Time Optimal Contouring Control of Industrial Biaxial Gantry: A Highly Efficient Analytical Solution of Trajectory Planning

Mingxing Yuan, Zheng Chen, *Member, IEEE*, Bin Yao, *Senior Member, IEEE*, and Xiaocong Zhu, *Member, IEEE*

***Abstract*—Contouring control is an important issue in industrial applications. For better product quality and higher productivity, achieving higher contour tracking performance with feed rate as large as possible is essential. However, most existing research has focused on improving contour tracking accuracy only. The potential instability issue due to control input saturation under high feed-rate operations has been largely ignored. In this paper, the minimum time trajectory planning problem is formulated as a step toward solving this practically important problem. A novel back and forward check algorithm is developed to take into account physical constraints of the system when generating the desired trajectory to be followed. Compared with traditional numerical searching methods, the proposed algorithm is very computationally efficient because the optimal solution at each node can be obtained by several analytical equations directly. Moreover, the algorithm can incorporate some velocity-dependent constraints easily. Planning a Lissajous curve is used as the case study to verify the optimality and computational efficiency of the proposed algorithm. Experiments are conducted on an industrial biaxial gantry. Several implementation issues on tracking complicated contours are discussed. Experimental results indicate that the proposed trajectory planning approach enables simultaneous achievement of high contouring tracking accuracy and higher feed rate or shorter operation time.**

***Index Terms*—Back and forward check (BFC), contouring control, feed-rate optimization, minimum time, trajectory planning.**

I. Introduction

Contouring tracking control is a critical issue in computer numerical control machining, robotics, and laser cutting industry. Contour error, by definition, is the shortest distance from the actual position to the desired contour. The early control strategy is to reduce the tracking error of each axis individually. However, the contouring error does not always coincide with the axes tracking errors. To deal with the coordination problem among multiaxes, various approaches such as the cross-coupled control schemes [1]–[3], the local task coordinate frame (LTCF) scheme [4]–[6], and the use of polar coordinate frame [7] have been developed over the years.

Contouring error, by definition, should be related to the desired contour itself only, and is thus independent of the motion on the desired contour. However, all the aforementioned contouring control strategies utilized the actual axes tracking errors to estimate the contouring error under the assumption that the tracking errors are much smaller than the radius of curvature of the desired contour. Recognizing the limitation of these algorithms, Yao *et al.* [8] proposed an orthogonal global task coordinate frame (GTCF) for the contouring control problem, and its effectiveness is also confirmed by Hu *et al.* in [9]. With GTCF, the contouring error calculation has nothing to do with the tracking errors of individual axes. Though GTCF is based on the assumption that the desired contour is mathematically defined in an algebraic equation form (i.e., $f(x,y) = 0$), the method is in principle applicable to the desired contour specified in a parameterized form (i.e., $x_d(v)$ and $y_d(v)$) through use of the transformation given in [8].

In the above researches on contouring controls, the reference trajectory to be followed has been assumed to be feasible without causing control input saturations and specific ways to generate such trajectories are not given. Trajectory generation, which is also called trajectory planning, has been widely used in the applications such as machining tools [10], [11] and robotics [12]–[14]. Minimum time and minimum energy-consumption are the main considerations in the cost functions of the trajectory planning problems. In this paper, the contouring tracking accuracy and the productivity efficiency are considered, and thus

Manuscript received December 14, 2015; revised April 19, 2016; accepted June 3, 2016. Date of publication July 18, 2016; date of current version February 14, 2017. Recommended by Technical Editor J. Yu. This work was supported in part by the National Natural Science Foundation of China under Grant 51475412, Grant 51375432 and Grant 61603332, in part by the Science Fund for Creative Research Groups of the National Natural Science Foundation of China under Grant 51521064, in part by the National Basic Research and Development Program of China under 973 Program Grant 2013CB035400, in part by the Project of Key Innovation Team of Zhejiang Province under Grant 2013TD14, and in part by the Fundamental Research Funds for the Central Universities under Grant 2016QNA4038. *(Corresponding author: Zheng Chen.)*

M. Yuan and X. Zhu are with the State Key Laboratory of Fluid Power Transmission and Control, Zhejiang University, Hangzhou 310027, China.

