Nonlinear Model Predictive Control of Ball-Plate System based on Gaussian Particle Swarm Optimization

Jianchao Fan
School of Electronic and Information Engineering
Dalian University of Technology, Dalian, P. R. China
fjchao@mail.dlut.edu.cn

Min Han
School of Electronic and Information Engineering
Dalian University of Technology, Dalian, P. R. China
minhan@dlut.edu.cn

Abstract—This paper presents a new nonlinear model predictive control (NMPC) strategy based on the Gaussian particle swarm optimization (GPSO). Through the Taylor expansion, NMPC transforms to a quadratic programming problem with unknown parameters. Hence, for the global convergence character and higher optimization accuracy, GPSO is employed to dynamically perform nonlinear constraint optimization. Finally, the proposed control strategy is applied to Ball-Plate system to verify the effectiveness.

Keywords—nonlinear model predictive control; Gaussian particle swarm optimization; Ball-plate system

I. INTRODUCTION

Model predictive control (MPC) is one of three common algorithms in model-predictive control, which has been developed well in the actual industrial processes. That is because MPC could explicitly handle MIMO systems with input, control single and output constraints. The general design objective of model predictive control is to compute a trajectory of a future manipulated variable $u$ to optimize the future behavior of the plant output $y$. The optimization is performed within a limited time window by giving plant information at the start of the time window [1, 2]. Recent review and survey on the MPC development course could be found in [3, 4]. Further research has been carried out on the design of model predictive control for linear system. However, many practical processes are inherently strong nonlinearities. The linear model is not adequate to dedicate the system dynamics, as would result in poor performance [5]. In general, NMPC design would lead to the formulations of non-convex optimization problems. As such, there is no effective approach that can guarantees global optimal controls for such problems. Thus, NMPC is a challenging task and further investigations on NMPC are deemed absolutely necessary and rewarding [6].


The objective function online learning is a key problem in the NMPC algorithms. The effectiveness of optimization approaches directly affects the control results. Many methods had been applied into this field. Zhu et al [15] proposed Bayesian optimization on constraint optimization. Several recurrent neural networks using duality or projection, for solving linear programming, quadratic programming and general convex nonlinear programming were developed [16-18]. They had shown with desirable features and superior performance. However, when subjected to some restrictions, NMPC objective functions are defined as a complex, non-linear, non-convex search space, which is more suitable for evolutionary algorithms [5]. Recently, many heuristic algorithms have been proposed, such as differential evolution (DE) algorithm, ant colony optimization (ACO) and particle swarm optimization (PSO) et al, which obtain more superior global convergence property and optimization precision. In [19], genetic algorithm was adopted for constraint problems. Yen et al [20, 21] utilized PSO based on the culture algorithm for multi-objective optimizations. Suganthan et al [22] employed differential evolution with ensemble of constraint handling techniques for solving 2010 congress on evolutionary computation (CEC) benchmark problems and achieved better results. All of these methods are independent on continuum, and differentiability properties of objective functions.

Hence, in this paper a new NMPC scheme based on GPSO approach (GPSO+NMPC) is proposed. At the beginning, the NMPC problem transforms to a quadratic programming question with unknown higher order infinitesimal. Then, the particle swarm optimization with Gaussian functions and chaos
mapping (GPSO) is adopted to dynamically perform control variables constraint optimization and on-line approximate to the unknown part. GPSO can achieve more accurate global optimal point for various types of functions. Therefore, the more accurate control variables and high order infinitesimal can be obtained. GPSO+NMPC is effective in the Ball-plate system simulation experiment.

The rest of this paper is organized as follows. Section II presents the NMPC is formulated as quadratic optimization problem with high order infinitesimal. Section III describes the GPSO algorithm. The unknown parameters learning and control scheme are shown in Section IV. Section V highlights the potential of the proposed approach through Ball-plate experimental examples. Concluding remarks are presented in Section VI.

