View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by IntechOpen

We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

154
Countries delivered to

Our authors are among the
TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™
Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

3

A new kind of nonlinear model predictive control algorithm enhanced by control lyapunov functions

Yuqing He and Jianda Han
State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences
P.R.China

1. Introduction

With the abilities of handling constraints and performance of optimization, model based predictive control (MPC), especially linear MPC, has been extensively researched in theory and applied in practice since it was firstly proposed in 1970s (Qin & Badgwell, 2003). However, when used in systems with heavy nonlinearities, nonlinear MPC (NMPC) results often in problems of high computational cost or closed loop instability due to their complicated structure. This is the reason why the gaps between NMPC theory and its applications in reality are larger and larger, and why researches on NMPC theory absorbs numerous scholars (Chen & Shaw, 1982; Henson, 1998 ; Mayne, et al., 2000 ; Rawlings, 2000). When the closed loop stability of NMPC is concerned, some extra strategies is necessary, such as increasing the length of the predictive horizon, superinducing state constraints, or introducing Control Lyapunov Functions (CLF).

That infinite predictive/control horizon (in this chapter, predictive horizon is assumed equal to control horizon) can guarantee the closed loop stability is natural with the assumption of feasibility because it implicates zero terminal state, which is a sufficient stability condition in many NMPC algorithm (Chen and Shaw, 1982). In spite of the inapplicability of infinite predictive horizon in real plants, a useful proposition originated from it makes great senses during the development of NMPC theory, i.e., a long enough predictive horizon can guarantee the closed loop stability for most systems (Costa & do Val, 2003; Primbs & Nevistic, 2000). Many existing NMPC algorithm is on the basis of this result, such as Chen & Allgower (1998), Magni et al. (2001). Although long predictive horizon scheme is convenient to be realized, the difficulty to obtain the corresponding threshold value makes this scheme improper in many plants, especially in systems with complicated structure. For these cases, another strategy, superinducing state constraints or terminal constraints, is a good substitue. A typical predictive control algorithm using this strategy is the so called dual mode predictive control(Scokaert et al., 1999 ; Wesselowske and Fierro, 2003 ; Zou et al., 2006), which is originated from the predictive control with zero terminal state constrains and can increase its the stability region greatly. CLF is a new introduced concept to design nonlinear controller. It is firstly used in NMPC by Primbs et al. in 1999 to obtain two typical predictive control algorithm with guaranteed stability.

Unfortunately, each approach above will result in huge computational burden simultaneously since they bring either more constraints or more optimizing variables. It is well known that the high computational burden of NMPC mainly comes from the online optimization algorithm, and it can be alleviated by decreasing the number of optimized variables. But this often deteriorates the closed loop stability due to the changed structure of optimal control problem at each time step.

In a word, the most important problem during designing NMPC algorithm is that the stability and computational burden are deteriorated by each other. Another problem, seldom referred to but top important, is that the stability can only be guarangteed under the condition of perfect optimization algorithm that is impossible in reality. Thus, how to design a robustly stable and fast NMPC algorithm has been one of the most difficult problems that many researchers are pursued.

In this chapter, we attempt to design a new stable NMPC which can partially solve the problems referred to above. CLF, as a new introduced concept to design nonlinear controller by directly using the idea of Lyapunov stability analysis, is used in this chapter to ensure the stability. Firstly, a generalized pointwise min-norm (GPMN) controller (a stable controller design method) based on the concept of CLF is designed. Secondly, a new stable NMPC algorithm, called GPMN enhanced NMPC (GPMN-ENMPC), is given through parameterized GPMN controller. The new algorithm has the following two advantages, 1) it can not only ensure the closed loop stability but also decrease the computational cost flexibly at the price of sacrificing the optimality in a certain extent; 2) a new tool of guide function is introduced by which some extra control strategy can be considered implicitly. Subsequently, the GPMN-ENMPC algorithm is generalized to obtain a robust NMPC algorithm with respect to the feedback linearizable system. Finally, extensive simulations are conducted and the results show the feasibility and validity of the proposed algorithm.

2. Concept of CLF

The nonlinear system under consideration in this chapter is in the form as:

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ u &\in U \subset R^m \end{aligned} \tag{1}$$

where $x \subset R^n$ is state vector, $u \subset R^m$ is input vector, $f(*)$ and $g(*)$ are nonlinear smooth functions with $f(0) = 0$. U is the control constraint.

Definition I:

For system (1), if there exists a C_1 function $V(x)$: $x \subset R^n \rightarrow R^+ \cup \{0\}$, such that

1) $V(0) = 0$, $V(x) > 0$ if $x \neq 0$;
2) $a_1(||x||) < V(x) < a_2(||x||)$, where $a_1(*)$ and $a_2(*)$ are class K_∞ functions;
3) $\inf_{u \in U \subset R^m} [V_x(x)f(x) + V_x(x)g(x)u] < 0, \quad \forall x \in \Omega_c - \{0\}$,where $\Omega_c \triangleq \{x \in R^n : V(x) \leq c\}$.

then $V(x)$ is called a CLF of system (1). Moreover, if x can be chosen as R^n and $V(x)$ satisfies the following condition,

$$V(x) \rightarrow \infty \implies ||x|| \rightarrow \infty$$

then $V(x)$ is called a global CLF of system (1). ■

If system (1) has uncertainty terms, i.e.,

$$\begin{aligned} \dot{x} &= f(x) + g(x)u + l(x)\omega \\ y &= h(x) \\ u &\in U \subset R^m \end{aligned} \tag{2}$$

where $\omega \in R^q$ is external disturbance; $l(*)$ and $h(*)$ are pre-defined nonlinear smooth functions; y is the interested output. We have the following concept of robust version CLF – called H_∞CLF,

Definition II,

For system (2), if there exists a C_1 function $V(x)$: $x \subset R^n \rightarrow R^+ \cup \{0\}$, such that

1) $V(0) = 0$, $V(x) > 0$ if $x \neq 0$;
2) $a_1(||x||) < V(x) < a_2(||x||)$, where $a_1(*)$ and $a_2(*)$ are class K_∞ functions;
3) $\inf_{u \in R^m} \{V_x(x)[f(x) + g(x)u] + \frac{1}{2\gamma^2} V_x(x) l(x) l^T(x) V_x^T + \frac{1}{2} h^T(x) h(x)\} < 0, \forall x \in \Omega_{c_1} - \Omega_{c_2}$, where $c_1 > c_2$.

then $V(x)$ is called a local H_∞CLF of system (2) in $\Omega_{c_1} - \Omega_{c_2}$. Furthermore, $V(x)$ is called a global H_∞CLF if c_1 can be chosen $+\infty$ with $V(x) \rightarrow \infty$ as $|x| \rightarrow \infty$. ■

Definition I and II indicate that if we can obtain a CLF or H_∞CLF of system (1) or (2), a 'permitted' control set can be found at every 'feasible' state, and the control action inside the set can guarantee the closed loop stability of system (1) or input output finite gain L_2 stability of system (2). Subsequently, in order to complete the controller design, what one needs to do is just to find an approach to select a sequence of control actions from the 'permitted control set', see Fig. 1.

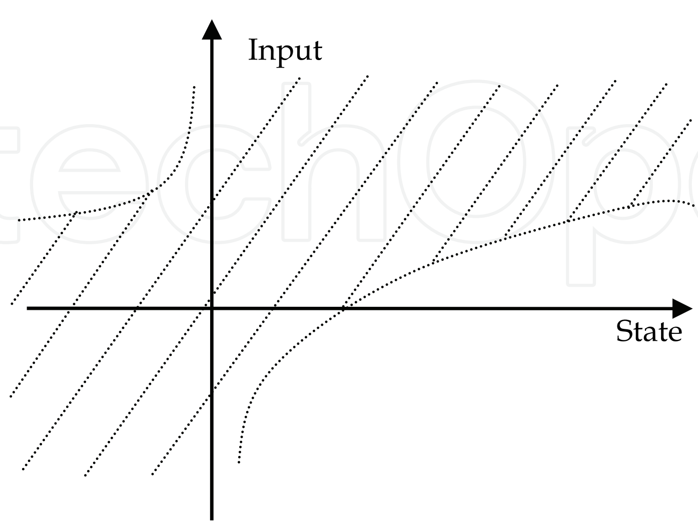


Fig. 1. Sketch of CLF, the shadow indicates the 'permitted' set of (x, u) $\dot{V}(x,u)$ along system (1)

CLF based nonlinear controller design method is also called direct method of Lyapunov function based controller design, and its difficulty is how to ensure the controller's continuousness. Thus, most recently, researchers mainly pay their attentions to designing continuous CLF based controller, and several universal formulas have been revealed. Sontag's formula (Sontag, 1989), for example, originated from the root calculation of 2nd-order equation, can be written as Eq. (3) through slightly modification by Freeman (Freeman & Kokotovic, 1996b),

$$u = \begin{cases} -\left[\dfrac{V_x(x) f + \sqrt{(V_x(x) f(x))^2 + q(x)(V_x(x) g(x) g^T(x) V_x^T(x))}}{V_x(x) g(x) g^T(x) V_x^T(x)} \right] & V_x g \neq 0 \\ 0 & V_x g = 0 \end{cases} \tag{3}$$

where $q(x)$ is a pre-designed positive definite function.
Pointwise Min-Norm (PMN) control is another well known CLF-based approach proposed by Freeman (Freeman & Kokotovic, 1996a),

$$\begin{aligned} \min_u \quad & \|u\| \\ s.t. \quad & V_x(x)[f(x) + g(x)u] \leq -\sigma(x) \\ & u \in U \end{aligned} \tag{4}$$

where $\sigma(x)$ is a pre-selected positive definite function. Controller (4) can also be explicitly denoted as (5) if the constraint set U can be selected big enough.

$$u = \begin{cases} -\dfrac{[V_x(x) f(x) + \sigma(x)] g^T(x) V_x^T(x)}{V_x(x) g(x) g^T(x) V_x^T(x)} & V_x(x) f(x) + \sigma(x) > 0 \\ 0 & V_x(x) f(x) + \sigma(x) \leq 0 \end{cases} \tag{5}$$

(3) and (5) provide two different methods on how to design continuous and stable controller based on CLF with respect to system (1). H_∞CLF with respect to system (2) is a new given concept, and there are no methods can be used to designed robust controller based on it. Although the closed loop stability can be guaranteed using controller (3) or controller (5), selection of parameters $q(x)$ or $\sigma(x)$ is too difficult to be used in real applications. This is mainly because these parameters heavily influence some inconsistent closed loop performance simultaneously. Furthermore, if the known CLF is not global, the selection of $q(x)$ and $\sigma(x)$ will also influence stability margin of the closed loop systems, which makes them more difficult to be selected (Sontag, 1989; Freeman & Kokotovic, 1996a). In this chapter, we will firstly give a new CLF based controller design strategy, which is superior compared to the existing CLF based controller design methods referred to above. Furthermore, the most important is that this new strategy can be used in designing robustly stable and fast NMPC algorithm.