Z. Chen is with the Ocean College, Zhejiang University, Hangzhou 310027, China (e-mail: cwlinus@gmail.com).

B. Yao is with the State Key Laboratory of Fluid Power Transmission and Control, Zhejiang University, Hangzhou 310027, China, and also with the School of Mechanical Engineering, Purdue University, West Lafayette, IN 47907 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMECH.2016.2592518

the cost function with a minimum time form will be adopted in our trajectory planning approach. In the time optimal trajectory planning, the system's physical constraints will be considered. The planned trajectory will make full use of the actuator's capability to obtain feed rate as large as possible, and thus a higher productivity can be achieved. On the other hand, the system's constraints will not be violated such that a good contouring tracking accuracy is guaranteed.

The general time optimal trajectory planning problem can be described as follows: optimizing a minimum time trajectory for a specified geometric path without violating actuators' physical constraints. The early approaches to this problem [12], [15]–[17] were to employ the velocity limit curve (VLC) and switching points. The optimal solution was a bang-bang-type trajectory, which was shown in [12]. The key to obtaining the optimal solution is to search the switching points of the acceleration. This method can indeed provide an optimal solution. However, it seems not easy to obtain the VLC and switching points. To bypass these problems, several different algorithms were developed. Shin and McKay [18] proposed a dynamic programming method to address the minimum time trajectory optimization problem. This approach put few restrictions on the cost function. However, the computation burden of the algorithm may be very large as the phase space dimension increases greatly. Constantinescu *et al.* [19], [20] proposed a smooth and minimum time trajectory planning approach by utilizing the flexible tolerance method and flexible polyhedron method. Zhang *et al.* [21] employed a control vector parametrization method to discretize the problem, and then applied the sequential quadratic programming to obtain the optimal solution. Recently, a two-pass scan algorithm was first proposed by Renton and Elbestawi [22], and then was further developed by Dong and Stori in [23]. Moreover, a detailed proof of the global optimality of the two-pass scan algorithm was presented in [10]. However, all the optimal solutions provided by the algorithms in [10], [19]–[23] were based on different numerical searching methods, which may not be computationally efficient.

Lu and Yao [24] proposed a two-loop trajectory planning approach. However, it was just near-optimal because the parametric acceleration was optimized in the outer loop and stayed unchanged when it was integrated in the inner loop.

In this paper, we propose a new numerical back and forward check (BFC) algorithm to address the minimum time trajectory planning problem by considering the system's velocity and acceleration constraints and integrate the algorithm with GTCF contouring controls in [8]. Compared with the traditional algorithms that utilize numerical searching methods to obtain the optimal solution, the time optimal trajectory planning algorithm proposed in this paper is very computationally efficient because the optimal solution at each node can be obtained by several analytical equations directly. In contrast with our previous conference publication [25], the improvements in this paper include the following.

1) The proofs of the lemmas in [25] have been presented, and the algorithm's implementation is discussed in detail.
2) Several contouring tracking control experiments are carried out on an industrial biaxial gantry to validate the effectiveness of the proposed BFC algorithm.
3) The GTCF proposed in [8] is enhanced to address the the contouring tracking case where the reference contour cannot be described with an algebraic equation analytically.

The contributions of this paper include the following.

1) The proposed algorithm is very computationally efficient and can be implemented easily.
2) The algorithm can incorporate some velocity-dependent constraints expediently.
3) The GTCF approach in [8] is enhanced to address the contouring tracking case in which the given geometric contour may not be described by an algebraic equation analytically.