II. PROBLEM FORMULATION

Consider that the discrete-time nonlinear system is expressed as the input-output form by

\[ y(k+1) = f(y(k), u(k)) \] (1)

Here \( f(\cdot, \cdot) \) is continuously differentiable function, \( y(k) \in \mathbb{R}^n \) and \( u(k) \in \mathbb{R}^m \) are the output and input of the system, respectively. By utilizing the rolling linearization approach [10], the above nonlinear system expand to Taylor series around each sampling point \( [y_r(k), u_r(k)] \)

\[ y(k+1) = f(y_r(k), u_r(k)) + \frac{\partial f(y, u)}{\partial y}_{y_r(k),u_r(k)} (y(k) - y_r(k)) \]
\[ + \frac{\partial f(y, u)}{\partial u}_{y_r(k),u_r(k)} (u(k) - u_r(k)) + \epsilon(k), u(k)) \] (2)

where \( \epsilon(k), u(k)) \) is unknown high order infinitesimal. The reference trajectory is

\[ y_r(k+1) = f(y_r(k), u_r(k)) \] (3)

Then, subtract (3) from (2),

\[ y(k+1) - y_r(k+1) = \frac{\partial f(y, u)}{\partial y}_{y_r(k),u_r(k)} (y(k) - y_r(k)) \]
\[ + \frac{\partial f(y, u)}{\partial u}_{y_r(k),u_r(k)} (u(k) - u_r(k)) + \epsilon(k), u(k)) \] (4)

Let \( y'(k) = y(k) - y_r(k), u'(k) = u(k) - u_r(k) \), and the formulation is transformed to

\[ y'(k+1) = \frac{\partial f(y, u)}{\partial y}_{y_r(k),u_r(k)} y'(k) + \frac{\partial f(y, u)}{\partial u}_{y_r(k),u_r(k)} u'(k) \]
\[ + \epsilon(k), u(k)) \] (5)

Rearrange the above equation as

\[ y'(k+1) = Ay'(k) + Bu'(k) + \epsilon(k) \] (6)

Here \( A = \frac{\partial f(y, u)}{\partial y}_{y_r(k),u_r(k)} \), \( B = \frac{\partial f(y, u)}{\partial u}_{y_r(k),u_r(k)} \), \( \epsilon(k) = \epsilon(y(k), u(k)) \). For the above system (6), NMPC could be defined as an optimization problem at each sampling interval,

\[ \min \sum_{j=1}^{N} [y'(k+j|k)]^T Q[y'(k+j|k)] \]
\[ + \sum_{j=1}^{N} [\Delta u(k+j|k)]^T R[\Delta u(k+j|k)] \]

s.t.
\[ u_{\min} \leq u(k+j|k) \leq u_{\max}, j = 0, 1, \cdots, N_u - 1; \]
\[ \Delta u_{\min} \leq \Delta u(k+j|k) \leq \Delta u_{\max}, j = 0, 1, \cdots, N_u - 1; \] (7)

\[ y_{\min} \leq y(k+j|k) \leq y_{\max}, j = 1, 2, \cdots, N; \]

Here \( k \) is the current sampling time; \( \Delta u(k+j|k) \) is the increment of control variable

\[ \Delta u(k+j|k) = u(k+j|k) - u(k+j-1|k), \] (8)

\( N \) is called the prediction horizon; \( N_u \) is called the control horizon dictating the number of parameters used to capture the future control trajectory; \( 0 < N_u \leq N \in \mathbb{R}^{\text{non}}, R \in \mathbb{R}^{\text{non}} \).

Define the following vector,

\[ \overline{y}(k) = [y(k+1|k), \cdots, y(k+N|k)]^T \in \mathbb{R}^{N \times n}, \]
\[ \overline{u}(k) = [u(k|k), \cdots, u(k+N_u-1|k)]^T \in \mathbb{R}^{N_u \times m}, \]
\[ \Delta \overline{u}(k) = [\Delta u(k+1|k), \cdots, \Delta u(k+N_u-1|k)]^T \in \mathbb{R}^{N_u \times m}, \]
\[ \overline{y}(k) = [\overline{y}(k+1|k), \cdots, \overline{y}(k+N|k)]^T \in \mathbb{R}^{N \times n}, \]

and \( \overline{y}(k), \overline{y}(k), \Delta \overline{u}(k) \) are defined as the same. Hence the predictive output \( \overline{y}(k) \) can be expressed as follow,

\[ \overline{y}(k) = S(k)y'(k) + M(k)\overline{u}(k) + \overline{\epsilon}(k) \]
\[ = S(k)y'(k) + M(k)\Delta \overline{u}(k) + V(k)u'(k-1) + \bar{\epsilon}(k) \] (9)

