps.

To obtain the analytical expression of (5) with constraint (7), the following steps are given.

**Step 1.** For every state $x$, the following equation is used to describe a super plane in $R^m (u \in R^m)$:

$$V_x f(x) + \sigma(x) + V_x g(x)u = 0 \quad (8)$$

The distance $d$ from zero to the super plane (8) is,

$$d = \frac{|V_x f(x) + \sigma(x)|}{\sqrt{V_x g(x)g^T(x)V_x}}. \quad (9)$$

**Step 2.** From (9), the ‘permitted’ stable control input set $K_V(x)$ in (5) can be depicted as Fig. 2a (the region filled by the dashed line), the right figure (left figure) shows the case that the super plane of (8) intersects (does not intersect) with the super sphere (7).

With respect to the case Fig. 2a-left, it is easy to obtain a minimal distance from any point $P$ to $K_V(x)$, and the corresponding point $Q$ in $K_V(x)$ with minimal distance from $P$ can also be obtained (i.e., the point of intersection of the super sphere (7) and the beeline connecting the centre of (7) and $P$). With respect to the case of Fig. 2a-right, the maximally inscribed super-ball $B'$ is used to replace the $K_V(x)$ (see Fig. 2b). Thus the same way as that above can be followed to obtain the minimal distance from any point $P$ to $B'$ and the corresponding point in $B'$.

3.2 With input constraints

If there exist input constraints, the analytical expression of (5) might be complicated or even do not exist. Therefore, in this section we only discuss a special case: super-spherical input constraint, i.e.,

$$U = \{(u_1, \ldots, u_m)|u_1^2 + \cdots + u_m^2 \leq r^2\} \quad (7)$$
Step 3. A new ‘permitted’ stable control input sets \( \overline{K_V}(x) \) can be defined as follows,

\[
\overline{K_V}(x) = \begin{cases} 
U, & \frac{|V_x f(x) + \sigma(x)|}{\sqrt{V_x g(x)g^T(x)V_x^T}} > r \\
[u||u-\gamma(x)|| \leq R(x)], & \frac{|V_x f(x) + \sigma(x)|}{\sqrt{V_x g(x)g^T(x)V_x^T}} \leq r 
\end{cases}
\]

where

\[
\gamma(x) = -\left( \frac{V_x f(x) + \sigma(x)}{2V_x g(x)g^T(x)V_x^T} \right) + \frac{r}{2\sqrt{V_x g(x)g^T(x)V_x^T}} g^T(x)V_x^T
\]

\[
R(x) = \frac{|V_x f(x) + \sigma(x)|}{\sqrt{V_x g(x)g^T(x)V_x^T}}
\]

It is obvious that \( \overline{K_V}(x) \subseteq K_V(x) \), i.e., the closed loop stability can be ensured if \( u(x) \in \overline{K_V}(x) \).

Step 4. The minimal distance from any point \( P \) to \( K_V(x) \) can be obtained as in Fig. 2b, and the analytic expression of the GPMN controller with input constraint can be described as:

\[
\tilde{u}_\xi(x)(x) = \begin{cases} 
\xi(x), & \|\xi(x) - \gamma(x)|| \leq R(x) \\
R(x), & \|\xi(x) - \gamma(x)|| \leq \frac{[\xi(x) - \gamma(x)] + \gamma(x)}{R(x)} \end{cases}
\]  

(11)

From the preceding procedure, it is evident that (11) is the solution of (5) with \( K_V(x) \) placed by \( \overline{K_V}(x) \).

IV. GPMN-ENHANCED NONLINEAR MODEL PREDICTIVE CONTROL

In order to obtain a stable NMPC with regulable computational burden, we propose to use GPMN to parameterize the control input profile in algorithm (2).
Assuming that \( \vartheta(x, \theta) \) is a parameterized map, where \( \theta \) is the vector parameter to be designed. The following equations are called parameterized NMPC:

\[
\begin{align*}
\dot{x} &= f(x) + g(x)\vartheta(x, \theta) \\
\vartheta(x, \theta) &= \arg \min_{\theta} J(x, \theta) \\
J(x, \theta) &= \int_{t}^{t+T} [l(x, \vartheta(x, \theta))] \, d\tau \\
s.t. \quad x &= \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \forall t \in [t, t+T].
\end{align*}
\]

Different from the ordinary NMPC (2) where one tries to optimize the continuous control profile of \( u \), NMPC (12) achieves the optimal performance by selection of the parameter vector \( \theta \). Thus, the computational cost will be dictated mainly by the dimension of \( \theta \).