II. Problem Formulation

In contouring tracking control task, the contour is often formulated as a parameterized form: $\mathbf{P}(u) = [P_x(u), P_y(u)]$, where $u \in [0, 1]$, and u is a function of time t. Denote $\dot{\mathbf{P}}(u) = \frac{d\mathbf{P}(u)}{dt}$, $\mathbf{P}'(u) = \frac{d\mathbf{P}(u)}{du}$, and the actuator's velocity and acceleration can be written as

$$\mathbf{v} = \dot{\mathbf{P}}(u) = \frac{d\mathbf{P}(u)}{du} \cdot \frac{du}{dt} = \mathbf{P}'(u) \cdot \dot{u}, \tag{1}$$

$$\mathbf{a} = \ddot{\mathbf{P}}(u) = \frac{d}{dt}(\mathbf{P}'(u) \cdot \dot{u}) = \mathbf{P}'(u) \cdot \ddot{u} + \mathbf{P}''(u) \cdot \dot{u}^2 \tag{2}$$

where $\dot{u}$ and $\ddot{u}$ are called pseudovelocity and pseudoacceleration, respectively. Then the minimum time trajectory planning problem can be formulated as

$$\min_{\dot{u}} \int_{t_0}^{t_f} \mathbf{1} dt \tag{3}$$

which is subject to

$$\begin{cases} -\mathbf{v}_{\max} \leq \mathbf{v} \leq \mathbf{v}_{\max} \\ -\mathbf{a}_{\max} \leq \mathbf{a} \leq \mathbf{a}_{\max} \\ \mathbf{v}(t_0) = \mathbf{0}, \mathbf{v}(t_f) = \mathbf{0} \end{cases} \tag{4}$$

where t_0 is the starting time. Without generality, we assume that $t_0 = 0$. t_f is the final time. $\mathbf{v}_{\max} = [v_{x\max}, v_{y\max}]$, $\mathbf{a}_{\max} = [a_{x\max}, a_{y\max}]$. Note that the final point is free in the time domain, and thus it becomes difficult to solve the optimization problem directly. To overcome the difficulty, we make a simple transformation and reformulate the objective function as

$$\min_{\dot{u}} \int_0^1 \frac{1}{\dot{u}} du. \tag{5}$$

Therefore, (5) shows that the pseudovelocity $\dot{u}$, an indicator of feed rate, should be as large as possible without violating the constraints (4).

III. BFC Algorithm

In this section, we propose a BFC algorithm to solve the problem (5) under the constraints (4). The BFC algorithm is a numerical approach with the parameter u discretized equidistantly as $0 = u_0 < u_1 < u_2 \cdots < u_{N-1} < u_N = 1$. Moreover, we assume that the pseudoacceleration $\ddot{u}_i$ is a constant in the interval $[u_{i-1}, u_i]$, and thus,

$$\ddot{u}_i = \frac{\dot{u}_i^2 - \dot{u}_{i-1}^2}{2 \cdot \Delta u} \tag{6}$$

where $\Delta u = u_i - u_{i-1} = \frac{1}{N}$. To obtain the solution to the problem (5), we just need to find the maximum feasible $\dot{u}_i$ at each node u_i under the constraints (4).

Based on (2), we can derive the bounds of the pseudoacceleration in the interval $[u_{i-1}, u_i]$ as

$$\ddot{u}_{\max}(u_{i-1}, \dot{u}_{i-1}) = \min\left\{ \frac{a_{x\max}}{|P_x'(u_{i-1})|} - \frac{P_x''(u_{i-1})}{P_x'(u_{i-1})}\dot{u}_{i-1}^2, \right.$$
$$\left. \frac{a_{y\max}}{|P_y'(u_{i-1})|} - \frac{P_y''(u_{i-1})}{P_y'(u_{i-1})}\dot{u}_{i-1}^2 \right\}, \tag{7}$$

$$\ddot{u}_{\min}(u_{i-1}, \dot{u}_{i-1}) = \max\left\{ - \frac{a_{x\max}}{|P_x'(u_{i-1})|} - \frac{P_x''(u_{i-1})}{P_x'(u_{i-1})}\dot{u}_{i-1}^2, \right.$$
$$\left. - \frac{a_{y\max}}{|P_y'(u_{i-1})|} - \frac{P_y''(u_{i-1})}{P_y'(u_{i-1})}\dot{u}_{i-1}^2 \right\}. \tag{8}$$