3. GPMN-ENMPC

3.1 CLF based GPMN controller

Since $q(x)$ and $\sigma(x)$ in controller (3) and controller (5) are difficult to select, a guide function is proposed in this subsection into the PMN controller to obtain a new CLF based nonlinear controller with respect to system (1), in the following section, this controller will be generated with respect to system (2). In the new controller, $\sigma(x)$ is only used to ensure the stability of the closed loop, while the other desired performance of the controller, for example tracking performance, can be guaranteed by the guide function, which, as new controller parameters, can be designed without deteriorating the stability. The following proposition is the main result of this subsection.

***Proposition I*:**

If $V(x)$ is a CLF of system (1) in Ω_c and $\xi(x)$: $R^n \to R^m$ is a continuous guide function such that $\xi(0) = 0$, then, the following controller can stabilize system (1),

$$\begin{aligned} u(x) &= \arg\min_{u \in K_V(x)} \{\|u - \xi(x)\|\} \\ K_V(x) &= \{y \mid V_x(x) f(x) + V_x(x) g(x) y \le -\sigma(x),\ y \in U\} \end{aligned} \tag{6}$$

where $\sigma(x)$ is a positive definite function of state, and $\xi(x)$, called guide function, is a continuous state function.

Proof of Proposition I:

Let $V(x)$ be a Lyapunov function candidate for system (1), then we have

$$\dot{V}(x) = V_x(x) f(x) + V_x(x) g(x) u \tag{7}$$

Substitute Eq. (6) into (7), it is not difficult to obtain the following inequality,

$$\dot{V}(x) = V_x(x) f(x) + V_x(x) g(x) u \le -\sigma(x)$$

Because $\sigma(x)$ is a positive definite function, proposition I is proved. ■

Controller (6) is called Generalized Pointwise Min-Norm (GPMN) controller. The difference between the proposed GPMN controller and the normal PMN controller of Eq. (4) can be illustrated in Fig.2: for the normal PMN algorithm (Fig. 2a), the controller output in each state point has the minimum 'permitted' norm (close to the state-axis as much as possible), while the GPMN controller's output has nearest distance from the guide function $\xi(x)$ (Fig. 2b). Thus, $\xi(x)$ in GPMN controller is actual a performance criterion which the controller is expected to pursue, while $\sigma(x)$ dedicates only on providing the 'permitted' stable control input sets.

Up to now, the design of new GPMN controller has been completed. However, in order to use a GPMN controller in reality or in NMPC algorithm, analytical form of the solution of Eq. (6) is necessary to be studied.

Firstly, if there are no input constraints (or the input constraint sets are big enough), the analytical form of controller (6) can be obtained as follows, based on the projection theory,

$$u_{\xi(x)}(x)=\begin{cases}-\dfrac{[V_x f+\sigma+V_x g\xi(x)]g^T V_x^T}{V_x g g^T V_x^T}+\xi(x), & V_x f+\sigma+V_x g\xi(x)>0\\ \xi(x) & , \quad V_x f+\sigma+V_x g\xi(x)\le 0\end{cases} \tag{8}$$

Secondly, if there exist input constraints, the analytical expression of controller (6) might be very complicated or even inexistent. Thus in this subsection, only analytical form of controller (6) with a typical super ball input constraint is researched, i.e., input constraints is as

$$U=\{(u_1,\cdots,u_m)\,|\,u_1^2+\cdots+u_m^2\le r^2\} \tag{9}$$

where (u_1, … ,u_m) is the input vector, and r is the radius of the super ball.
In order to obtain the analytical expression of Eq. (6) with input constraint as Eq. (9), we propose the following 4 steps (For a general control input constraint U, one can always find a maximal inscribed super ball B of it, and then use B replacing U before continuing the following processes):

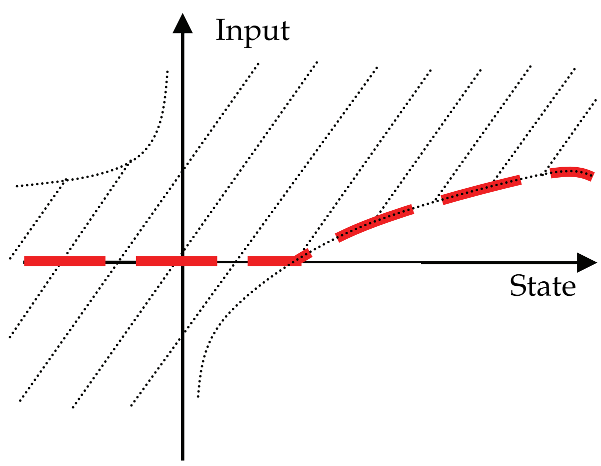


Fig. 2a. the sketch of PMN

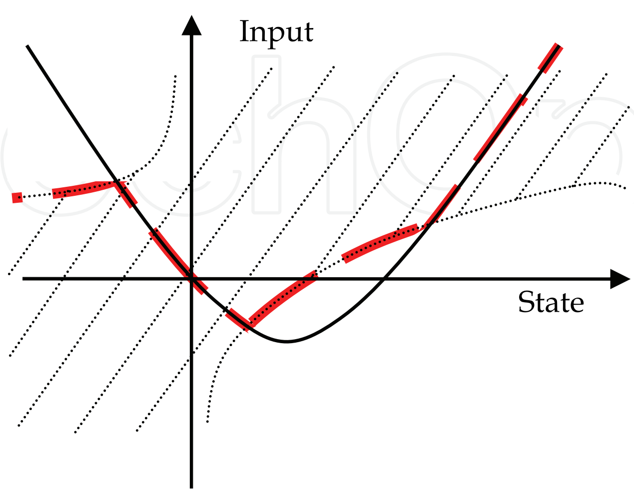


Fig. 2b. the sketch of GPMN
* the dashed line is the PMN controller in a) and the GPMN control in b); the solid line denotes the guide function of ξ(x).

Step1: For each state x, the following equation denotes a super plane in R^m ($u \in R^m$).

$$V_x f(x) + \sigma(x) + V_x g(x) u = 0 \tag{10}$$

Let d be the distance from zero to the super plane (10), we have,

$$d = \frac{|V_x f(x) + \sigma(x)|}{\sqrt{V_x g(x) g^T(x) V_x^T}} \tag{11}$$

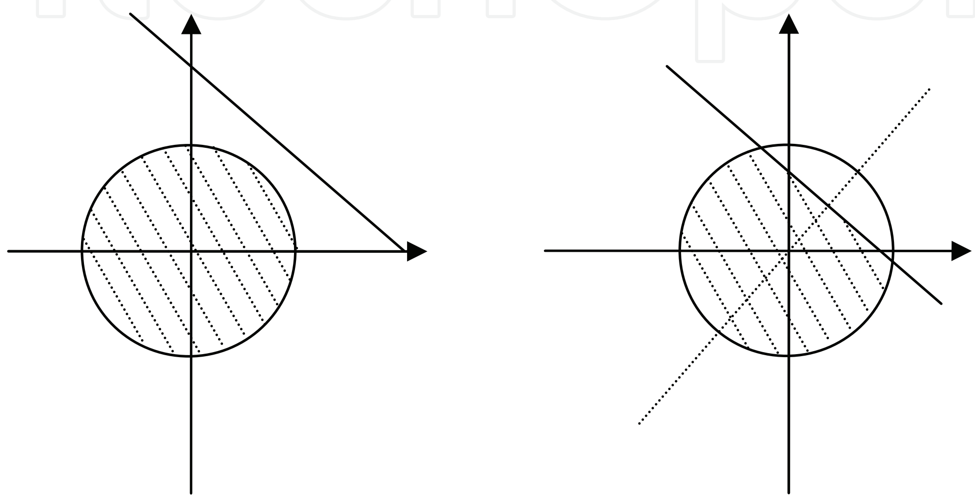


Fig. 3a

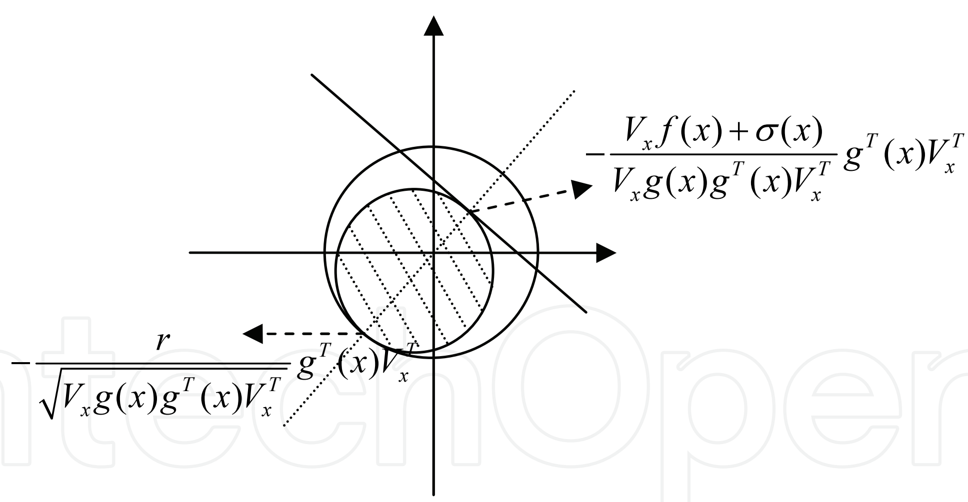


Fig. 3b.
* Sketch of the process to build the analytic GPMN controller

Step2: From Eq. (11), the 'permitted' stable control input set $K_V(x)$ in controller (6) can be denoted as Fig. 3a, where the right figure (left figure) is the case that the super plane of (10) intersects (does not intersect) with the super ball (9), and the region filled by the dotted line is the 'permitted' stable control input set. For the case denoted by the left figure of Fig. 3a, it is easy to obtain a minimal distance from any point p to $K_V(x)$, and the corresponding point, i.e., the controller's output, in $K_V(x)$ with minimal distance from p can also be obtained (the point of intersection of the super ball (9) and the beeline connecting the centre of it and p). With respect to the case of the right figure, the maximally inscribed super ball B' is used to replace $K_V(x)$ (see Fig. 3b). Thus, the same processes as above can be used to obtain the output of controller (6).

