Local optimality for hinging hyperplanes-based model predictive control

Jun Xu ∗ Xiaolin Huang ∗ Shuning Wang ∗

∗ Tsinghua National Laboratory for Information Science and Technology, Department of Automation, Tsinghua University, Beijing, 100084, China
(e-mail: yun-xu05, huangxl06@mails.tsinghua.edu.cn, swang@mail.tsinghua.edu.cn)

Abstract: The model predictive control based on hinging hyperplanes (HH) model is considered in this paper. Compared with previous work (Chikkula and Lee (1998)), in which a global optimum is obtained, here, we are dedicated to finding the local optima to lessen the computational burden. The necessary and sufficient conditions for a point to be locally optimal are given and based on these conditions, a multi-start algorithm searching for sufficiently good local optimum is proposed. Simulation results show the effectiveness of the proposed control strategy.

Keywords: Predictive control; piecewise linear; local optimality

1. INTRODUCTION

The idea of model predictive control (MPC) can be traced back to 1960s (García et al. (1989)), and the first MPC techniques were developed in the late 1970s (Qin and Badgwell (2003)). In MPC, based on the present state, a sequence of future control actions over a control horizon is computed by solving a finite horizon open-loop optimization problem, only the first control action is implemented on the system. The whole procedure keeps repeating at next time instants, hence, it is also called receding horizon control (RHC). MPC mainly consists of a predictive model and a controller that are built using process identification and optimization techniques, respectively. The predictive model is used to predict outputs of the system at future discrete time instants while the controller design problem is an open-loop optimization problem. Compared with traditional control strategy, in which the control signal is pre-computed off-line, MPC can handle processes with many manipulated and controlled variables and constraints on them systematically, thus is more appropriate for complex system. The flow chart of MPC is depicted in Fig.1.

In early MPC, linear predictive model is often used (Muske and Rawlings (1993)). However, linear assumption is inadequate for some highly or moderately nonlinear processes operating on large regimes (Henson (1998)), such as polymerization reactors, biochemical reactors, pH processes and high purity distillation. Nonlinear predictive model then begins to attract the attention of researchers, including neural networks (Yu et al. (1999), Chen and Yea (2002), Aggelogiannaki and Sarimveis (2007)), Volterra kernel (Maner and Doyle (1997)), piecewise linear model (Chikkula and Lee (1998), Xu et al. (2009b)) and so on. In Morari and Lee (1999), it is pointed out that the piecewise linear structure has potentials in industrial applications and this paper focuses on continuous piecewise linear (CPWL) model, which is linear in subregions and also continuous on the boundaries. The additional continuity may facilitate the controller design procedure by introducing continuity into the optimization problem, which will be confirmed later in this paper.

In Chikkula and Lee (1998), the hinging hyperplanes (HH) predictive model is used, which is one kind of CPWL model and proposed in Breiman (1993). In Xu et al. (2009b), a novel CPWL model, called adaptive hinging hyperplanes (AHH) is used as the predictive model. Both the two CPWL models have advantages and disadvantages, for example, HH model is simple but not robust to the initial value in the identification procedure, AHH model is more efficient to identify but has more complex structure (Xu et al. (2009a)). No matter which CPWL model we choose, the resulting optimization problem is actually a continuous optimization problem. For this kind of problem, the local optimum plays a great role in the solving of the problem.

* Jointly supported by the National Natural Science Foundation of China (61074118, 60974008) and the Research Fund of Doctoral Program of Higher Education (200800030029)
However, in Chikkula and Lee (1998), the optimization problem is formulated as a mixed integer quadratic programming (MIQP) and the local optimality cannot be guaranteed unless a global optimum is got. In Xu et al. (2009b), the local optimality condition is also not clearly stated.

As the HH model owns relatively simple structure, in this paper, we consider HH-based MPC and give the conditions for a point to be locally optimal. The multi-start algorithm searching for a good local optimum is proposed. The (at least) locally optimal control inputs are then implemented on a nonlinear system and the result is satisfactory.

The paper is organized as follows. In Section 2, similar to the description in Chikkula and Lee (1998), the HH predictive model is built and the controller design problem based on this is formulated. The continuity of the optimization problem is verified. Section 3 states the local optimal condition of a continuous piecewise quadratic programming, in which the subregion is a polyhedron. Then, back to the HH predictive model, Section 4 gives a more clear condition for local optimum and propose a multi-start searching algorithm to obtain a local optimum good enough. One example is given in Section 5 to illustrate the efficiency of the proposed strategy. Last, Section 6 ends the paper with some conclusions.