At the node u_i, the maximum candidate pseudovelocity can be computed based on (1) as follows:

$$\dot{u}_{i\_\max\_c} = \min\left\{ \frac{v_{x\max}}{|P_x'(u_i)|}, \frac{v_{y\max}}{|P_y'(u_i)|} \right\}. \tag{9}$$

It is obvious that the pseudovelocity at the node u_i is subject to the following constraint:

$$0 \leq \dot{u}_i \leq \dot{u}_{i\_\max\_c}. \tag{10}$$

Therefore, the actual pseudoacceleration $\ddot{u}_i$ in the interval $[u_{i-1}, u_i]$ varies in the following range:

$$\ddot{u}_i = \frac{\dot{u}_i^2 - \dot{u}_{i-1}^2}{2\Delta u} \in \left[-\frac{\dot{u}_{i-1}^2}{2\Delta u}, \frac{\dot{u}_{i\_\max\_c}^2 - \dot{u}_{i-1}^2}{2\Delta u} \right]$$
$$= [\ddot{u}_{i\min}, \ddot{u}_{i\max}]. \tag{11}$$

The basic idea of the BFC algorithm is to check the relationship between the sets $[\ddot{u}_{i\min}, \ddot{u}_{i\max}]$ and $[\ddot{u}_{\min}(u_{i-1}, \dot{u}_{i-1}), \ddot{u}_{\max}(u_{i-1}, \dot{u}_{i-1})]$.

Lemma 1: If $\ddot{u}_{\max}(u_{i-1}, \dot{u}_{i-1}) < \ddot{u}_{i\min} = -\frac{\dot{u}_{i-1}^2}{2\Delta u}$, then there are no intersections between the sets $[\ddot{u}_{i\min}, \ddot{u}_{i\max}]$ and $[\ddot{u}_{\min}(u_{i-1}, \dot{u}_{i-1}), \ddot{u}_{\max}(u_{i-1}, \dot{u}_{i-1})]$.

1) The only way to make the two sets intersect is to reduce $\dot{u}_{i-1}$.
2) There must exist a feasible $\dot{u}_{i-1}'$ s.t. $\ddot{u}_{i\min}' = -\frac{\dot{u}_{i-1}'^2}{2\Delta u} \leq \ddot{u}_{\max}(u_{i-1}, \dot{u}_{i-1}')$. The maximum feasible $\dot{u}_{i-1}'$ is computed as

$$\dot{u}_{i-1}' = \sqrt{\frac{\frac{a_{k\max}}{|P_k'(u_{i-1})|}}{\left|\frac{1}{2\Delta u} - \frac{P_k''(u_{i-1})}{P_k'(u_{i-1})}\right|}}, (k \text{ is } x \text{ or } y). \tag{12}$$

3) The pseudovelocity at the node u_i is then determined as

$$\dot{u}_i = \sqrt{\dot{u}_{i-1}'^2 + 2\Delta u \cdot \ddot{u}_{\max}(u_{i-1}, \dot{u}_{i-1}')}. \tag{13}$$

Proof: $\ddot{u}_{\max}(u_{i-1}, \dot{u}_{i-1}) < \ddot{u}_{i\min} = -\frac{\dot{u}_{i-1}^2}{2\Delta u}$ implies

$$\ddot{u}_{\max}(u_{i-1}, \dot{u}_{i-1}) - \ddot{u}_{i\min}$$
$$= \frac{a_{k\max}}{|P_k'(u_{i-1})|} + \left(\frac{1}{2\Delta u} - \frac{P_k''(u_{i-1})}{P_k'(u_{i-1})}\right)\dot{u}_{i-1}^2 < 0. \tag{14}$$