where

\[ S(k) = [A(k) \ A(k)^2 \cdots A(k)^N]^T \in \mathbb{R}^{n \times n}, \]
\[ B(k) \]
\[ (A(k)+1)B(k) \]
\[ \vdots \]
\[ V(k) = (A(k)^{N_u-1} \cdots (A(k)+1)B(k) \]
\[ (A(k)^{N_u} \cdots (A(k)+1)B(k) \]
\[ \vdots \]
\[ (A(k)^N \cdots (A(k)+1)B(k) \]
The above optimization problem can be presented as
\[
\begin{align*}
\min & \quad Q(S(k))' - M(k)\Delta\vec{u}(k) - V(k)u(k-1) - \vec{F}(k) \\
\text{s.t.} & \quad \Delta\vec{u}_{\text{min}} \leq \Delta\vec{u}(k) \leq \Delta\vec{u}_{\text{max}} \\
& \quad \vec{v}_{\text{min}} \leq \vec{v}(k) \leq \vec{v}_{\text{max}} \\
& \quad \vec{y}_{\text{min}} \leq \vec{y}(k) \leq \vec{y}_{\text{max}} \\
& \quad \vec{u}_{\text{min}} \leq \vec{u}(k-1) + H\Delta\vec{u}(k) \leq \vec{u}_{\text{max}} \\
& \quad \bar{u}_{\text{min}} \leq \bar{u}(k) \leq \bar{u}_{\text{max}}
\end{align*}
\]

The above optimization problem can be rewritten as
\[
\begin{align*}
\min & \quad \frac{1}{2} v^T W v + c^T v \\
\text{s.t.} & \quad l_{\text{min}} \leq E v \leq l_{\text{max}}
\end{align*}
\]

where the coefficient matrix are
\[
W = 2(M^TQM + R) \in \mathbb{R}^{N_m \times N_m},
\]
\[
c = -2M^TQ(Sy' - Vu'(k-1) - \vec{F}(k)) \in \mathbb{R}^{N_m},
\]
\[
E = [H \quad I \quad M]^T \in [\mathbb{R}^{12N_m \times N_m}]^{N_m},
\]
\[
l_{\text{min}} = \begin{bmatrix}
\bar{u}_{\text{min}} - \bar{u}(k-1) \\
\Delta\vec{u}_{\text{min}} \\
\vec{y}_{\text{min}} - \vec{y}(k) - S(k)y(k) - V(k)u(k-1) - \vec{F}(k)
\end{bmatrix}
\]
\[
l_{\text{max}} = \begin{bmatrix}
\bar{u}_{\text{max}} - \bar{u}(k-1) \\
\Delta\vec{u}_{\text{max}} \\
\vec{y}_{\text{max}} - \vec{y}(k) - S(k)y(k) - V(k)u(k-1) - \vec{F}(k)
\end{bmatrix}
\]

Hence, through obtaining the solution of (11), the optimal control vector \( \Delta\bar{u}(k) \) can be found.

### III. Gaussian Particle Swarm Optimization

GPSO [12] is a heuristic algorithm which maintains a swarm of candidate solutions, referred as ‘particles’. The best solution \( P_{\text{best}} \) of each particle has achieved so far, which is denoted as \( P_i = (p_{i,1}, p_{i,2}, \cdots, p_{i,d}) \). The best position \( P_{\text{gbest}} \) among all the particles in the population is represented as \( P_g = (p_{g,1}, p_{g,2}, \cdots, p_{g,d}) \). GPSO relies on the exchange of information between particles, in other words this is a hypothesis that social sharing of information among individuals offers an evolutionary advantage. GPSO is similar to other heuristic algorithms in that the system is initialized with a population of random solutions. The velocity for each particle in a \( D \)-dimensional problem space is dynamically adjusted according to the flying experiences of its own and fellows. Therefore, it finds the global best solution by simply adjusting the trajectory of each individual towards its own best position and towards the best neighboring particle at each sampling time.

The velocity and position update equations for the \( d \)-th dimension of the \( i \)-th particle in the swarm may be denoted as
\[
\begin{align*}
\vec{v}_{i,d}(k+1) &= w\vec{v}_{i,d}(k) + c_1 r_1 \cdot (\vec{p}_{i,d}(k) - \vec{x}_{i,d}(k)) \\
& \quad + c_2 r_2 \cdot (\vec{P}_{g,d}(k) - \vec{x}_{i,d}(k)) \\
x_{i,d}(k+1) &= \vec{x}_{i,d}(k) + \vec{v}_{i,d}(k+1)
\end{align*}
\]

where \( w \in [0,1] \) is the inertia weight, determining how much of the previous velocity of the particle is preserved. And \( r_1, r_2 \in [0,1] \) denote two uniform random numbers samples, \( c_1, c_2 \) are acceleration constants. In a \( D \)-dimensional search space, the position vector of the \( i \)-th particle is represented as \( \vec{x}_i = [x_{i,1}, x_{i,2}, \cdots, x_{i,d}] \), where \( x_{i,d} \in [x_{\text{min},d}, x_{\text{max},d}] \), \( d = 1, \cdots, D \). \( x_{\text{min},d} \) and \( x_{\text{max},d} \) are the lower and upper bounds for the dimension \( d \), respectively. Let \( V_i = (v_{i,1}, v_{i,2}, \cdots, v_{i,D}) \) be the current velocity for particle \( i \), which is limited to a maximum velocity \( V_{\text{max}} = (V_{\text{max},1}, \cdots, V_{\text{max},D}) \).