2. MODEL PREDICTIVE CONTROL BASED ON HINGING HYPERPLANES

In this section, MPC is implemented by applying the HH predictive model. The HH method is proposed by Breiman (Breiman (1993)) and actually a tool of nonlinear function approximation. By approximating the nonlinear system with the HH model, the controller is designed for a continuous piecewise linear system.

2.1 HH predictive model

In nowadays research literature, MPC is formulated always in the state space. Assume the nonlinear process is described by the nonlinear discrete time model,

\[ x(t + 1) = f(x(t), u(t)), \]

where \( f \) is a nonlinear function, \( x(t) \in \mathbb{R}^{d_x}, u(t) \in \mathbb{R}^{d_u} \) are the state and input variables at time \( t \), respectively.

To be simple, thereafter in this paper, the state variable is restricted to be in 1-dimension and rewrite as \( x \). The HH approximated model can be written as follows,

\[ \dot{x}(t + 1) = \hat{f}(x(t), u(t)) \]

\[ = a_0 x(t) + \beta_0 u(t) + \gamma_0 + \sum_{m=1}^{M} \eta_m h_m(x(t), u(t)). \]

in which \( a_0, \gamma_0 \in \mathbb{R}, \beta_0 \in \mathbb{R}^{d_u} \) and \( h_m(x(t), u(t)), 1 \leq m \leq M \) is the basis function. The HH model can be seen as a linear combination of basis functions and a linear function, \( \eta_m \) is the linear coefficients, \( \eta_m = \pm 1 \). The basis function is also called the hinge function. The name “hinge function” is followed from the fact that in 3-dimension, the basis function is formed by two hyperplanes joined together at a line or a joint, which in Breiman (1993) is named as a hinge. The hinge function can be expressed as

\[ h_m(x(t), u(t)) = \max\{0, a_m x(t) + \beta_m u(t) + \gamma_m\} \]

where \( a_m, \gamma_m \in \mathbb{R}, \beta_m \in \mathbb{R}^{d_u} \). Fig. 2 describes this function in 3-dimension, i.e., the independent variables \( x(t) \) and \( u(t) \) are both in 1-dimensional.

The parameters \( a_m, \beta_m, \gamma_m(0 \leq m \leq M) \) and \( \eta_m(1 \leq m \leq M) \) in the HH approximated model can be identified by the hinge finding algorithm (HFA) (Breiman (1993), which is actually a Newton algorithm. In Pucar and Sjöberg (1998), a damped Newton algorithm is proposed and the convergence of the algorithm is improved.

Assume the HH predictive model divide the space of \( [x(t), u(t)^T] \) into \( N \) nonoverlapping subregions \( \Omega_1, \Omega_2, \ldots, \Omega_N \), in each subregion, the HH predictive model describes a linear relationship,

\[ \hat{x}(t + 1) = a_t x(t) + b_t u(t) + d_t, ~ \forall [x(t), u(t)^T]^T \in \Omega_i \]

where \( a_t, d_t \in \mathbb{R} \) and \( b_t \in \mathbb{R}^{d_u} \).

2.2 Controller design problem based on HH model

Based on the HH predictive model built off-line, the online controller design problem can be expressed as,

\[ \min_u J = \sum_{t=0}^{ph} \Phi (\hat{x}(t)) + \sum_{t=0}^{ch-1} \Psi (u(t)) \]

s.t. \( x_{\min} \leq \hat{x}(t) \leq x_{\max}, t = 1, \ldots, ph \)

\[ u_{\min} \leq u(t) \leq u_{\max}, t = 0, \ldots, ch - 1 \]

\[ u(t) = u(ch - 1), ~ t = ch, \ldots, ph - 1 \]

Suppose \( U = [u(0)^T, \ldots, u(ph - 1)^T]^T \) is the control variable, \( x(0) \) is the initial state and \( u(-1) \) is the last control input. \( \hat{x}(t) \) is the output of the HH predictive model at time \( t \), \( ph \) is the prediction horizon and \( ch \) is the control horizon, they satisfy \( ch \leq ph \). The notation \( u_{\min} \leq u(t) \) implies that each component of the vector \( u_{\min} \) is less than that of the vector \( u(t) \). In the cost function (5a), the functions \( \Phi \) and \( \Psi \) take the following form,

\[ \Phi(\hat{x}(t)) = P(\hat{x}(t) - x_s)^2 \]

\[ \Psi(u(t)) = (u(t) - u(t - 1))^T Q(u(t) - u(t - 1)) \]

where \( P \) is a positive integer while \( Q = Q^T \) is a positive definite matrix. As is stated in the problem (5), things we should do at time \( t \) is to decide the control \( U \) to drive the system to the set point \( x_s \), utilizing the given information of \( x(0) \) and \( u(-1) \). After obtaining the solution \( U \) of the controller design problem (5), actually only the first input vector \( u(0) \) is implemented on the system. At each time instant, the optimization problem (5) is reformulated and solved in real time, taking into account the new available data.
Thanks to the continuous piecewise linear property of the HH model, without considering the constraints on the state variable, the cost function and predicted states are continuous.