It is clear that the following inequality holds

$$\frac{1}{2\Delta u} - \frac{P_k''(u_{i-1})}{P_k'(u_{i-1})} < 0. \tag{15}$$

Denote

$$f_1(u_{i-1}, \dot{u}_{i-1}) = \frac{a_{k\max}}{|P_k'(u_{i-1})|} + \left(\frac{1}{2\Delta u} - \frac{P_k''(u_{i-1})}{P_k'(u_{i-1})}\right)\dot{u}_{i-1}^2 \tag{16}$$

then $f_1(u_{i-1}, \dot{u}_{i-1})$ is a monotonically decreasing function with respect to $\dot{u}_{i-1}$. Thus, there must exist a feasible $\dot{u}_{i-1}'$ that is smaller than $\dot{u}_{i-1}$ s.t. $f_1(u_{i-1}, \dot{u}_{i-1}') = \ddot{u}_{\max}(u_{i-1}, \dot{u}_{i-1}') - \ddot{u}_{i\min}' \geq 0$ based on the fact that $f_1(u_{i-1}, \dot{u}_{i-1} = 0) > 0$. Therefore, the problem must become feasible by reducing $\dot{u}_{i-1}$. The maximum candidate feasible $\dot{u}_{i-1}'$, as shown in (12), is then obtained by letting $\ddot{u}_{\max}(u_{i-1}, \dot{u}_{i-1}') = \ddot{u}_{i\min}'$.

The result expressed in (13) is evident. ■

Remark 1: The $\dot{u}_{i-1}'$ in (12) is determined by the *X*-axis or *Y*-axis. The computation procedure of $\dot{u}_{i-1}'$ is presented in detail as follows.

Step 1: Calculate the $\dot{u}_{i-1}'$ in (12) for the *X*-axis and *Y*-axis, respectively. Denote the calculation results as $\dot{u}_{i-1}^x$ for *X*-axis and $\dot{u}_{i-1}^y$ for *Y*-axis.

Step 2: Substitute $\dot{u}_{i-1}^k$ (k is x or y) into (7) and obtain the maximum pseudoacceleration $\ddot{u}_{\max}^k(u_{i-1}, \dot{u}_{i-1}^k)$. We call that $\dot{u}_{i-1}^k$ is feasible if $\ddot{u}_{\max}^k(u_{i-1}, \dot{u}_{i-1}^k)$ is determined by the k (k is X or Y) axis, that is, if k is x, for example, and $\frac{a_{x\max}}{|P_x'(u_{i-1})|} - \frac{P_x''(u_{i-1})}{P_x'(u_{i-1})} \cdot (\dot{u}_{i-1}^x)^2 \leq \frac{a_{y\max}}{|P_y'(u_{i-1})|} - \frac{P_y''(u_{i-1})}{P_y'(u_{i-1})} \cdot (\dot{u}_{i-1}^x)^2$, then $\ddot{u}_{\max}^x(u_{i-1}, \dot{u}_{i-1}^x)$ is determined by the *X*-axis, and thus $\dot{u}_{i-1}^x$ is feasible.

Step 3:

Case 1: $\dot{u}_{i-1}^x$ is feasible, and $\dot{u}_{i-1}^y$ is infeasible, then $\dot{u}_{i-1}' = \dot{u}_{i-1}^x$.

Case 2: $\dot{u}_{i-1}^y$ is feasible, and $\dot{u}_{i-1}^x$ is infeasible, then $\dot{u}_{i-1}' = \dot{u}_{i-1}^y$.

Case 3: Both $\dot{u}_{i-1}^x$ and $\dot{u}_{i-1}^y$ are feasible, then $\dot{u}_{i-1}' = \max\{\dot{u}_{i-1}^x, \dot{u}_{i-1}^y\}$.