In order to overcome the particle swarm optimization algorithm premature convergence, the PSO methods is trained dynamically from the early global search to the latter local approximate. The basic Gaussian function is shown as in Fig. 1

In above figure, there are two positions \( x_1 \) and \( x_2 \) standing for the current location of particles, respectively. There are two major parameters in the Gaussian formula as shows in equation (6), which are the center \( c \) and variance \( \sigma \)
\[
f(x; \sigma, c) = e^{-\frac{(x-c)^2}{2\sigma^2}}
\]
Moreover, using the Gaussian function character, the particles closed to the global best position are adjusted larger by (15), and the ones far from $P_{\text{ghost}}$ obtain less modification, which continually update through the original (12). What is more, according to (16), with the increase of the iterative steps $k$, the Gaussian function becomes steep. Thus, the amplitude of updating becomes smaller and smaller, which could guarantee the approximation capability of the PSO algorithm in the latter period of optimal progress.

In addition, using the variable $x'_j$, the chaos random mapping is introduced into the trajectory modification, as show in (17) and (18), respectively

$$z^*_j = 4z^*_j(1-z^*_j)$$  \hspace{0.5cm} (17)$$

$$x'_j = x_{\min} + z_j(x_{\max} - x_{\min})$$  \hspace{0.5cm} (18)$$

here $z^*_j$ is a iterative mapping parameter of chaos, $i$ and $n$, denote the dimension and the index of the particle, respectively, $n = 1, 2, \ldots, m$. The initial condition $z^*_j$ is a random number in the range $[0, 1]$ except 0.25, 0.5 and 0.75. Hence, the sequence $z^*_j$ shows the chaotic character, which is better than the uniform distribution in the aspect of travel ergodicity. $x_{\min}$ and $x_{\max}$ are minimum and maximum position for the particles. Through (14) adjustments, the particles of premature convergence are relocated as much as possible in the coverage of the entire solution space. That could maintain the continual particulate global search capability, and avoid trapping in the local extreme points.

IV. NONLINEAR MODEL PREDICTIVE CONTROL BASED ON GAUSSIAN PARTICLE SWARM

PSO can achieve better global optimization results for different functions. Hence, the prediction steps are represented by population particles. Control actions $\Delta u(k) = [\Delta u(k | k), \ldots, \Delta u(k + N_p - 1 | k)]^T \in \mathbb{R}^{n_u}$ to be applied to the system in a specified future time are encoded into corresponding data structures that form the population. $\Delta u_{\min} \leq \Delta u(k) \leq \Delta u_{\max}$ also means $x_{\min} \leq x_{\min} \leq x_{\max} \leq x_{\max}$ in the GPSO. That Utilizing GPSO solves constraint problem (11), the more accuracy control variable can be obtained. Although the optimal parameter vector $\Delta u(k)$ contains many time control variable, with the receding horizon control principle, we only implement the first sample of this sequence, i.e. $\Delta u(k | k) = P_{\text{ghost}}$, where $P_{\text{ghost}}$ is the first position of the best particle in the population. The unknown high order infinitesimal calculation and the whole control scheme are shown as follows.

A. Online-learning of Unknown High Order Infinitesimal

Because the high order infinitesimal part is unknown, the simple version of GPSO is adopted to online estimate. The algorithm is presented as follows.
step 1. Initialization. Let $F(0) = 0$, then calculate $c(0), l_{\min}(0), l_{\max}(0), W$ and $E$;

step 2. Draw from the GPSO position updating strategy. Calculate $F_i(i + 1) = F_i(i) + w(i) \hat{y}_i(i) - F_i(i)$, such as (12).