**Proposition 1.** Assume $x_{\text{min}} = -\infty$ and $x_{\text{max}} = \infty$, if $\hat{x}(t+1)$ is a continuous piecewise linear function of $(\hat{x}(t), U)$, then the cost function $J(x(0), U)$ is continuous piecewise quadratic and the predicted states $\hat{x}(t)$, $1 \leq t < ph$ is continuous piecewise linear with respect to $(x(0), U)$.

**Proof.** For the CPWL system (in this paper is the HH predicted system), let $\varphi(t) = [x(t), u(t)]^T$, given a sequence $\varpi = [\varpi_1, \ldots, \varpi_{ph}]^T$ over the prediction horizon, assume the system switches according to this sequence, i.e.,

$$\varphi(0) \in \Omega_{\varpi_1}, \ldots, \varphi(ph-1) \in \Omega_{\varpi_{ph}}$$

then at time instant $t$, the predicted state is

$$\hat{x}(t) = a_{\varpi} \cdot x(t-1) + b_{\varpi} \cdot u(t-1) + d_{\varpi}.$$  

Furthermore, it can be written as a linear function of the initial state $x(0)$ and the optimized control variable $U$,

$$\hat{x}(t) = a_{\varpi} \cdot a_{\varpi-1}, \ldots, a_{\varpi-1} \cdot x(0) + a_{\varpi-2} \cdot a_{\varpi-3}, \ldots, a_{\varpi-3} \cdot a_{\varpi-4}, \ldots, a_{\varpi-ph} \cdot a_{\varpi-ph-1}, \ldots, a_{\varpi-ph} \cdot x(0) \cdot U$$

As there are no constraints on the state variable $x$, the domain of $\hat{x}(t)$ is connected. Let $X_{\varpi} = \{[x(0), U]^T | \varphi(t-1) = [\hat{x}(t-1), u(t-1)]^T \in \Omega_{\varpi_1}, t = 1, \ldots, ph\}$, then according to the expression of $\hat{x}(t)$, $X_{\varpi}$ is a polyhedron. In $X_{\varpi}$, $J(x(0), U)$ is a quadratic function and the state $\hat{x}(t)$, $1 \leq t \leq ph$, admits a linear expression.

For two different switching sequences $\varpi$ and $\pi$, if $X_{\varpi} \cap X_{\varpi} \neq \emptyset$, then at each time instant $t$ ($0 \leq t \leq ph-1$), there exists,

$$\varphi(t-1) \in \Omega_{\varpi_1} \cap \Omega_{\pi_1}$$

As the CPWL predictive model indicates, the state $\hat{x}(t)$ is continuous on the common boundary of the two subregions, hence, both $J$ and $\hat{x}(t)$ ($1 \leq t \leq ph$) are continuous on $X_{\varpi} \cap X_{\varpi}$.

From Proposition 1 and the property of continuous function, $J$ and $\hat{x}(t)$ are also continuous on the space of $U$. The domain of $U$ is divided into nonoverlapping polyhedra corresponding to different switching sequences $\varpi$, denoted by $\Gamma_{\varpi}$. In $\Gamma_{\varpi}$, the states over the prediction horizon are linear and the cost function is quadratic. Moreover, for two different switching sequences $\varpi$ and $\pi$, at the intersection (if it is not empty) of the corresponding subregions $\Gamma_{\varpi} \cap \Gamma_{\varpi}$, both the states and the cost function are continuous.

In order to solve the problem (5), the problem has to be feasible and the definition of feasibility is as follows.

**Definition 2.** The optimization problem (5) is feasible, if there exists control sequence $U = [u(0)^T, \ldots, u(ph-1)^T]^T$ such that the constraints (5b)-(5d) are satisfied. The initial state $x(0)$ is called the feasible initial state and the set of feasible control sequence related to $x(0)$ is denoted as $F(x(0))$.