Lemma 2: If $\ddot{u}_{\min}(u_{i-1}, \dot{u}_{i-1}) > \ddot{u}_{i\max} = \frac{\dot{u}_{i\_\max\_c}^2 - \dot{u}_{i-1}^2}{2\Delta u}$, then there are no intersections between the sets $[\ddot{u}_{i\min},\ddot{u}_{i\max}]$ and $[\ddot{u}_{\min}(u_{i-1}, \dot{u}_{i-1}),\ddot{u}_{\max}(u_{i-1}, \dot{u}_{i-1})]$.

1) The only way to make the two sets intersect is to reduce $\dot{u}_{i-1}$.
2) There must exist a $\dot{u}_{i-1}'$, s.t. $\ddot{u}_{\min}(u_{i-1}, \dot{u}_{i-1}') \leq \ddot{u}_{i\max}' = \frac{\dot{u}_{i\_\max\_c}^2 - \dot{u}_{i-1}'^2}{2\Delta u}$. The maximum feasible $\dot{u}_{i-1}'$ is computed as

$$\dot{u}_{i-1}' = \sqrt{\frac{\frac{a_{k\max}}{|P_k'(u_{i-1})|} + \frac{\dot{u}_{i\_\max\_c}^2}{2\Delta u}}{\left|\frac{1}{2\Delta u} - \frac{P_k''(u_{i-1})}{P_k'(u_{i-1})}\right|}}, (k \text{ is } x \text{ or } y). \tag{17}$$

3) The pseudovelocity at the node u_i is determined as

$$\dot{u}_i = \dot{u}_{i\_\max\_c}. \tag{18}$$

Proof: The method of proving Lemma 2 is similar to that of Lemma 1, and is omitted here. ■

The computation process of $\dot{u}'_{i-1}$ in (17) is similar to the procedure presented in Remark 1, and is omitted here.

Algorithm 1: The back and forward check algorithm.

```
Initialize $\dot{u}_i = 0, i = 0, 1, \ldots, N$;
while $u_i \leq u_N$ do
    if $u_i < u_N$ then
        Compute $\dot{u}_{i\_\max\_c}$, $\ddot{u}_{\max}$ and $\ddot{u}_{\min}$;
        if $\ddot{u}_{\max}(u_{i-1}, \dot{u}_{i-1}) < \ddot{u}_{\min}(u_{i-1}, \dot{u}_{i-1})$ then
            Determine $\dot{u}'_{i-1}$ by (20);
            Execute the check back operations;
        end if
        Check the relationship between the sets
        $[\ddot{u}_{i\min}, \ddot{u}_{i\max}]$ and
        $[\ddot{u}_{\min}(u_{i-1}, \dot{u}_{i-1}), \ddot{u}_{\max}(u_{i-1}, \dot{u}_{i-1})]$;
        if $\ddot{u}_{\max}(u_{i-1}, \dot{u}_{i-1}) < \ddot{u}_{i\min} = -\frac{\dot{u}_{i-1}^2}{2\Delta u}$ then
            Determine $\dot{u}'_{i-1}$ and $\dot{u}_i$ based on (12) and (13),
            respectively;
        else if $\ddot{u}_{\min}(u_{i-1}, \dot{u}_{i-1}) > \ddot{u}_{i\max} = \frac{\dot{u}_{i\_max\_c}^2 - \dot{u}_{i-1}^2}{2\Delta u}$
        then
            Determine $\dot{u}'_{i-1}$ and $\dot{u}_i$ based on (17) and (18),
            respectively;
        else if
        $\ddot{u}_{\min}(u_{i-1}, \dot{u}_{i-1}) \leq \ddot{u}_{i\max} \leq \ddot{u}_{\max}(u_{i-1}, \dot{u}_{i-1})$
        then
            $\dot{u}_i = \dot{u}_{i\_\max\_c}$;
        else
            $\dot{u}_i = \sqrt{\dot{u}_{i-1}^2 + 2\Delta u \cdot \ddot{u}_{\max}(u_{i-1}, \dot{u}_{i-1})}$;
        end if
        if $\dot{u}_{i-1}$ has reduced to $\dot{u}'_{i-1}$ then
            Execute the check back operations;
        end if
    else
        $\dot{u}_N = 0$;
        Execute the check back operations.
    end if
    $u_i = u_i + \Delta u$;
end while
```