Adopt cosine dynamic inertia weight $\max_i cos(\frac{i}{\max_i})i$. $i_{\max}$ is the maximum iterative number. $\hat{y}_i$ is the current output of original model;

step 3. Using $F_i(i + 1)$ updates parameters $c(i + 1), l_{\min}(i + 1), l_{\max}(i + 1)$ of GPSO algorithm;

step 4. Solve optimization problem (11) by applying GPSO algorithm, and obtain the control variable $\Delta u_i(i + 1)$;

step 5. If $|F_i(i + 1) - F_i(i)| < \gamma$ or $i \geq i_{\max}$, $\gamma$ is a small positive number, the cycle stops; otherwise, return to step 2 to continue;

B. Control Scheme

The control scheme of NMPC based on GPSO is shown as Fig. 2. Hence, the control process is as follows.

step 1. Let $k = 1$ as sampling time, $N$ is the prediction horizon; $N_u$ is the control horizon, weight matrix $Q, R$;

step 2. Calculate process model matrix and GPSO parameters;

step 3. Employing GPSO to obtain control variable $\Delta u_i(k)$;

step 4. Acquire control input $u(k) = \Delta u(k | k) + u(k - 1)$;

step 5. $k = k + 1$, go back to step 3.

Figure 2. The diagram of GPSO+NMPC

V. BALL-PLATE SYSTEM SIMULATIONS

Ball-Plate system is a learning and research platform for the automatic control, motion control, digital image processing, machine vision and other professional courses. Specifically it is developed for the classical control theory, modern control theory, computer image processing program, which are easy to perform corresponding experiments. The schematic diagram of Ball-Plate system is as shown in Fig. 3.

Figure 3. The diagram of NMPC+GPSO

Ball-Plate system is a multi-variable, non-linear control plant. As a mechanical system with two degrees of freedom, the system state variables are defined as

$$X = (x_1, x_2, \cdots, x_8)^T = (x, x, \theta_1, \theta_1, y, y, \theta_1, \theta_1)^T \quad (19)$$

where $x$ and $y$ are positions of the ball, $\theta_1$ and $\theta_1$ are angle of the plate from x-axis and y-axis, respectively. Therefore, according to the physical energy formula [23], the state equation of Ball-Plate system is expressed as

$$
\begin{bmatrix}
 x_1 \\
 x_2 \\
 x_3 \\
 x_4 \\
 x_5 \\
 x_6 \\
 x_7 \\
 x_8
\end{bmatrix}
= 
B
\begin{bmatrix}
 x_1 x_2^2 + x_2 x_3 x_8 - g \sin x_1 \\
 x_2 \\
 x_3 \\
 0 \\
 x_5 \\
 x_6 \\
 x_7 \\
 x_8
\end{bmatrix}
+ 
\begin{bmatrix}
 0 & 0 \\
 0 & 0 \\
 0 & 0 \\
 0 & 0 \\
 0 & 0 \\
 0 & 0 \\
 0 & 1 \\
 0 & 1
\end{bmatrix}
\begin{bmatrix}
 u_x \\
 u_y
\end{bmatrix}
\quad (20)
$$

here $B = \frac{m}{m + J_b R^2}$, $m$ is weight of the ball, $R$ is radius of the ball, and $J_b$ is mass moment of inertia of the ball. The input-output description is as follows

$$[x, y]^T = g(\theta_1, \theta_1) \quad (21)$$

Hence, Ball-Plate system is a nonlinear system with two inputs and two outputs. The weight matrices are $Q = I$ and $R = I$. Prediction horizon $N = 10$, and control horizon $N_u = 3$, $\gamma = [0.01, \cdots, 0.01]^T$. The reference single is the square single with the constraints $-1 \leq u_x \leq 1; -1 \leq u_y \leq 1$. Fig. 4 shows the better performances in tracking the reference singles. Therefore, the proposed GPSO+NMPC scheme is effective.
VI. CONCLUSION AND FUTURE REMARK

The constraint optimization without any mathematical limitation is able to be solved by proposed heuristic GPSO methods. At each sampling time, the nonlinear system is decomposed to a linear system with unknown high order infinitesimal. Through the online learning based the GPSO evolutionary mechanism, the unknown part is estimated. Simulation experiment of Ball-Plate system verifies that the proposed nonlinear model predictive control approach based on Gaussian particle swarm optimization algorithm can obtain the accurate performance results. The GPSO may cost more time to calculate the optimal control variable compared with other classical methods. However, GPSO can be reformulated to the parallel calculation form with parallel computer hardware like Graphic Process Unit (GPU). Hence, that the heuristic algorithm is employed to the control optimization field is very necessary.

ACKNOWLEDGMENT

The authors would like to thank J. Wang at The Chinese University of Hong Kong for his constructive simulation help and valuable suggestions which had significantly improved the quality of this paper.

REFERENCES