In this paper, we intend to divide the optimization problem into subproblems related to different switching sequences $\varpi$, and the subproblem is a quadratic programming, hence, the feasibility is considered in each subproblem. The next section describes how to solve the continuous piecewise quadratic programming with polyhedral subregions.

3. LOCAL OPTIMALITY OF CONTINUOUS PIECEWISE QUADRATIC PROGRAMMING

Apparently, traversing all possible switching sequences will give a global optimum of the optimization problem (5). However, the number of possible switching sequences will grow exponentially with the prediction horizon $ph$. Due to the continuous property of this nonlinear problem, the searching of a local minimum is also of great value.

Suppose there are $K$ possible switching sequences and the space of $U$ is divided into $K$ nonoverlapping subregions $\Gamma_1, \ldots, \Gamma_K$, which can be expressed by linear inequalities.

$$\forall U \in \Gamma_k, 1 \leq k \leq K,$$

there exists,

$$\begin{bmatrix} J \\ \hat{x}(1) \\ \vdots \\ \hat{x}(ph) \end{bmatrix} = \begin{bmatrix} \frac{1}{2}U^TH_kU + x(0)D_k^TU + \frac{1}{2}G_kx(0)^2 \\ A_{1k}x(0) + B_{1k}^TU + C_{1k} \\ \vdots \\ A_{ph,k}x(0) + B_{ph,k}^TU + C_{ph,k} \end{bmatrix}$$

where $H_k \in \mathbb{R}^{(ph \times ph) \times (ph \times ph)}$, $A_{1k}, D_k \in \mathbb{R}^{(ph \times ph)}$, $A_{ph,k}, C_k \in \mathbb{R}^{(1 \times t \leq ph)}$. Bearing in mind the fact that in (5a) $P \in \mathbb{R}$ is positive and $Q = Q^T$ is definite, it follows that $H_k = H_k^T$ is positive definite and $G_k$ is positive. Under this scheme, the piecewise constraint (5b) becomes linear, i.e.,

$$x_{\text{min}} \leq A_{hk}x(0) + B_{hk}^TU + C_{hk} \leq x_{\text{max}}, t = 1, \ldots, ph$$

Hence, together with the expression of $\Gamma_k$, linear constraints (5c) and equality constraints (5d), the subproblem in the subregion $\Gamma_k$ can be expressed as follows,

$$\min_{J_k} = \frac{1}{2}U^TH_kU + x(0)D_k^TU + \frac{1}{2}G_kx(0)^2$$

s.t. $E_kU \preceq T_kx(0) + V_k$

where the matrix $E_k, T_k, V_k$ is determined by $\Gamma_k$ as well as the constraints (5b)-(5c). Denote the matrix in (11c) as $S_U$, with size $d_u \cdot (ph-ch) \times d_u \cdot ph$. $d_u$ is $d_u \times d_u$ square matrix with elements 0 and $I_d$ is a $d_u \times d_u$ identity matrix.

As is mentioned above, usually $K$ is large and obtaining the global optimum is very time-consuming. Therefore, given the initial state $x(0)$, we are dedicated to searching for local optimum and next we'll discuss conditions for a point to be locally optimal.

Given a point $U^*$, suppose the subregions containing it are $\Gamma_{k_1}, \ldots, \Gamma_{k_N}$, we claim that $U^*$ is locally optimal for the problem (5) if and only if $U^*$ is the optimum of the subproblems in all the concerned subregions.

**Lemma 3.** Given a point $U^*$, denote the set of subregions containing $U^*$ as $\{\Gamma_{k_1}, \ldots, \Gamma_{k_N}\}$, where $N_k$ is the number of the set, then $U^*$ is a local optimum for the continuous piecewise quadratic programming problem (5) if and only if $\forall U \in \Gamma_k, k \in \{k_1, \ldots, k_N\}$, there exists $J(U^*) \leq J(U)$.

**Proof.** First the necessity. From the fact that $U^*$ is locally optimal, we have, $\forall \epsilon > 0$, there exists a positive scalar $\delta$, 

$$5521$$
such that $\forall \|U - U^*\| \leq \delta$, the inequality $J(U^*) \leq J(U)$ holds, where $\|\cdot\|$ is some kind of norm.