Step 3: A new 'permitted' stable control input sets $\overline{K_V(x)}$ is defined,

$$\overline{K_V(x)} = \begin{cases} U & \dfrac{|V_x f(x) + \sigma(x)|}{\sqrt{V_x g(x) g^T(x) V_x^T}} > r \\ \{u \mid \|u - \gamma(x)\| = R^2(x)\} & \dfrac{|V_x f(x) + \sigma(x)|}{\sqrt{V_x g(x) g^T(x) V_x^T}} \le r \end{cases} \tag{12}$$

where

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

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

It is obvious that $\overline{K_V(x)} \subseteq K_V(x)$, thus the stability of the closed loop can be ensured from Proposition I.

Step 4: The analytical expression of GPMN controller with super-ball input constraint can thus be described as

$$\overline{u}_{\xi(x)}(x) = \begin{cases} \xi(x) & \|\xi(x) - \gamma(x)\| \le R(x) \\ \dfrac{R(x)}{\|\xi(x) - \gamma(x)\|}[\xi(x) - \gamma(x)] + \gamma(x) & else \end{cases} \tag{13}$$

where $\xi(x)$ is the guide function of controller (6). ∎

From the preceding procedure, it is evidently that Eq. (13) is the solution of Eq. (6) with $K_V(x)$ being placed by $\overline{K_V(x)}$.

3.2 GPMN-ENMPC

In order to achieve a stable NMPC with reduced computational burden, we propose to use the GPMN to parameterize the control input sequence in NMPC. Assuming that $\vartheta(x,\theta)$ is a function of state x, where θ is the vector of unknown parameters, the following NMPC can be formulated,

$$
\begin{aligned}
&u^* = \vartheta(x, \theta^*) \\
&\theta^* = \arg\min_{\theta \subset IR^l} J(x, \theta) \\
&J(x, \theta) = \int_t^{t+T} l(x, \vartheta(x, \theta)) d\tau \\
&s.t. \quad \dot{x} = f(x) + g(x)\vartheta(x, \theta) \\
&\qquad \vartheta(x, \theta) \in U, \forall t \in [t, t+T]
\end{aligned}
\tag{14}
$$

NMPC algorithm of (14) is different from the normal NMPC in the following aspect: in normal NMPC algorithm, one tries to optimize the continuous control profile of u (Mayne et al., 2000), while controller (14) tries to achieve good performance by optimizing the parameter vector θ. Thus, the computational cost of controller (14) dependents mainly on dimension of θ instead of that of control input profile in normal NMPC algorithm. The most important problem of the latter algorithm is that its computational cost increases rapidly with the control horizon. Based on (14), our new designed NMPC controller is introduced in the following proposition.

Proposition II:
Assuming $V(x)$ is a known CLF of system (1), Ω_c is the stability region of $V(x)$, then controller (14) with the following GPMN controller $\vartheta(x, \theta)$,

$$
\vartheta(x, \theta) = u(x, \theta) = \arg\min_{u \in K_V(x)} \{\|u - \xi(x, \theta)\|\} \tag{15}
$$

($u(x,\theta)$ is the GPMN control and $\xi(x,\theta)$ the guide function in Eq. (6)), is stable in Ω_c. Furthermore, if $V(x)$ is a global CLF, controller of (14) combined with (15) is stable over R^n. (14), combined with (15), is called GPMN-Enhanced NMPC (GPMN-ENMPC).
Proof of Proposition II:
At any time instant t, by assuming that θ^* is the optimal parameters at t, control input at t can be represented as $u(x,\theta^*)$. From Proposition I, we can conclude that the control inputs $u(x,\theta^*)$ can guarantee a negative definite $\dot{V}(x)$. Due to the randomicity of t, GPMN-ENMPC actually makes the $\dot{V}(x)$ negative in any time instant, which means that the closed loop stability of controller (14) and (15) is guaranteed. ■

3.3 Selection of $\xi(x,\theta)$
Theoretically, $\xi(x,\theta)$ in (15) can be selected in any forms since it does not influence the closed loop stability, which is guaranteed by GPMN. However, it is natural that $\xi(x,\theta)$ will influence other closed loop performances of the GPMN-ENMPC except the stability.
Since optimality is the main concern in designing NMPC algorithm, the Bellman's Optimization Principle (BOP, Lewis & Syrmos, 1995) is used to design $\xi(x,\theta)$ in this sub-section.
In BOP, with the following quadratic cost function,

$$
J(x, \theta) \overset{\Delta}{=} \int_t^{t+T} (x^T P x + u^T Q u) d\tau \tag{16}
$$

and $J^*(x_0,\theta)$ denoting the optimal value function of $J(x_0,\theta)$ in state x_0, the following controller of system (1) is optimal,

$$u^* = -\frac{1}{2}(Q^{-1})^T g^T(x)\frac{\partial J^*}{\partial x} \tag{17}$$

Unfortunately, in most applications, it is impossible to obtain $J^*(x^*,\theta)$.
Based on the Stone-Weierstrass theorem (Brinkhuis & Tikhomirov, 2005), any continuous function defined in a bounded set can be uniformly approximated by a polynomial function,

$$B_k^{J^*}(x_1,\cdots,x_n) = \sum_{\substack{v_1,\cdots,v_n \geq 0 \\ v_1+\cdots+v_n \leq k}} J^*(\frac{v_1}{k},\cdots,\frac{v_n}{k})p_{k;v_1,\cdots,v_n}(x_1,\cdots,x_n) \tag{18}$$

where

$$\begin{aligned} &p_{k;v_1,\cdots,v_n}(x_1,\cdots,x_n) = \begin{pmatrix} k \\ v_1,\cdots,v_n \end{pmatrix} x_1^{v_1}\cdots x_n^{v_n}(1-x_1-\cdots-x_n)^{k-v_1-\cdots-v_n}, \\ &\begin{pmatrix} k \\ v_1,\cdots,v_n \end{pmatrix} = \frac{k!}{v_1!v_2!\cdots v_n!(k-v_1-\cdots-v_n)!}, \end{aligned} \tag{19}$$

and

$$\lim_{\substack{k_i\to\infty \\ (i=1,\cdots,k)}} B_{k_1,\cdots,k_n}^{J^*}(x_1,\cdots,x_n) = J^*(x_1,\cdots,x_n) \tag{20}$$

Thus, take the coefficients of the Bernstein polynomial as the parameters θ, and select θ optimally using the NMPC algorithm, a 'quasi-optimal' function closed to $J^*(x^*,\theta)$ can be obtained. That means we can complete the design of GPMN-ENMPC algorithm by taking

$$\xi(x,\theta) = \sum_{\substack{v_1,\cdots,v_n \geq 0 \\ v_1+\cdots+v_n \leq k}} \lambda_{v_1,\cdots,v_n} p_{k;v_1,\cdots,v_n}(x_1,\cdots,x_n) \tag{21}$$

where $\lambda_{v_1,\ldots,v_n}$, $v_1,\ldots,v_n \geq 0$ and $v_1+\ldots+v_n \leq k$ are the parameters to be optimized, k is the order of the Bernstein polynomial, and

$$\theta = [\lambda_{k_1,k_2,\ldots,k_n}]_{n^k\times 1} \tag{22}$$

It should be noted that the order of the Bernstein polynomial determines the consequent optimization cost, i.e., the higher the order is, the higher the computational cost is. About the GPMN-ENMPC, we have the following remarks:
Remark-1: Selection of $\xi(x,\theta)$ as Eq. (21) provides a feasible way to complete the GPMN-ENMPC of (20) and (21). By this way, the computation cost is controllable, namely, one can select the order of k to meet the CPU capability of a specific real system. This makes the GPMN-ENMPC feasible to be implemented.

Remark-2: The selection of k does not influence the closed loop stability, which has already guaranteed by the GPMN scheme. But there still exist trade-offs between computation cost and the optimal performance which is determined by $\xi(x,\theta)$.

Remark-3: Compared to nominal NMPC algorithm, the huge computational burden problem of GPMN-ENMPC algorithm is improved due to the following two reasons: 1) the dimension of optimizing variables is one of key elements which increase the computational burden of NMPC, while that of GPMN-ENMPC algorithm is independent of the predictive horizon; 2) online considerations of control input constraints are not necessary in GPMN-ENMPC algorithm since it can be dealt with offline during designing GPMN controller.

3.4 The Feasibility of GPMN-ENMPC

Another important problem, normally called the feasibility problem of NMPC, is that general NMPC algorithm may not guarantee that a control set always exists to meet all of the input and state constraints, while the proposed GPMN-ENMPC can guarantee such a control sequence always exists. This is because for any θ, from the proposition-I, one can always obtain a stable GPMN controller, i.e., $u(x,\theta)$ of (6) meeting all input and state constraints. Therefore, by Eq. (14) and (15), there will always exist a feasible control $u=\vartheta(x,\theta)$, and the task left is just to find an optimal parameter set of θ to minimize the cost function of $J(x,\theta)$ in Eq. (14).

4. H_∞ GPMN-ENMPC

In section 3, GPMN-ENMPC algorithm is introduced with respect to system (1). In this section, it will be generalized to deal with the disturbed system as Eq. (2). Firstly, an H_∞ controller with partially known disturbances is given, and then it is used to design H_∞GPMN controller, which followed by the designing process of H_∞GPMN-ENMPC.

4.1 H_∞ Control With Partially Known Disturbances

Suppose the following two assumptions are satisfied with respect to system (2),

***Assumption I*:**
System (2) is static feedback linearizable, i.e., there exists a state feedback controller $u=k(x)$ such that (2) can be transformed into a linear system without considering ω.

Assumption II:
The disturbances of system (2) are ***partially obtainable***, i.e., the variables ω can be used to construct controller.

Assumption II is reasonable because the uncertainty information ω can often be measured or estimated in reality (He & Han, 2007; Chen, 2004). Moreover, the tracking problem of general nonlinear system, where ω is composed of known desired trajectory, can also be modeled as Eq. (2). However, the higher order derivative of the disturbances with respective to time is often difficult to be obtained due to the heavy additive noise. Thus, the disturbances are often '*partially obtainable*'.

Based on assumption I, system (2) can be changed into the following equations through some coordination transformation,

$$\begin{aligned} \dot{z}_1 &= z_2 + F_1(z)\Delta \\ &\vdots \\ \dot{z}_n &= f_1(z) + g_1(z)u + F_n(z)\Delta \\ y &= z_1 \end{aligned} \tag{23}$$

where $z = [z_1, z_2, \ldots, z_n]^T$ is the new state variable.