Subsequently, GPMN control (5) is proposed to replace \( (x, \theta) \) in (12), leading to the GPMN enhanced NMPC and the following proposition.

**Proposition 2.** Assuming \( V(x) \) is a known CLF of system (1), \( \Omega \) is the stability region, then controller (13) with the following \( \vartheta(x, \theta) \)

\[
\vartheta(x, \theta) = u\xi(x, \theta)(x, \theta) = \arg \min_{u \in \mathcal{K}(x)} \{ ||u - \xi(x, \theta)|| \}
\]

where \( u\xi(x, \theta)(x, \theta) \) is the GPMN control and \( \xi(x, \theta) \) the guide function in (5), is stable in \( \Omega \). Furthermore, if \( V(x) \) is a global CLF, controller (12)–(13), called GPMN-Enhanced NMPC (GPMN-ENMPC), is stable over \( \mathbb{R}^n \).

**Proof.** At any time instant \( t \), \( \theta^* \) is the optimal parameter of the GPMN-ENMPC algorithm, the control input at \( t \) is \( u\xi(x, \theta^*) \) which is a GPMN controller of (1). Thus, from Proposition 1, \( u\xi(x, \theta^*) \) can guarantee a negative definite \( \dot{V}(x) \). Due to the randomness of time instant \( t \), the GPMN-ENMPC can make \( \dot{V}(x) \), along with the trajectory of system (1), negative in any time instant, which means that the closed loop stability of controller (12)–(13) can be guaranteed.

**Remark 1.** From Proposition 2, closed loop stability of GPMN-ENMPC algorithm can be guaranteed regardless of the structure of \( \xi(x, \theta) \) and selection of parameter \( \theta \). This is the reason why we are able to obtain the flexibility in regulating the structure of the guide function and changing the dimension of \( \theta \) to pursue interested closed loop performance and proper computational cost.

**Remark 2.** Compared to the algorithm in [3], huge computational burden problem is improved due to the following two reasons: (i) as a key reason of the huge computational burden of the proposed algorithm, the dimension of optimizing variables of GPMN-ENMPC algorithm is independent of the predictive horizon; (ii) online considerations of control input constraints are not necessary since it can be dealt with offline during designing of the GPMN controller.

Theoretically, \( \xi(x, \theta) \) in (13) can be selected in any form since it does not affect the closed loop stability. However, \( \xi(x, \theta) \) will definitely affect the other performance of the GPMN-ENMPC except for the stability. In this paper, we propose to use Bellman’s principle of optimality to design \( \xi(x, \theta) \). In general, \( J(x, \theta) \) in (2) has the following quadratic form,

\[
J(x, u) = \left( x^{T} P x + u^{T} Q u \right) \, d\tau.
\]

Thus, from Bellman’s principle of optimization, the optimal control is as,

\[
u^* = -\frac{1}{2} \left( Q^{-1} \right)^{T} g^{T}(x) \frac{\partial J^*}{\partial x}
\]

where \( J^*(x^*, u) \) is the optimal value function of \( J(x, u) \). Unfortunately, in most situations, \( J^*(x^*, u) \) is impossible to be obtained.

Simultaneously, by the Stone-Weierstrass theorem [13], the following polynomial can be used to approximate the unknown optimal value function,

\[
\begin{align*}
B_k^*(x_1, \ldots, x_n) &= \sum_{v_1, \ldots, v_n \in \mathbb{R}} J^* \left( \frac{v_1}{k}, \ldots, \frac{v_n}{k} \right) p_{k; v_1, \ldots, v_n} (x_1, \ldots, x_n).
\end{align*}
\]

By selecting polynomial parameters optimally, a ‘quasi-optimal’ function, which is close to \( J^*(x^*, u) \), can be obtained. It means, \( \xi(x, \theta) \) can be selected as Bernstein polynomial with pre-specified order, i.e.,

\[
\begin{align*}
\xi(x, \theta) &= \sum_{v_1, \ldots, v_n \in \mathbb{R}} \theta_{v_1, \ldots, v_n} p_{k; v_1, \ldots, v_n} (x_1, \ldots, x_n)
\end{align*}
\]

where \( v_1, \ldots, v_n \geq 0 \) are the parameters to be optimized, \( k \) is the order of the Bernstein polynomial.