Suppose there is a $\tilde{k} \in \{k_1, \ldots, k_{N_K}\}$ and $\tilde{U} \in \Gamma_{\tilde{k}}$ such that $J(\tilde{U}) < J(U^*)$. As $\Gamma_{\tilde{k}}$ is a polyhedron, $\forall 0 < \zeta \leq 1$, the point $U_\zeta = U^* + \zeta (\tilde{U} - U^*)$ satisfies $U_\zeta \in \Gamma_{\tilde{k}}$. Hence, we can find a $\zeta$ such that $\|U_\zeta - U^*\| \leq \delta$ and $U_\zeta \in \Gamma_{\tilde{k}}$. According to the convexity of the quadratic cost function $J_{\tilde{k}}$, there exists,

$$J(U_\zeta) = J_{\tilde{k}}(U_\zeta) \leq \zeta J_{\tilde{k}}(U^*) + (1 - \zeta) J_{\tilde{k}}(\tilde{U}) < J_{\tilde{k}}(U^*) = J(U^*),$$

contradicting with the assumption that $U^*$ is the local optimum.

Then the sufficiency. As $U^*$ is the optimum in each subregion containing it, it is obvious that $U^*$ is locally optimal. □

Based on the above discussions, assume the number of inequality constraints for each subproblem (11), i.e., the number of rows in $E_k$, is $c_k$, the following lemma gives the necessary and sufficient conditions for a point to be locally optimal.

**Lemma 4.** Given a point $U^*$, denote the set of subregions containing $U^*$ as $\{\Gamma_{k_1}, \ldots, \Gamma_{k_{N_K}}\}$, where $N_K$ is the number of the set, then $U^*$ is locally optimal if and only if for each $\Gamma_{k}, k \in \{k_1, \ldots, k_{N_K}\}$, there exists nonnegative scalars $\lambda_{k_1}, \ldots, \lambda_{k_{N_K}}$ and scalars $v_{k_1, 1}, \ldots, v_{k, t_{t_{ph-ch}}}$ such that the following holds,

$$E_k U^* - T_k x(0) - V_k \leq 0,$$  \hspace{1cm} (12)

$$H_k U^* + D_k x(0) + \sum_{n=1}^{c_n} \lambda_{k_n} (E_k^{(n)} U^*) + d_u(u^{(ph-ch)}) + \sum_{n=1}^{c_n} v_{k_n} (S_k^{(n)} U^*) = 0,$$  \hspace{1cm} (13)

$$\lambda_{k,n} \cdot (E_k^{(n)} U^* - T_k^{(n)} x(0) - V_k^{(n)}) = 0, \forall 1 \leq n \leq c_k,$$  \hspace{1cm} (14)

where $E_k^{(n)}$ is the n-th row of $E_k$ and the same hold for the matrices $T_k, V_k$ and $S_k$.

**Proof.** First, we emphasize that the value of $c_k$ may differ for different $k$, which is due to the different number of constraints expressing different subregions.

From Lemma 3, the point $U^*$ is locally optimal if and only if it is the optimum in all the subregions $\Gamma_{k_1}, \ldots, \Gamma_{k_{N_K}}$ containing it. In each subregion $\Gamma_k$, $k \in \{k_1, \ldots, k_{N_K}\}$, the subproblem is a quadratic programming and the Karush-Kuhn-Tucker (KKT) conditions are necessary and sufficient for $U^*$ to be optimal, which can be expressed as

$$E_k U^* - T_k x(0) - V_k \leq 0,$$

$$\lambda_{k,n} \geq 0, \forall 1 \leq n \leq c_k,$$

$$\nabla J_k(U^*) + \sum_{n=1}^{c_n} \lambda_{k,n} \nabla (E_k^{(n)} U - T_k^{(n)} x(0) - V_k^{(n)})U^* + d_u(u^{(ph-ch)}) + \sum_{n=1}^{c_n} v_{k,n} \nabla (S_k^{(n)} U)U^* = 0,$$

$$\lambda_{k,n} \cdot (E_k^{(n)} U^* - T_k^{(n)} x(0) - V_k^{(n)}) = 0, \forall 1 \leq n \leq c_k,$$

According to the expression in the $k$-th subregion, $\nabla J_k(U^*)$ can be written as $H_k U^* + D_k x(0)$, hence the KKT conditions is the same as there exists nonnegative scalars $\lambda_{k,n}$ and scalars $v_{k,n}$ such that

$$E_k U^* - T_k x(0) - V_k \leq 0,$$

$$H_k U^* + D_k x(0) + \sum_{n=1}^{c_k} \lambda_{k,n} (E_k^{(n)} U^*) + \sum_{n=1}^{c_n} v_{k,n} (S_k^{(n)} U^*) = 0,$$

$$\lambda_{k,n} \cdot (E_k^{(n)} U^* - T_k^{(n)} x(0) - V_k^{(n)}) = 0, \forall 1 \leq n \leq c_k.$$  \hspace{1cm} (15)

After obtaining the necessary and sufficient conditions for a point $U^*$ to be locally optimal, a natural question is how to determine the subregions $\Gamma_{k_1}, \ldots, \Gamma_{k_{N_K}}$ containing it as well as the linear constraints $E_k U^* - T_k x(0) - V_k \leq 0, k \in \{k_1, \ldots, k_{N_K}\}$. In the next section, we attempt to give an answer.