An H_∞ robust controller for system (23) can be designed based on the following Theorem,

Theorem I**:**

Consider system (23), if there exists a control $u = u_1(z)$ and a radially unbounded function $V(x)$ to satisfy the following inequality,

$$\begin{aligned} &\sum_{i=1}^{n-1} V_{z_i} z_{i+1} + V_{z_n}[f_1(z) + g_1(z)u_1(z)] + \frac{2}{\gamma^2} V_z \begin{bmatrix} F_1^T(z) & F_2^T(z) & \cdots & F_n^T(z) \end{bmatrix}^T \times \\ &\begin{bmatrix} F_1^T(z) & F_2^T(z) & \cdots & F_n^T(z) \end{bmatrix} V_z^T + z_1^2 \le 0 \end{aligned} \tag{24}$$

Then, controller

$$\begin{aligned} u = {} & g_1^{-1}(z)[f_1(\bar{z}) + g_1(\bar{z})u_1 - f_1(z) + \sum_{i=1}^{n} \bar{F}_2(\bar{z}, \rho, \cdots, \rho^{(i-1)})\rho^{(n-i)}] + [\bar{V}_{\bar{z}_n} g_1(z)]^{-1} \times \\ & \{\frac{2}{\gamma^2} \bar{V}_{\bar{z}} \begin{bmatrix} F_1(\bar{z}) - \bar{F}_1(\bar{z}, \rho) \\ F_2(\bar{z}) - \bar{F}_2(\bar{z}, \rho, \dot{\rho}) \\ \vdots \\ F_n(\bar{z}) - \bar{F}_n(\bar{z}, \rho, \cdots, \rho^{(n-1)}) \end{bmatrix} \begin{bmatrix} F_1(\bar{z}) - \bar{F}_1(\bar{z}, \rho) \\ F_2(\bar{z}) - \bar{F}_2(\bar{z}, \rho, \dot{\rho}) \\ \vdots \\ F_n(\bar{z}) - \bar{F}_n(\bar{z}, \rho, \cdots, \rho^{(n-1)}) \end{bmatrix}^T \bar{V}_{\bar{z}}^T \} \end{aligned} \tag{25}$$

can make the system (23) finite gain L_2 stable from $\Delta+\rho$ to y, and the gain is less than or equal to γ. ρ is a new defined signal to further attenuate the disturbances.

Proof of Theorem I:

Define new variables,

$$\begin{aligned} \bar{z}_1 &= z_1 \\ \bar{z}_2 &= z_2 - F_1(z)\rho \\ &\vdots \\ \bar{z}_n &= z_n - \sum_{i=1}^{n-1} F_i(z)\rho^{(n-i)} \end{aligned} \tag{26}$$

Then, system (23) can be written as

$$\begin{aligned}
&\dot{\bar{z}}_1 = \bar{z}_2 + \bar{F}_1(\bar{z},\rho)(\Delta+\rho)\\
&\dot{\bar{z}}_2 = \bar{z}_3 + \bar{F}_2(\bar{z},\rho,\dot{\rho})(\Delta+\rho)\\
&\quad\vdots\\
&\dot{\bar{z}}_n = f_1(z) + g_1(z)u - \sum_{i=1}^{n}\bar{F}_i(\bar{z},\rho,\cdots,\rho^{(i-1)})\rho^{(n-i)} + \bar{F}_n(\bar{z},\rho,\cdots,\rho^{(n-1)})(\Delta+\rho)\\
&y = \bar{z}_1
\end{aligned} \tag{27}$$

where

$$\bar{F}_j(\bar{z},\rho,\cdots,\rho^{(j-1)}) \triangleq F_j(z)\Big|_{z_i=\bar{z}_j+\sum_{j=1}^{i-1}F_i(z)\rho^{(n-i)}}$$

Let

$$\bar{V}(\bar{z}) = V(z)\,|_{z=\bar{z}} \tag{28}$$

where $\bar{z} = \begin{bmatrix}\bar{z}_1 & \bar{z}_2 & \cdots & \bar{z}_n\end{bmatrix}^T$. Computing the HJI equation (Khalil, 2002) of system (27) with respect to $\bar{V}(\bar{z})$, we have,

$$\begin{aligned}
&\sum_{i=1}^{n-1}\bar{V}_{z_i}\bar{z}_{i+1} + \bar{V}_{\bar{z}_n}[f_1(z)+g_1(z)u - \sum_{i=1}^{n}\bar{F}_2(\bar{z},\rho,\cdots,\rho^{(i-1)})\rho^{(n-i)}] +\\
&\frac{2}{\gamma^2}\bar{V}_{\bar{z}}\begin{bmatrix}\bar{F}_1^T(\bar{z},\rho) & \bar{F}_2^T(\bar{z},\rho,\dot{\rho}) & \cdots & \bar{F}_n(\bar{z},\rho,\cdots,\rho^{(n-1)})\end{bmatrix}^T \times\\
&\begin{bmatrix}\bar{F}_1^T(\bar{z},\rho) & \bar{F}_2^T(\bar{z},\rho,\dot{\rho}) & \cdots & \bar{F}_n(\bar{z},\rho,\cdots,\rho^{(n-1)})\end{bmatrix}\bar{V}_{\bar{z}}^T + \tilde{z}_1^2
\end{aligned} \tag{29}$$

Thus, combine controller (25) and Eq. (29), we have,

$$\begin{aligned}
(29) = &\sum_{i=1}^{n-1}\bar{V}_{z_i}\bar{z}_{i+1} + \bar{V}_{\bar{z}_n}[f_1(z)+g_1(z)u - \sum_{i=1}^{n}\bar{F}_2(\bar{z},\rho,\cdots,\rho^{(i-1)})\rho^{(n-i)}] +\\
&\frac{2}{\gamma^2}\bar{V}_{\bar{z}}\begin{bmatrix}\bar{F}_1^T(\bar{z},\rho) & \bar{F}_2^T(\bar{z},\rho,\dot{\rho}) & \cdots & \bar{F}_n^T(\bar{z},\rho,\cdots,\rho^{(n-1)})\end{bmatrix}^T \times\\
&\begin{bmatrix}\bar{F}_1^T(\bar{z},\rho) & \bar{F}_2^T(\bar{z},\rho,\dot{\rho}) & \cdots & \bar{F}_n^T(\bar{z},\rho,\cdots,\rho^{(n-1)})\end{bmatrix}\bar{V}_{\bar{z}}^T + \bar{z}_1^2\\
= &\{\sum_{i=1}^{n-1}V_{z_i}z_{i+1} + V_{\bar{z}_n}[f_1(z)+g_1(z)u] + \frac{2}{\gamma^2}V_z\begin{bmatrix}F_1^T(z) & F_2^T(z) & \cdots & F_n^T(z)\end{bmatrix}^T \times\\
&\begin{bmatrix}F_1^T(z) & F_2^T(z) & \cdots & F_n^T(z)\end{bmatrix}V_z^T + z_1^2\}\,|_{z=\bar{z}}\\
\le &\; 0
\end{aligned} \tag{30}$$

Based on theorem 5.5 in reference (Khalil, 2002), controller (25) can make system (23) finite gain L_2 stable from $\Delta+\rho$ to y, and the L_2 gain is less than or equal to γ. ∎

Furthermore, ρ can be used to further attenuate the disturbances which are partially obtainable from assumption II by the following equation,

$$\rho(s) = \frac{B(s)}{A(s)}\Delta(s) \tag{31}$$

where s is the Laplace operator. Thus, the new external disturbances $\Delta+\rho$ can be denoted as,

$$\Delta(s) + \rho(s) = \frac{A(s)+B(s)}{A(s)}\Delta(s) \tag{32}$$

From Eq. (32), proper A(s) and B(s) is effective for attenuating the influence of external disturbances on the closed loop system. Thus, we have designed an H_∞ controller (25) and (31) with partially known uncertainty information.

4.2 H_∞ GPMN Controller Based on Control Lyapunov Functions

In this sub-section, by using the concept of H_∞CLF, H_∞ GPMN controller is designed as following proposition,

***Proposition III**:*

If $V(x)$ is a local H_∞CLF of system (23), and $\xi(x)$: $R^n \rightarrow R^m$ is a continuous guide function such that $\xi(0) = 0$, then, the following controller, called H_∞GPMN, can make system (23) finite gain L_2 stable from Δ to output $\boldsymbol{y}$,

$$u^{H\infty}(x) = \arg\min_{u \in K_V^{H\infty}(x)} \{\|u - \xi(x)\|\} \tag{33}$$

where

$$K_V^{H_\infty}(x) = \{u \in U(x) : V_x[f(x) + g(x)u] + \frac{1}{2\gamma^2}V_x l(x) l^T(x) V_x^T + \frac{1}{2}h^T(x)h(x) \le -\sigma(x)\} \tag{34}$$

■

Proof of Proposition III can be easily done based on the definition of finite gain L_2 stability and H_∞CLF. The analytical form of controller (33) can also be obtained as steps in section 3. Here only the analytical form of controller without input constraints is given,

$$u^{H\infty}(x) = \begin{cases} \xi - \dfrac{[V_x(f + \frac{1}{2\gamma^2} ll^T V_x^T + g\xi) + \frac{1}{2}h^T h + \sigma]g^T V_x^T}{V_x g g^T V_x^T} & \lambda > 0 \\ \xi & \lambda \le 0 \end{cases} \tag{35}$$

where

$$\lambda = V_x f + \sigma + V_x g\xi; \quad f = f(x); \quad g = g(x); \quad \xi = \xi(x);$$
$$\sigma = \sigma(x); \quad V_x = V_x(x); \quad h = h(x); \quad l = l(x)$$

It is not difficult to show that H∞GPMN satisfies inequality (24) of Theorem I, thus, it can be used as $u_1(z)$ in controller (25) to bring the advantages of H∞GPMN controller to the robust controller in section 4.1.

4.3 H∞GPMN-ENMPC

As far as the external disturbances are concerned, nominal model based NMPC, where the prediction is made through a nominal certain system model, is an often used strategy in reality. And the formulation of it is very similar to non-robust NMPC, so dose the GPMN-ENMPC.