Note that a reduced $\dot{u}'_{i-1}$ may make the problem infeasible in the interval $[u_{i-2}, u_{i-1}]$, although the original combination of $\dot{u}_{i-1}$ and $\dot{u}_{i-2}$ is feasible in the interval $[u_{i-2}, u_{i-1}]$. Therefore, we must conduct an additional operation that checks whether $\dot{u}_{i-2}$ and $\dot{u}'_{i-1}$ are feasible or not in the interval $[u_{i-2}, u_{i-1}]$, and this check back operation should continue until a feasible combination of $\dot{u}_{k-1}$ and $\dot{u}'_k$ $(k \leq i-1)$ is found, as summarized by the following lemma.

Lemma 3: Assume that the original combination of $\dot{u}_{i-2}$ and $\dot{u}_{i-1}$ is feasible in the interval $[u_{i-2}, u_{i-1}]$, but the combination of $\dot{u}_{i-2}$ and a reduced $\dot{u}'_{i-1}$ becomes infeasible. Then, 1) the only way to make the problem feasible in the interval $[u_{i-2}, u_{i-1}]$ is to reduce $\dot{u}_{i-2}$ to some $\dot{u}'_{i-2}$; 2) there must exist a $\dot{u}'_{i-2}$ that makes the problem feasible, and the maximum feasible $\dot{u}'_{i-2}$ is determined as

$$\dot{u}'_{i-2} = \sqrt{\frac{\frac{a_{k\max}}{|P'_k(u_{i-2})|} + \frac{\dot{u}'^2_{i-1}}{2\Delta u}}{\left|\frac{1}{2\Delta u} - \frac{P''_k(u_{i-2})}{P'_k(u_{i-2})}\right|}}, (k \text{ is } x \text{ or } y); \tag{19}$$

and 3) further check back operations may be necessary after $\dot{u}_{i-2}$ is reduced to $\dot{u}'_{i-2}$. However, there must exist a node u_m $(m \leq i-2)$ at which the combination of $\dot{u}_{m-1}$ and $\dot{u}'_m$ is feasible.

Proof: The proofs of the first conclusion and the third one are similar to those in [11], and are omitted here.

The $\dot{u}'_{i-2}$ in (19) is obtained by simply letting $\ddot{u}'_{i-1} = \ddot{u}_{\min}(u_{i-2}, \dot{u}'_{i-2})$. ■

The computation process of $\dot{u}'_{i-2}$ in (19) is similar to the procedure presented in Remark 1, and is omitted here.

All the discussions above are based on the assumption that $\ddot{u}_{\max}(u_{i-1}, \dot{u}_{i-1}) \geq \ddot{u}_{\min}(u_{i-1}, \dot{u}_{i-1})$. However, this assumption may not be true all the time.

Lemma 4: If $\ddot{u}_{\max}(u_{i-1}, \dot{u}_{i-1}) < \ddot{u}_{\min}(u_{i-1}, \dot{u}_{i-1})$, then 1) $\ddot{u}_{\max}(u_{i-1}, \dot{u}'_{i-1})$ will eventually be no smaller than $\ddot{u}_{\min}(u_{i-1}, \dot{u}'_{i-1})$ by reducing $\dot{u}_{i-1}$ and 2) the maximum feasible $\dot{u}'_{i-1}$ is computed as

$$\dot{u}'_{i-1} = \sqrt{\frac{\frac{a_{x\max}}{|P'_x(u_{i-1})|} + \frac{a_{y\max}}{|P'_y(u_{i-1})|}}{\left|\frac{P''_x(u_{i-1})}{P'_x(u_{i-1})} - \frac{P''_y(u_{i-1})}{P'_y(u_{i-1})}\right|}}. \tag{20}$$

Proof: The proof of the first conclusion is similar to the proof method in [11], and is omitted here.