4. ON-LINE CONTROLLER DESIGN BASED ON HH PREDICTIVE MODEL

When refined to each subregion $k$, actually we can only consider the active constraints, i.e., such constraints satisfying $E_k^{(n)} U^* - T_k^{(n)} x(0) - V_k^{(n)} = 0$. The reason for this is the complementary slackness (14), if there exists $E_k^{(n)} U^* - T_k^{(n)} x(0) - V_k^{(n)} < 0$, then the corresponding lagrange multiplier $\lambda_{k,n}$ is zero and can be omitted from the conditions (13)-(14) (Boyd and Vandenberghe (2004)).

According to the discussion in Section 2, the HH model describing the nonlinear system is

$$\dot{x}(t + 1) = \alpha_0 \dot{x}(t) + \beta_0^T u(t) + \gamma_0$$

$$+ \sum_{m=1}^{M} \gamma_m \max\{0, \alpha_m \dot{x}(t) + \beta_m^T u(t) + \gamma_m\}$$

Given a point $U^* = [u(0)^T, \ldots, u^{(ph-1)})^T]^T$, we can decide the following pairs $(x(0), u(0)^T)^T, (\dot{x}(1), u(1)^T)^T, \ldots, (\dot{x}(ph-1), u^{(ph-1)})^T$ using the above expression. For each pair $(x(t), u(t)^T)^T, 0 \leq t \leq ph - 1$, let $M_t = \{m | \alpha_m \dot{x}(t) + \beta_m^T u(t) + \gamma_m = 0\}$, then the cardinality of $M_t$, denoted as $|M_t|$, describes the number of elements in $M_t$. If $M_t$ is not empty, $U^*$ is in the intersection of some subregions.

In fact, at the point $U^*$, the active constraints can be divided into feasibility constraints (if $U^*$ reaches the boundary of the feasible region), and $\alpha_m \dot{x}(t) + \beta_m^T u(t) + \gamma_m = 0, 0 \leq t \leq ph - 1, \forall m \in M_t$. Assume

$$z_m = \begin{cases} -1 & \text{if } \alpha_m \dot{x}(t) + \beta_m^T u(t) + \gamma_m \geq 0 \\ 1 & \text{if } \alpha_m \dot{x}(t) + \beta_m^T u(t) + \gamma_m < 0 \end{cases}$$

then for each combination of $z_k, m_k$, i.e., a string of $-1$ and 1, with length $\sum_{t=0}^{ph-1} |M_t|$, the expression of $J, \dot{x}, \ldots, \dot{x}(ph - 1)$ in (10) can be determined. Actually, different combinations relate to different subregions containing $U^*$ and it is clear that these subregions are nonoverlapping. Obviously, the number of all possible combinations is $\prod_{t=0}^{ph-1} 2^{|M_t|}$. 

5522
Suppose the active feasibility constraint related to the k-th combination $z_{k,m_t}$ is $\bar{E}_k U^* - \bar{T}_k x(0) - \bar{V}_k \leq 0$, with $\bar{c}_k$ linear equalities. From the above discussions, if $U^* \in F(x(0))$, necessary and sufficient conditions for it to be locally optimal can be described by the following theorem.

**Theorem 5.** Given a point $U^* = \{u^*(0)^T, \ldots, u^*(ph - 1)^T\}^T \in F(x(0))$, then $U^*$ is locally optimal for the optimization problem (5) if and only if for the k-th combination $z_{k,m_t}$, $\forall 1 \leq k \leq \sum_{t=0}^{ph-1} |M_t|$, with the active feasibility constraint $\bar{E}_k U^* - \bar{T}_k x(0) - \bar{V}_k \leq 0$, there exist nonnegative scalars $\lambda_{k,m_t}$ $(m_t \in M_t)$, $\alpha$, $\beta$, $\gamma$, $\bar{c}_k$, $\bar{v}_k$ plus scalars $\lambda_{k,n}$ such that

$$H_k U^* + D_k x(0) + \sum_{t=0}^{ph-1} \sum_{m_t \in M_t} \lambda_{k,m_t} z_{k,m_t} E_{m_t}$$

$$+ \sum_{n=1}^{\bar{c}_k} \lambda_{k,n} (\bar{E}_{(n)}^k)^T + \sum_{n=1}^{\bar{c}_k} \lambda_{k,n} (S_{U}^{(n)})^T = 0$$