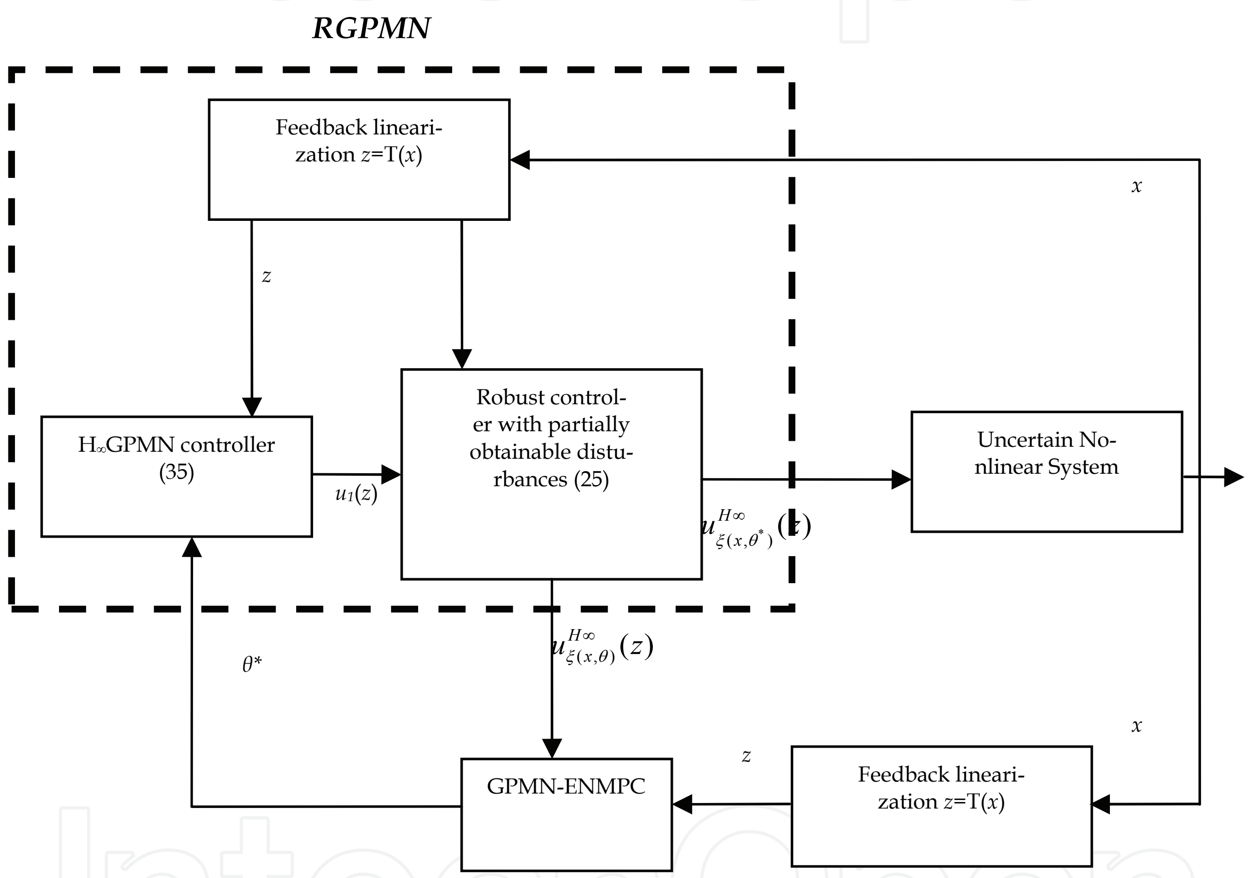


Fig. 4. Structure of new designed RNRHC controller

However, for disturbed nonlinear system like Eq. (23), GPMN-ENMPC algorithm can hardly be used in real applications due to weak robustness. Thus, in this subsection, we will combine it to the robust controller from sub-section 4.1 and sub-section 4.2 to overcome the drawbacks originated from both GPMN-ENMPC algorithm and the robust controller (25) and (35). The structure of the new parameterized H∞GPMN-ENMPC algorithm based on controller (25) and (35) is as Fig. 4.

Eq. (36) is the new designed H∞GPMN-ENMPC algorithm. Compared to Eq. (14), it is easy to find out that the control input in the H∞GPMN-ENMPC algorithm has a pre-defined structure given in section 4.1 and 4.2.

$$
\begin{aligned}
&u^* = u^{H\infty}(x,\theta^*) \\
&\theta^* \overset{\Delta}{=} \arg\min_{u\in U} J(x,u) \\
&J(x,u) \overset{\Delta}{=} \int_t^{t+T} l(x(\tau),u(\tau))d\tau \\
&s.t. \quad \dot{x} = f(x) + g(x)u \\
&\qquad\ \ u(t) = u^{H\infty}(x,\theta)
\end{aligned}
\tag{36}
$$

5. Practical Considering

Both GPMN-ENMPC algorithm and H_∞GPMN-ENMPC algorithm can be divided into two processes, including the implementation process and the optimization process as Fig.5.

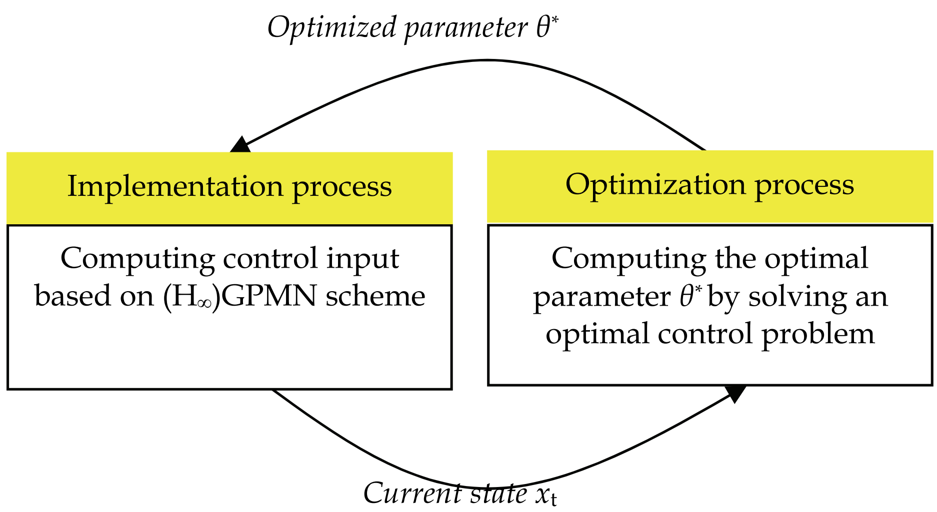


Fig. 5. The process of (H_∞)GPMN-ENMPC

The implementation process and the optimization process in Fig. 5 are independent. In implementation process, the (H_∞)GPMN scheme is used to ensure the closed loop (L_2) stability, and in the optimization process, the optimization algorithm is responsible to improving the optimality of the controller. And the interaction of the two processes is realized through the optimized parameter θ^* (from optimization process to implementation process) and the measured states (form implementation process to optimization process).

5.1 Time Interval Between Two Neighboring Optimizing Process

Sample time in controller implemented using computer is often very short, especially in mechatronic system. This is very challenging to implement complicated algorithm, such as GPMN-ENMPC in this chapter. Fortunately, the optimization process of the new designed controller will end up with a group of parameters which are used to form a stable (H_∞)GPMN controller, and the optimization process itself does not influence the closed loop stability at all. Thus, theoretically, any group of optimized parameters can be used for several sample intervals without destroying the closed loop stability.

Fig.6 denotes the scheduling of (H_∞)GPMN-ENMPC algorithm. In Fig.6, t is the current time instant; T is the prediction horizon; T_S is the sample time of the (H_∞)GPMN controller; and T_I is the duration of every optimal parameter $\theta^*(t)$, i.e., the same parameter θ^* is used to implement the (H_∞)GPMN controller from time t to time $t+T_I$.

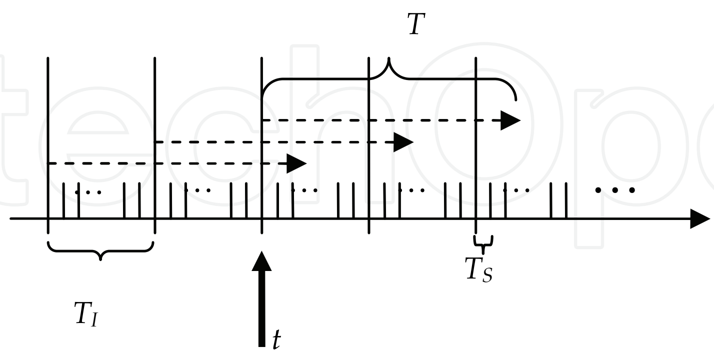


Fig. 6. Scheduling of ERNRHC

5.2 Numerical Integrator

How to predict the future's behavior is very important during the implementation of any kind of MPC algorithms. In most applications, the NMPC algorithm is realized by computers. Thus, for the continuous systems, it will be difficult and time consuming if some accurate but complicated numerical integration methods are used, such as Newton-Cotes integration and Gaussian quadratures, etc. In this chapter, we will discretize the continuous system (1) as follows (take system (1) as an example),

$$x(kT_O + T_O) = f(x(kT_O))T_O + g(x(kT_O))u(kT_O)T_O \tag{37}$$

where T_o is the discrete sample time. Thus, the numerical integrator can be approached by the operation of cumulative addition.

5.3 Index Function

Replace $x(kT_o)$ with $x(k)$, the index function can be designed as follows,

$$J(x(k_0),\theta_c) \stackrel{\Delta}{=} \theta_l^{T*} Z \theta_l^* + \sum_{i=k_0}^{k_0+N} J(x(i),\theta_c) \tag{38}$$

where k_0 denotes the current time instant; N is the predictive horizon with N=Int(T/T_o) (here Int(*) is the operator to obtain a integer nearest to *); θ_c is the parameter vector to be optimized at current time instant; and θ_l^* is the last optimization result; Q, Z, R are constant matrix with Q>0, Z>0, and R≥0.

The new designed item $\theta_l^{T*} Z \theta_l^*$ is used to reduce the difference between two neighboring optimized parameter vector, and improve the smoothness of the optimized control inputs u.

6. Numerical Examples

6.1 Example1 (GPMN-ENMPC without control input constrains)

Consider the following pendulum equation (Costa & do Va, 2003),

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \dfrac{19.6\sin x_1 - 0.2x_2^2 \sin 2x_1}{4/3 - 0.2\cos^2 x_1} + \dfrac{-0.2\cos x_1}{4/3 - 0.2\cos^2 x_1} u \end{cases} \tag{39}$$

A local CLF of system (39) can be given as,

$$V(x) = x^T P x = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 151.57 & 42.36 \\ 42.36 & 12.96 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{40}$$

Select

$$\sigma(x) = 0.1(x_1^2 + x_2^2) \tag{41}$$

The normal PMN control can be designed according to (5) as,

$$u = \begin{cases} -\dfrac{\rho(x)(4/3 - 0.2\cos^2 x_1)}{0.4\cos x_1 (42.36x_1 + 12.96x_2)} & \rho(x) < 0 \\ 0 & \rho(x) \le 0 \end{cases}$$

$$\rho(x) = 2[(151.57x_1 + 42.36x_2)x_2 + (42.36x_1 + 12.96x_2)\frac{19.6\sin x_1 - 0.2x_2^2 \sin 2x_1}{4/3 - 0.2\cos^2 x_1}] + (10.54x_2 + 1.27x_1)^2 + x_2^2 \tag{42}$$

Given initial state $x_0 = [x_1, x_2]^T = [-1,2]^T$, and desired state $x_d = [0,0]^T$, time response of the closed loop for PMN controller is shown in Fig. 7 in solid line. It can be seen that the closed loop with PMN controller (42) has a very low convergence rate for state x_1. This is mainly because the only regulable parameter to change the closed loop performance is $\sigma(x)$, which is difficult to be properly selected due to its great influence on the stability region.