**Remark 3.** We state that the computational cost is regulable since one can select \( k \) to meet the CPU capability of a particular system.

© 2011 John Wiley and Sons Asia Pte Ltd and Chinese Automatic Control Society
Remark 4. Although $k$ does not influence the closed loop stability, there still exists a trade-off between the computational cost and the optimal performance which is determined by $\zeta(x, \theta)$.

V. NUMERICAL EXAMPLES

Consider the following pendulum equation [4],

$$
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= \frac{19.6 \sin x_1 - 0.2 x_2^2 \sin 2x_1}{4/3 - 0.2 \cos^2 x_1} - 0.2 \cos x_1 u + \frac{4/3 - 0.2 \cos^2 x_1}{4/3 - 0.2 \cos^2 x_1} u.
\end{align*}
$$

(17)

A local CLF of system (17) can be given as,

$$
V = x^T P x = [x_1 \ x_2] \begin{bmatrix} 151.57 & 42.36 \\ 42.36 & 12.96 \end{bmatrix} x.
$$

(18)

Select

$$
\sigma(x) = 0.1(x_1^2 + x_2^2).
$$

(19)

We first design the PMN controller of system (17) according to (5),

$$
\begin{align*}
\rho(x) &= \begin{cases} 
\frac{\rho(x)(4/3 - 0.2 \cos^2 x_1)}{0.4 \cos x_1 (42.36 x_1 + 12.96 x_2)} & \rho(x) < 0 \\
0 & \rho(x) \leq 0
\end{cases} \\
\rho(x) &= 2 \begin{pmatrix} (151.57 x_1 + 42.36 x_2) x_2 \\
+ (42.36 x_1 + 12.96 x_2) \\
19.6 \sin x_1 - 0.2 x_2^2 \sin 2x_1 \\
+ (10.54 x_2 + 1.27 x_1)^2 + x_2^2
\end{pmatrix}
\end{align*}
$$

(20)

Given the initial state $x_0 = [x_1 \ x_2]^T = [-1, 2]^T$, and the desired state $x_d = [0, 0]^T$, the time response of the closed loop is shown in Fig. 3 as the solid line. It can be seen that the closed loop with PMN controller (20) has a very low convergence rate for state $x_1$. This is mainly because the only regulatable parameter to change the closed loop performance is $\sigma(x)$, which, unfortunately, is difficult to select properly due to its great influence on the stability region.

To design GPMN-ENMPC, two different guide functions are selected based on (16),

$$
\zeta(x, \theta) = \theta_{1,0} (1 - x_1 - x_2) + \theta_{1,1} x_1 + \theta_{0,1} x_2
$$

(21)

Other parameters of GPMN-ENMPC are designed as follows:

$$
J = \int_0^T \begin{bmatrix} 20 & 0 \\
0 & 1 \end{bmatrix} (x + 0.01 u^2) \ dt
$$

(22)

$$
l(x, u) = x^T \begin{bmatrix} 20 & 0 \\
0 & 1 \end{bmatrix} x + 0.01 u^2;
$$

$$
f(x) = \begin{bmatrix} x_2 \\
19.6 \sin x_1 - 0.2 x_2^2 \sin 2x_1 \\
4/3 - 0.2 \cos^2 x_1 \end{bmatrix}
$$

(23)

$$
g(x) = \begin{bmatrix} 0 \\
-0.2 \cos x_1 \\
4/3 - 0.2 \cos^2 x_1 \end{bmatrix}
$$

(24)

The integral of $J$ in the NMPC algorithm is approached by the following equation:

$$
\int_{t}^{t+T} \ h(t) \ dt = \sum_{i=1}^{\text{int}[T/\Delta T]} h(t+i\Delta T) \Delta T
$$

(25)

where the integral constant $\Delta T$ is 0.1s.

The genetic algorithm (GA) in the MATLAB toolbox is used to solve the optimization problem. The time response of the GPMN-ENMPC algorithm with different predictive horizon $T$ and approaching order are presented in Fig. 3, where the dotted line denotes the case of $T = 0.6s$ with guide function (24).
and the dashed line is the case of $T=1.5s$ with guide function (25). From Fig. 3, it can be seen that the convergence performance of the proposed NMPC algorithm is better than the PMN controller, and both the prediction horizon and the guide function will result in the change of the closed loop performance.