(16)

where $E_{m_t}$ can be expressed as

$$E_{m_t} = \alpha_{m_t} B_{m_t}^T + \begin{bmatrix} 0_{d_u \times d_u} & I_{d_u} & 0 \end{bmatrix} \cdot \beta_{m_t}$$

(17)

**Proof.** As Lemma 4 states, $U^*$ is the local optimum if and only if for any subregion containing it, conditions (12)-(14) hold. The k-th combination $z_{k,m_t}$ relates to a subregion, and it is obvious that traversing all combinations of $z_{k,m_t}$ cover all the subregions containing $U^*$. Hence, $U^*$ is locally optimal for the optimization problem (5) if and only if it is optimal for the k-th combination $(\forall 1 \leq k \leq \sum_{t=0}^{ph-1} |M_t|)$. If $U^*$ is feasible, according to the complementary slackness, only the following active constraints have to be considered, i.e.,

$$z_{k,m_t} \cdot (\alpha_{m_t} \dot{x}(t) + \beta_{m_t}^T u(t) + \gamma_{m_t}) \leq 0$$

$\forall 0 \leq t \leq ph - 1, \forall m_t \in M_t$

$$\bar{E}_k U - \bar{T}_k x(0) - \bar{V}_k \leq 0$$

(18)

(19)

The number of linear inequalities is $\sum_{t=0}^{ph-1} |M_t| + \bar{c}(k)$, corresponding to nonnegative scalars $\lambda_{k,m_t}$, $\lambda_{k,n}$. Replacing the linear inequality matrix $E_{k}$ in (13) with the above inequalities, we obtain (16).

In the above conditions (16), if $M_t(0 \leq t \leq ph - 1)$ is empty, $U^*$ is an interior point of some subregion and the optimality condition (16) reduces to

$$H_k U^* + D_k x(0) + \sum_{n=1}^{\bar{c}_k} \lambda_{k,n} (S_{U}^{(n)})^T = 0.$$  \hspace{1cm} (20)

In the case when there are many active constraints, i.e., the number of intersecting subregions at $U^*$ is large, those active constraints may be dependent, or in other words, some active constraints are redundant, in which case there may exist multiple solutions of the scalars $\lambda_{k,m_t}$, $\lambda_{k,n}$ and $v_{k,n}$. At this time, $U^*$ is locally optimal if and only if as least one group of $\lambda_{k,m_t}$, $\lambda_{k,n}$ are nonnegative.

If a point is not locally optimal and $M_t \neq \emptyset$, then there must exist a combination $z_{k,m_t}$ such that we can get better solution through the quadratic programming

$$\min J_{k_o} \quad \text{s.t.} \quad z_{k,m_t} \cdot (\alpha_{m_t} x(t) + \beta_{m_t}^T u(t) + \gamma_{m_t}) \leq 0,$$

$\forall t = 0, \ldots, ph - 1, \forall m_t \in M_t$

(21a)

$$\alpha_{m_t} x(t) + \beta_{m_t}^T u(t) + \gamma_{m_t} \cdot (\alpha_{m_t} x(t) + \beta_{m_t}^T u(t) + \gamma_{m_t}) \geq 0$$

$\forall t = 0, \ldots, ph - 1, \forall m_t \in M_t$

(21b)

$\bar{E}_{k_o} U - \bar{T}_{k_o} x(0) - \bar{V}_{k_o} \leq 0,$

(21c)

where $\alpha_{m_t} x(t) + \beta_{m_t}^T u(t) + \gamma_{m_t}$ is fixed when given $U^*$. In fact, if $M_t = \emptyset$, we can also construct this quadratic programming by just omitting the constraints (21b).

Based on the discussion of the local optimality of a point, we propose the multi-start algorithm for on-line controller design, in which $M_t$ starting points are chosen and result in the same number of local optima, and the one with the minimal cost is selected as the final solution. During the travel from one starting point, the cost keeps decreasing. Considering the restriction of the sampling interval $T_s$, all of the above searching must be done within $T_s$. The multi-start algorithm can be summarized as follows.