To design GPMN-ENMPC, two different guide functions are selected based on Eq. (21),

$$\xi(x,\theta) = \theta_{0,0}(1 - x_1 - x_2) + \theta_{1,0}x_1 + \theta_{0,1}x_2 \tag{43}$$

$$\xi(x,\theta) = \theta_{0,0}(1 - x_1 - x_2)^2 + 2(\theta_{0,1}x_2 + \theta_{1,0}x_1)(1 - x_1 - x_2) + 2\theta_{1,1}x_1x_2 + \theta_{0,2}x_2^2 + \theta_{2,0}x_1^2 \tag{44}$$

CLF $V(x)$ and $\sigma(x)$ are given in Eq. (40) and Eq. (41), and others conditions in GPMN-ENMPC are designed as follows,

$$J = \int_0^T (x^T \begin{bmatrix} 20 & 0 \\ 0 & 1 \end{bmatrix} x + 0.01u^2) dt \tag{45}$$

$$l(x,u)=x^T\begin{bmatrix}20 & 0\\ 0 & 1\end{bmatrix}x+0.01u^2;\ f(x)=\begin{bmatrix}x_2\\ \dfrac{19.6\sin x_1-0.2x_2^2\sin 2x_1}{4/3-0.2\cos^2 x_1}\end{bmatrix};$$
$$g(x)=\begin{bmatrix}0\\ \dfrac{-0.2\cos x_1}{4/3-0.2\cos^2 x_1}\end{bmatrix};z=0.1I \tag{46}$$

Integral time interval T_o in Eq. (37) is 0.1s. Genetic algorithm (GA) in MATLAB toolbox is used to solve the online optimization problem. Time response of GPMN-ENMPC algorithm with different predictive horizon T and approaching order are presented in Fig. 7, where the dotted line denotes the case of T = 0.6s with guide function (43), and the dashed line is the case of T = 1.5s with guide function (44). From Fig. 7, it can be seen that the convergence performance of the proposed NMPC algorithm is better than 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 others controller. In view of this, we propose to estimate the optimality by the following index function,

$$J=\lim_{\Gamma\to\infty}\int_0^{\Gamma}(x^T\begin{bmatrix}20 & 0\\ 0 & 1\end{bmatrix}x+0.01u^2)dt \tag{47}$$

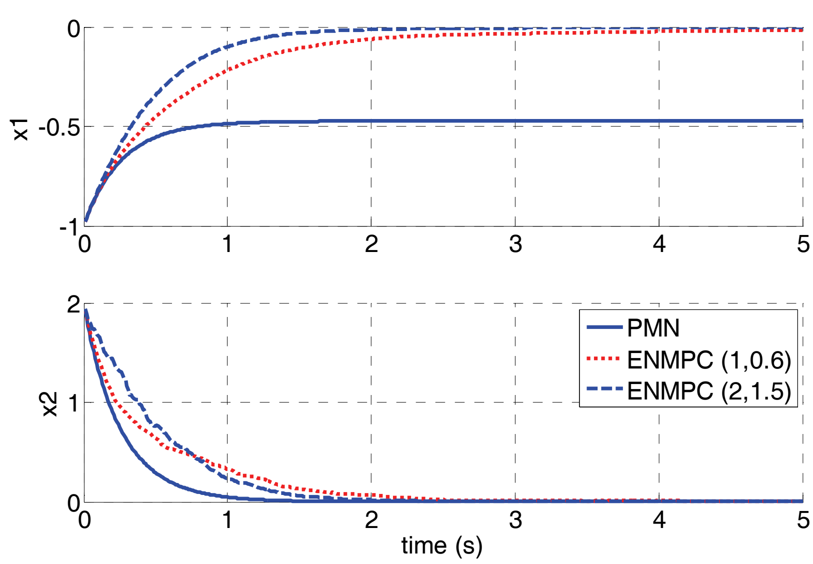


Fig. 7. Time response of different controller, where the (a,b) indicates that the order of $\xi(x,\theta)$ is a, and the predictive horizon b

The comparison results are summarized in Table 1, from which the following conclusions can be obtained, 1) GPMN-ENMPC has better optimizing performance than 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. 3) A longer prediction horizon will usually be followed by a better optimal performance.

J	ENMPC				PMN	
	x_0 = (-1,2)		x_0 = (0.5,1)		x_0 = (-1,2)	x_0 = (0.5,1)
	k = 1	k = 2	K = 1	k = 2	----	
T=0.6	*29.39*	*28.87*	*6.54*	*6.26*	→+∞	→+∞
T=0.8	*23.97*	*23.83*	*5.02*	*4.96*	→+∞	→+∞
T=1.0	*24.08*	*24.07*	*4.96*	*4.90*	→+∞	→+∞
T=1.5	*26.31*	*24.79*	*5.11*	*5.28*	→+∞	→+∞

Table 1. the cost value of different controller
* k is the order of Bernstein polynomial used to approach the optimal value function; T is the predictive horizon; x_0 is the initial state

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 (the smaller these values are selected, the less the computational cost is). However, it will be natural that the optimality maybe deteriorated to some extent with the decreasing of the computational burden. In preceding paragraphs, we have researched the optimality of GPMN-ENMPC algorithm with different optimizing parameters, and now the optimality comparisons among the closed loop systems with different GA parameters will be done. And the results are listed in Table 2, from which the certain of 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.

OP	G=100 PS=50	G=50 PS=50	G=50 PS=30	G=50 PS=20	G=50 PS=10
cost	26.2	28.1	30.8	43.5	45.7

Table 2. The relation between the computational cost and the optimality
*x_0 = (-1,2), T=*1.5*, k = 1, OP means Optimization Prameters, G means Generations, PS means Population Size

Finally, in order to verify that the new designed algorithm is improved in the computational burden, simulations comparing the performance of the new designed algorithm and algorithm in (Primbs, 1999) are conducted with the same optimizing algorithm. Time interval of two neighboured optimization (T_I in Table 3) in Primbs' algorithm is important since control input is assumed to be constant at every time slice. Generally, large time interval will result in poor stability.

While our new GPMN-ENMPC results in a group of controller parameter, and the closed loop stability is independent of T_I. Thus different T_I is considered in these simulations of Primbs' algorithm and Table 3 lists the results. From Table 3, the following items can be concluded: 1) with same GA parameters, Primbs' algorithm is more time-consuming and poorer in optimality than GPMN-ENMPC. This is easy to be obtained through comparing results of Ex-2 and Ex-5; 2) in order to obtain similar optimality, GPMN-ENMPC takes much less time than Primbs' algorithm. This can be obtained by comparing 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 3***.

	Algorithm in (Primbs, 1999)				GPMN-ENMPC	
	Ex-1	Ex-2	Ex-3	Ex-4	Ex-5	Ex-6
TI	0.1		0.05		0.1	
OP	G=100 PS=50	G=50 PS=50	G=100 PS=50	G=50 PS=50	G=50 PS=50	G=50 PS=30
Average Time Consumption	2.2075	1.8027	2.9910	2.2463	1.3961	0.8557
Cost	31.2896	35.7534	27.7303	31.8055	28.1	31.1043

Table 3. Performance comparison of GPMN-ENMPC and Primbs' algorithm
*x_0 = (-1,2), TI means time interval of two neighbored optimization; OP means Optimization Prameters; G means Generations, PS means Population Size. Other parameters of GPMN-ENMPC are $T=1.5$, $k = 1$

6.2 Example 2 (GPMN-ENMPC with control input constraint)

In order to show the performance of the GPMN-ENMPC in handling input constraints, we give another simulation using the dynamics of a mobile robot with orthogonal wheel assemblies (Song, 2007). The dynamics can be denoted as Eq. (48),

$$\dot{x} = f(x) + g(x)u \tag{48}$$

where

$$f(x) = \begin{bmatrix} x_2 \\ -2.3684x_4x_6 - 0.5921x_2 \\ x_4 \\ 2.3684x_2x_6 - 0.5921x_4 \\ x_6 \\ -0.2602x_6 \end{bmatrix}$$

$$g(x) = \begin{bmatrix} 0 & 0 & 0 \\ 0.8772(\sqrt{3}\sin x_5 - \cos x_5) & 0.8772*2\cos x_5 & 0.8772(\sqrt{3}\sin x_5 - \cos x_5) \\ 0 & 0 & 0 \\ 0.8772(\sqrt{3}\cos x_5 - \sin x_5) & 0.8772*2\sin x_5 & 0.8772(-\sqrt{3}\cos x_5 - \sin x_5) \\ 0 & 0 & 0 \\ -1.4113 & -1.4113 & -1.4113 \end{bmatrix}$$