The improvement of the optimality is the main advantage of MPC compared with other controllers. In view of this, we propose to estimate the optimality by the following index function

$$J = \lim_{t \to \infty} \int_{0}^{t} \left( x^T \begin{bmatrix} 20 & 0 \\ 0 & 1 \end{bmatrix} x + 0.01u^2 \right) dt.$$  \hspace{1cm} (26)

The comparison results are summarized in Table I, from which we can obtain the following conclusions:

1. GPMN-ENMPC has better performance than the PMN controller in terms of optimization.
2. In most cases, GPMN-ENMPC with higher order $\xi(x, \theta)$ will usually result in a smaller cost than that with lower order $\xi(x, \theta)$. This is mainly because higher order $\xi(x, \theta)$ indicates larger inherent optimizing parameter space.

Another advantage of the GPMN-ENMPC algorithm is the flexibility of the trade-offs between the optimality and the computational time. The computational time is influenced by the dimension of optimizing parameters and the parameters of the optimizing algorithm, such as the maximum number of iterations and the size of the population. However, it will be natural that the optimality may deteriorate with the decreasing computational burden. In preceding paragraphs, we have researched the optimality of the GPMN-ENMPC algorithm with different optimizing parameters, and now the optimality comparisons among the closed loop systems with different GA parameters will be done. The results are listed in Table II, from which the optimality loss with the changing of the optimizing algorithm’s parameters can be observed. This can be used as the criterion to determine the trade-off between the closed loop performance and the computational efficiency of the algorithm.

Finally, in order to verify that the new designed algorithm improves the computational burden, simulations comparing the performance of the new designed algorithm and algorithm in [3] are conducted with the same optimizing algorithm—GA. The time interval of the two neighboring optimizations (TI in Table III) in the algorithm of [3] is important since control input is assumed to be constant at every time slice. Generally, a large time interval will result in poor stability. While our new GPMN-ENMPC algorithm results in a group of controller parameter, and the closed loop stability is independent of TI. Thus different TI is considered in these simulations of the algorithm in [3], and Table III lists the results. From Table III, the following items can be concluded:

1. With the same GA parameters, algorithm in [3] is more time-consuming and poorer in optimality than GPMN-ENMPC. This is easy to obtain through comparing results of Ex-2 and Ex-5.
2. In order to obtain similar optimality, GPMN-ENMPC takes much less time than the algorithm in [3]. This can be obtained by comparing the results of Ex-1/Ex-4 and Ex-6, as well as Ex-3 and Ex-5. The reasons for these phenomena have been introduced in Remark 2.

### VI. CONCLUSION AND FUTURE WORK

In this paper, nonlinear model predictive control (NMPC) is researched and a new NMPC algorithm is proposed. The main contributions of this paper are two-fold:

1. A new generalized point-wise min-norm (GPMN) control, which can be parameterized through a guide function, is introduced to design a real time and stable NMPC algorithm.
2. Input constraints with the form of super-ball can be dealt with offline to further reduce the online computational burden of the proposed algorithm.

The newly designed NMPC algorithm, called GPMN-enhancement NMPC (GPMN-ENMPC), has
the following three advantages:

1. Closed loop stability can be always guaranteed.
2. Performance other than optimality and stability can be considered in the new algorithm through selecting proper guide function.
3. Computational cost of the new NMPC algorithm is regulable according to the performance requirement and available CPU capabilities.

However, the newly designed algorithm can be further improved in the following three aspects in the future:

1. The selection of guide function is important in the GPMN-ENMPC algorithm since it will influence the closed loop performance except for stability. Thus, one aspect of our future research is to find out how the guide function influences the closed loop performance.
2. The optimizing problem to be solved online in GPMN-ENMPC algorithm is particular because the GPMN controller is often nondifferentiable. Although the GA algorithm presents its feasibility in this paper, new optimizing algorithm is still necessary to further reduce the computational burden.
3. In this paper, a super-ball type input constraint is considered to reduce the online computational burden. In the future, algorithms considering other kinds of input constraints and state constraints will be researched.

REFERENCES