**Algorithm 1.**

Step 1: Set a value for $M_s$, $j = 1$, $t_s = T_s/M_s$, then $t_s$ is the time permitted for the searching starting from one starting point. The initial cost $J_{opt}$ is set to be a positive scalar large enough.

Step 2: Generate a feasible initial control sequence $U^0$ using (16) to judge whether it is locally optimal.

Step 3: If $U^0$ is locally optimal or the searching time $t_j \geq 0.9 t_s$, $J^j = J(U^0)$, $U^j = U^0$, go to Step 5. If it is not, construct the quadratic programming (21).

Step 4: Solve the quadratic programming (21), get a new control sequence $U$, let $U^0 = U$, go to Step 3.

Step 5: If $J^j < J_{opt}$, $J_{opt} = J^j$, $U_{opt} = U^j$. If $j < M_s$, $j = j + 1$, go to Step 2, else, exit.

The resulting control sequence $U_{opt}$ is the local optimum with the least cost and $U^{(opt)}(0)$ is implemented on the dynamic system. The same things are done at the next sample instants.

5. SIMULATION STUDY

In this section, we’ll give one example to illustrate the control strategy proposed. The example comes from Chikkula and Lee (1998), in which the HH predictive model has been built through model identification procedure. What we want to emphasize is the performance of the local optima, while compared with the global optimum.

**Example 6.** Isothermal CSTR

Consider the isothermal CSTR process given by the following differential equation,

$$\frac{dy}{dt} = u - y - \frac{1}{1 + \frac{y}{k_1} + \frac{y}{k_2}},$$

(22)

in which $y$ is the reactant concentration and $u$ is the feed reactant concentration. $k_1$ and $k_2$ are the kinetic parameters and $\beta$ is a constant. The single enzyme-catalyzed reactions with substrate inhibited kinetics and ethylene hydrogenation in an isothermal CSTR can be
described by the above model. This example was originally proposed by Bruns and Bailey (1975) and later studied by Chikkula and Lee (1998).

According to Chikkula and Lee (1998), the regression vector is chosen to be \( [y(t - 1), u(t - 1)]^T \) and the output \( y(t) \) at time \( t \) can be approximated by the HH predictive model as follows,

\[
y(t) = -0.6557y(t - 1) - 0.1257u(t - 1) + 0.2347 + \max\{0, 0.5949y(t - 1) + 0.6881u(t - 1) - 0.4405\}\]
\[
+ \max\{0, 0.4850y(t - 1) + 0.1047u(t - 1) - 0.1099\}\]
\[
- \max\{0, -0.7895y(t - 1) - 0.0086u(t - 1) + 0.2163\}\]
\[
- \max\{0, -0.1998y(t - 1) - 0.0024u(t - 1) + 0.0052\},
\]

(23)

Considering a step change in the set point from 0.3 to 0.001, the tuning parameters are the same as in Chikkula and Lee (1998) and the sampling interval is set to be 0.4 min.

If using the traversing strategy, at each time instant, one has to traverse \( 2^M \cdot ph \) switching sequences to find the global optimum, which takes about 4 min with MATLAB R2007a in AMD Athlon(tm) 64 X2 Dual Core Processor 5000+ 2.60 G. However, using the HH predictive model (23), the controller design problem is a continuous piecewise quadratic programming problem and can be solved using Algorithm 1 efficiently.

Based on the HH predictive model, for 20 time instants, the system is implemented with the control sequence obtained by Algorithm 1, and global traversing (which cannot be finished within the sampling interval and is just used to give the global optimum for comparison). The control performances in these two situations are compared in Fig. 3.

![Fig. 3. Outputs of HH-based MPC (using Algorithm 1 and global traversing)](image)

From the figure, for this simple example, we know that the local searching strategy Algorithm 1 performs as well as the global searching, while consuming much less time. The system can be driven to the set-point efficiently.

6. CONCLUSION

In this paper, the model predictive control is implemented using the hinging hyperplanes (HH) predictive model, which is a continuous piecewise linear model. Utilizing the continuity of the HH model at the stage of controller design, the local optimality of the controller is discussed. Necessary and sufficient condition for a point to be locally optimal is given under the scheme of HH predictive model and a multi-start algorithm is proposed to get sufficiently good solution. Simulation results show the good performance of the proposed MPC strategy, while compared with the global optimal controller.

However, the feasibility and stability issues raised in HH-based MPC should also be considered in our future work. We’ll also attempt further application in industry.

REFERENCES