$x_1 = x_w; x_2 = \dot{x}_w; x_3 = y_w; x_4 = \dot{y}_w; x_5 = \varphi_w; x_6 = \dot{\varphi}_w$; x_w, y_w, φ_w are respective the x-y positions and yaw angle; u_1, u_2, u_3 are motor torques.
Suppose that control input is limited in the following closed set,

$$U = \{(u_1, u_2, u_3) \,|\, (u_1^2 + u_2^2 + u_3^2)^{1/2} \le 20\} \tag{49}$$

System (48) is feedback linearizable, and by which we can obtain a CLF of system (48) as follows,

$$V(x) = x^T P x \tag{50}$$

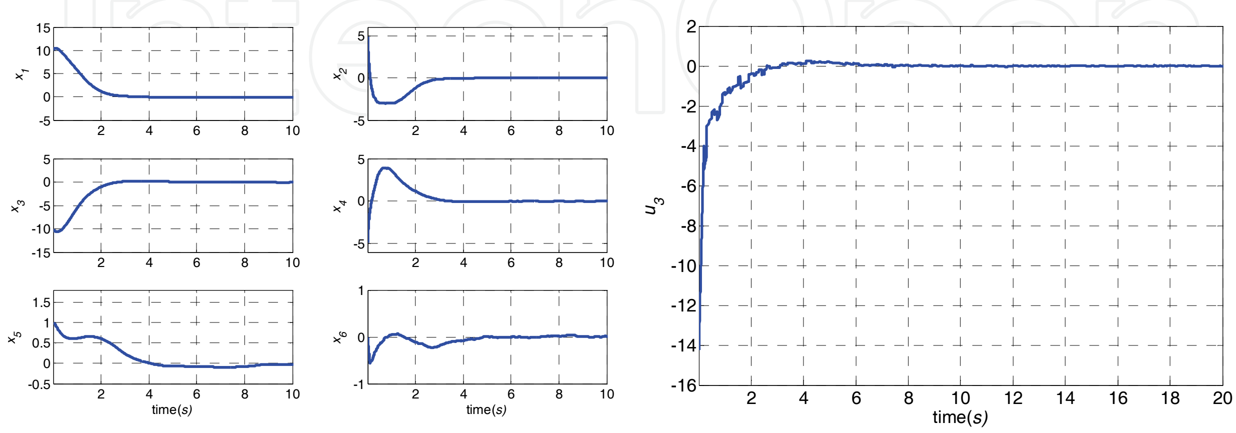
where

$$P = \begin{bmatrix} 1.125 & 0.125 & 0 & 0 & 0 & 0 \\ 0.125 & 0.156 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.125 & 0.125 & 0 & 0 \\ 0 & 0 & 0.125 & 0.156 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.125 & 0.125 \\ 0 & 0 & 0 & 0 & 0.125 & 1.156 \end{bmatrix}$$

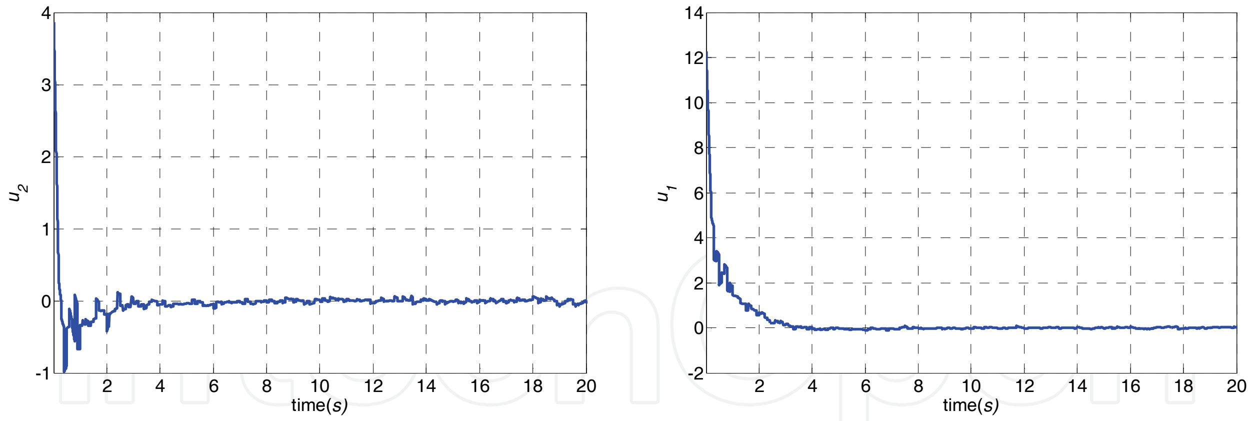
The cost function $J(x)$ and $\sigma(x)$ are designed as,

$$\begin{aligned} J(x) &= \int_{t_0}^{t_0+T} (3x_1^2 + 3x_3^2 + 3x_5^2 + x_2^2 + x_4^2 + x_6^2 + 5u_1^2 + 5u_2^2 + 5u_3^2)dt + \theta^T(k-1)Z\theta(k-1); \\ \sigma(x) &= 0.1(x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2); Z = 0.1I \end{aligned} \tag{51}$$

System (48) has 6 states and 3 inputs, which will introduce large computational burden if using the GPMN-ENMPC method. Fortunately, one of the advantages of GPMN-ENMPC is that the optimization does not destroy the closed loop stability. Thus, in order to reduce the computation burden, we reduce the frequency of the optimization in this simulation, i.e., one optimization process is conducted every 0.1s while the controller of (13) is calculated every 0.002s, i.e., $T_I = 0.1$s, $T_s = 0.002$s.



a) states response b) control input u_1



c) control input u_2 d) control input u_3

Fig. 8. GPMN-ENMPC controller simulation results on the mobile robot with input constraints

Initial States (x_1; x_2; x_3; x_4; x_5; x_6)	Feedback linearization controller	GPMN-NMPC
(10; 5; 10; 5; 1; 0)	2661.7	1377.0
(10; 5; 10; 5; -1; 0)	3619.5	1345.5
(-10; -5; 10; 5; 1; 0)	2784.9	1388.5
(-10; -5; 10; 5; -1; 0)	8429.2	1412.0
(-10; -5; -10; -5; 1; 0)	394970.0	1349.9
(-10; -5; -10; -5; -1; 0)	4181.6	1370.9
(10; 5; -10; -5; 1; 0)	3322	1406
(10; 5; -10; -5; -1; 0)	1574500000	1452.1
(-5; -2; -10; -5; 1; 0)	1411.2	856.1
(-10; -5; -5; -2; 1; 0)	1547.5	850.9

Table 4. The comparison of the optimality

Simulation results are shown in Fig.8 with the initial state (10; 5; -10; -5; 1; 0), From Fig.8, it is clear that GPMN-ENMPC controller has the ability to handling input constraints.
In order to evaluate the optimal performance of the GPMN-ENMPC, we proposed the following cost function according to Eq. (51),

$$\cos t = \lim_{\Gamma \to +\infty} \int_0^{\Gamma} (3x_1^2 + 3x_3^2 + 3x_5^2 + x_2^2 + x_4^2 + x_6^2 + 5u_1^2 + 5u_2^2 + 5u_3^2) dt \quad (52)$$

Table 4 lists the costs by feedback linearization controller and GPMN-ENMPC for several different initial states, from which it can be seen that the cost of GPMN-ENMPC is less than the half of the cost of feedback linearization controller when the initial is (10; 5; -10; -5; 1; 0). And in most cases listed in Table 4, the cost of GPMN-ENMPC is about one second of that of feedback linearization controller. Actually, in some special cases, such as the initial of (10; 5; -10; -5; -1; 0), the cost ratio of feedback linearization controller to GPMN-ENMPC is more than 1000000.

6.3 Example 3 (H_∞GPMN-ENMPC)

In this section, a simulation will be given to verify the feasibility of the proposed H_∞GPMN-ENMPC algorithm with respect to the following planar dynamic model of helicopter,

$$\begin{cases} \ddot{x} = -9.8\cos\phi\sin\theta + \Delta_1 \\ \ddot{y} = 9.8\sin\phi + \Delta_2 \\ \ddot{\phi} = 0.05\dot{\theta}^2 \sin\phi\cos\phi + \dot{\phi}\dot{\theta}\tan\theta(0.5+0.05\cos^2\phi) + \\ \qquad 0.07\cos\phi\tan\theta + L + M\sin\phi\tan\theta + \Delta_3 \\ \ddot{\theta} = -0.05\dot{\phi}\dot{\theta}\sin\phi\cos\phi - 0.07\sin\phi + M\cos\phi + \Delta_4 \end{cases} \tag{53}$$

where Δ_1, Δ_2, Δ_3, Δ_4 are all the external disturbances, and are selected as following values,

$$\begin{cases} \Delta_1 = 3; \quad \Delta_2 = 3 \\ \Delta_3 = 10\sin(0.5t) \\ \Delta_4 = 10\sin(0.5t) \end{cases}$$

Firstly, design an H_∞CLF of system (53) by using the feedback linearization method,

$$V = X^T \overline{P} X \tag{54}$$

where,

$$X = [x, \dot{x}, \ddot{x}, \dddot{x}, y, \dot{y}, \ddot{y}, \dddot{y}]^T$$

$$\overline{P} = \begin{bmatrix} 14.48 & 11.45 & 3.99 & 0.74 & 0 & 0 & 0 & 0 \\ 11.45 & 9.77 & 3.44 & 0.66 & 0 & 0 & 0 & 0 \\ 3.99 & 3.44 & 1.28 & 0.24 & 0 & 0 & 0 & 0 \\ 0.74 & 0.66 & 0.24 & 0.05 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 14.48 & 11.45 & 3.99 & 0.74 \\ 0 & 0 & 0 & 0 & 11.45 & 9.77 & 3.44 & 0.66 \\ 0 & 0 & 0 & 0 & 3.99 & 3.44 & 1.28 & 0.24 \\ 0 & 0 & 0 & 0 & 0.74 & 0.66 & 0.24 & 0.05 \end{bmatrix}$$

Thus, the robust predictive controller can be designed as Eq. (25), (35) and (36) with the following parameters,

$$\sigma(x) = X^T X$$

$$J = \theta_l^{T*} I \theta_l^* + \sum_{i=1}^{N} [x^{\mathrm{T}}(iT_o) P x(iT_o) + u^{\mathrm{T}}(iT_o) Q u(iT_o)] T_o$$

$$\xi(x,\beta) = \begin{bmatrix} \beta_1 + \beta_2 x + \beta_3 \dot{x} + \beta_4 y + \beta_5 \dot{y} + \beta_6 \phi + \beta_7 \dot{\phi} + \beta_8 \theta + \beta_9 \dot{\theta} \\ \beta_{10} + \beta_{11} x + \beta_{12} \dot{x} + \beta_{13} y + \beta_{14} \dot{y} + \beta_{15} \phi + \beta_{16} \dot{\phi} + \beta_{17} \theta + \beta_{18} \dot{\theta} \end{bmatrix}$$

$$P = \begin{bmatrix} 50000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 50000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; T_o = 0.1s;\ T_s = 0.02s;\ T_I = 1s;\ N = 20;\ Z = I$$

Time response of the H_∞GPMN-ENMPC is as solid line of Fig.9 and Fig.10. Furthermore, the comparisons between the performance of the closed loop controlled by the proposed H_∞GPMN-ENMPC and some other controller design method are done. The dashed line in Fig.9 and Fig.10 is the time response of the feedback linearization controller. From Fig.9 and Fig.10, the disturbance attenuation performance of the H_∞GPMN-ENMPC is apparently better than that of feedback linearization controller, because the penalty gain of position signals, being much larger than other terms, can be used to further improve the ability.

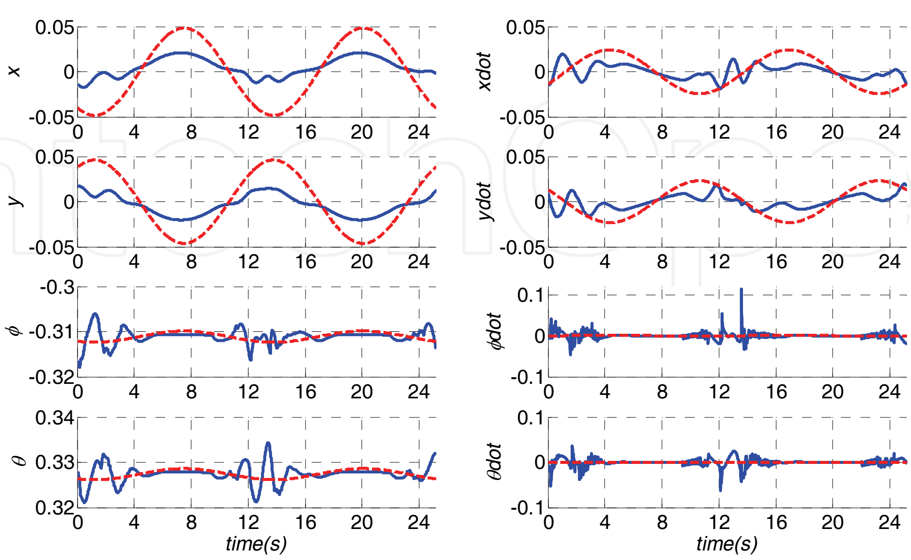


Fig. 9. Time response of states

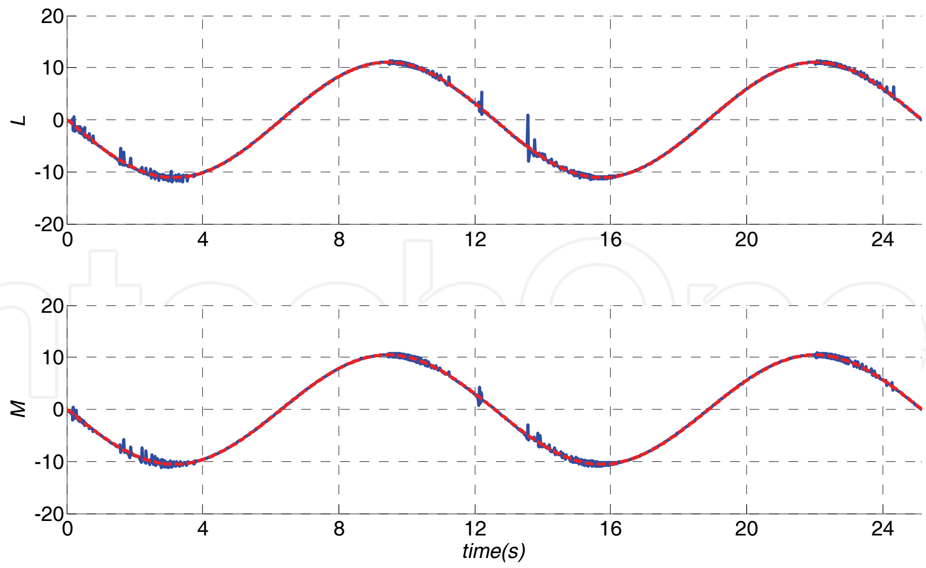


Fig. 10. Control inputs

Simultaneously, the following index is used to compare the optimality of the two different controllers,

$$J = \lim_{\Gamma \to \infty} \int_0^{0+\Gamma} [x^{\mathrm{T}}(t)Px(t) + u^{\mathrm{T}}(t)Qu(t)]dt \tag{55}$$

The optimality performance of H_∞GPMN-ENMPC, computed from Eq. (55), is about 3280, and the feedback linearization controller is about 5741, i.e., the H_∞GPMN-ENMPC has better optimality than the feedback linearization controller.

7. Conclusion

In this paper, nonlinear model predictive control (NMPC) is researched and a new NMPC algorithm is proposed. The new 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. Also, the new GPMN-ENMPC is generalized to a robust version with respect to input-output feedback linearizable nonlinear system with partially known uncertainties. Finally, extensive simulations have been conducted, and the results have shown the feasibility and validity of the new designed method.

8. References

Brinkhuis, J. & Tikhomirov, V., *Optimization : insights and applications*, Princeton University Press, ISBN : 978-0-691-10287-0, Oxfordshire, United Kingdom

Chen, C. & Shaw, L., On receding horizon feedback control, *Automatica*, Vol. 18, No. 3, May, 1982, 349-352, ISSN : 0005-1098

Chen, H. & Allgower, F., A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, Vol. 34, No. 10, Oct, 1998, 1205-1217, ISSN : 0005-1098

Chen, W., Disturbance observer based control for nonlinear systems, *IEEE/ASME Transactions on mechatronics*, Vol. 9, No. 4, 2004, 706-710, ISSN : 1083-4435

Costa, E. & do Val, J., Stability of receding horizon control of nonlinear systems, *Proceedings of The 42nd IEEE Conference on Decision and Control*, pp. 2077-2801, ISSN : 0191-2216, HI, USA, Dec, 2003, IEEE, Maui

Freeman, R. & Kokotovic, P, Inverse optimality in robust stabilization, *SIAM Journal on Control and Optimization*, Vol. 34, No. 4, Aug, 1996a, 1365-1391, ISSN : 0363-0129

Freeman, R. & Kokotovic, P.(1996b), *Robust nonlinear control design: state-space and Lyapunov techniques*, Birkhauser, ISBN: 978-0-8176-4758-2, Boston

Henson, M., Nonlinear model predictive control: current status and future directions. *Computers & Chemical Engineering*, Vol. 23, No. 2, Dec, 1998, 187-202, ISSN : 0098-1354

He, Y., and Han, J., Acceleration feedback enhanced H_∞ disturbance attenuation control, *Proceedings of The 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON)*, pp. 839-844, ISSN : 1553-572X, Taiwan, Nov, 2007, IEEE, Taipei

Khalil, H. (2002), *Nonlinear systems*, 3rd edition, Printice Hall, ISBN: 0-13-067389-7, NJ, USA

Lewis, F. & Syrmos, V. (1995), *Optimal control*, John Wiley & Sons, ISBN : 0-471-03378-2, Bangalore, India

Magni, L., De Nicolao, G., Magnani, L. & Scattolini, R., A stabilizing model-based predictive control algorithm for nonlinear systems. *Automatica*, Vol. 37, No. 9, Sep, 2001, 1351-1362, ISSN : 0098-1354

Mayne, D., Rawlings, J., Rao, C. & Scokaert, P., Constrained model predictive control: stability and optimality. *Automatica*, Vol. 36, No. 6, Jun, 2000, 789-814, ISSN : 0005-1098

Pothin, R., Disturbance decoupling for a class of nonlinear MIMO systems by static measurement feedback, *Systems & Control Letters*, Vol. 43, No. 2, Jun, 2001, 111-116, ISSN : 0167-6911

Primbs, J. A., Nevistic, V. & Doyle, J. C., Nonlinear optimal control: a control Lyapunov function and receding horizon perspective, *Asian Journal of Control*, Vol. 1, No. 1, Jan, 1999, 14-24, ISSN: 1561-8625

Primbs, J. & Nevistic, V., Feasibility and stability of constrained finite receding horizon control, *Automatica*, Vol. 36, No. 7, Jul, 2000, 965-971, ISSN : 0005-1098

Qin, S., & Badgwell, T., A survey of industrial model predictive control technology. *Control Engineering Practice*, Vol. 11, No. 7, Jul, 2003, 733-764, ISSN :0967-0661

Rawlings, J., Tutorial overview of model predictive control, *IEEE Control System Magazine*, Vol. 20, No. 3, Jun, 2000, 38-52, ISSN : 0272-1708

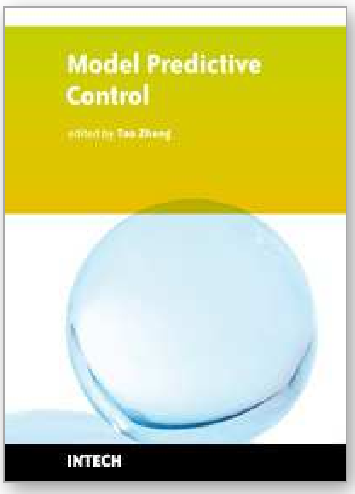
Scokaert, P., Mayne, D. & Rawlings, J., Suboptimal model predictive control (feasibility implies stability), *IEEE Transactions on Automatica Control*, Vol. 44, No. 3, Mar, 1999, 648-654, ISSN : 0018-9286

Song, Q., Jiang, Z. & Han, J., Noise covariance identification-based adaptive UKF with application to mobile robot system, *Proceedings of IEEE International Conference on Robotics and Automation (ICRA 2007)*, pp. 4164-4169, ISSN: 1050-4729, Italy, May, 2007, Roma

Sontag, E., A 'universal' construction of Artstein's theorem on nonlinear stabilization, *Systems & Control Letters*, Vol. 13, No. 2, Aug, 1989, 117-123, ISSN : 0167-6911

Zou, T., Li, S. & Ding, B., A dual-mode nonlinear model predictive control with the enlarged terminal constraint sets, *Acta Automatica Sinica*, Vol. 32, No. 1, Jan, 2006, 21-27, ISSN : 0254-4156

Wesselowske, K. & Fierro, R., A dual-mode model predictive controller for robot formations, Proceedings of the 42nd IEEE Conference on Decision and Control, pp. 3615-3620, ISSN : 0191-2216, HI, USA, Dec, 2003, IEEE, Maui



Model Predictive Control
Edited by Tao Zheng

ISBN 978-953-307-102-2
Hard cover, 304 pages
Publisher Sciyo
Published online 18, August, 2010
Published in print edition August, 2010

Frontiers of Model Predictive Control Robust Model Predictive Control Nonlinear Model Predictive Control Excellent Applications Guide for Researchers and Engineers Recent Achievements of Authors over the World Theory with Practical Examples Kinds of Algorithms for Choice

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Yuqing He and Jianda Han (2010). A New Kind of Nonlinear Model Predictive Control Algorithm Enhanced by Control Lyapunov Functions, Model Predictive Control, Tao Zheng (Ed.), ISBN: 978-953-307-102-2, InTech, Available from: http://www.intechopen.com/books/model-predictive-control/a-new-kind-of-nonlinear-model-predictive-control-algorithm-enhanced-by-control-lyapunov-functions

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.