The maximum feasible $\dot{u}'_{i-1}$, as shown in (20), is obtained by letting $\ddot{u}_{\max}(u_{i-1}, \dot{u}'_{i-1}) = \ddot{u}_{\min}(u_{i-1}, \dot{u}'_{i-1})$. ■

Based on the aforementioned lemmas and discussions, we provide the detailed computation procedure of the proposed minimum time trajectory planning in Algorithm 1.

Remark 2: The proposed BFC algorithm can incorporate some additional velocity-dependent constraints easily. We can add the additional velocity-dependent constraints into (9) directly, and thus a conservative $\dot{u}_{i\_\max\_c}$ will be obtained.

Note that in the actual implementation of the BFC algorithm, a larger N means the trajectory is discretized into more segments, and thus a smoother trajectory can be achieved. On the other hand, a larger N means longer computation time and requires more memory space. Therefore, an appropriate N should be selected.

IV. Computation Results

In this section, to verify the effectiveness of the proposed trajectory planning algorithm, a Lissajous curve is selected as the case study. In the simulation of the BFC trajectory planning algorithm, both of the velocity bounds for the *X*-axis and *Y*-axis are set as 1.5 m/s, and for the acceleration limits, 40 m/s^2 for the *X*-axis and 9 m/s^2 for the *Y*-axis. The proposed BFC trajectory planning algorithm is implemented on the platform VC++ 6.0, which runs on a personal PC with Intel Core i3-3220 3.3 GHz processor, and a 64-bit Windows system.

In the computation results, the VLC employed in [12] will be utilized to further validate the effectiveness of the proposed trajectory planning algorithm. It should be noted that, however, the BFC trajectory planning algorithm proposed in this paper does not need to employ this concept in the actual implementation.

Furthermore, in this section, an extended BFC algorithm incorporating some additional velocity-dependent constraints is presented. The selected additional velocity-dependent constraints can be formulated as

$$\mathbf{v} \notin [\mathbf{v}_l, \mathbf{v}_h], \quad \text{if } u_i \in [u_l, u_h] \tag{21}$$

that is, the velocities of the X-axis and the Y-axis should not be in a predefined range, which may be used to avoid resonance. The $\mathbf{v}_l$ and $\mathbf{v}_h$ in (21) are the lower and upper bound of the predefined range, and we choose the same values for both the X-axis and the Y-axis in this case. The extended constraints of the problem can be represented as a combination of (4) and (21).

Note that when considering the additional velocity-dependent constraints, the trajectory will be first planned based on Algorithm 1. However, if the additional constraint (21) is violated, i.e., $\mathbf{v} \in [\mathbf{v}_l, \mathbf{v}_h]$ for some $u_i \in [u_l, u_h]$, the maximum candidate pseudovelocity in (9) will be revised as

$$\dot{u}'_{i\_\max\_c} = \min\left\{\frac{v_l}{|P'_x(u_i)|}, \frac{v_l}{|P'_y(u_i)|}\right\}. \tag{22}$$

The extended BFC algorithm is then given in Algorithm 2.

Algorithm 2: The extended back and forward check algorithm.

```
Initialize $\dot{u}_i = 0, i = 0, 1, \ldots, N$;
while $u_i \leq u_N$ do
    if $u_i < u_N$ then
        Compute $\ddot{u}_{\max}(u_{i-1}, \dot{u}_{i-1})$ and $\ddot{u}_{\min}(u_{i-1}, \dot{u}_{i-1})$;
        if the velocity of $X$-axis or $Y$-axis falls into the
        predefined range then
            Determine $\dot{u}'_{i\_\max\_c}$ based on (22);
        else
            Determine $\dot{u}_{i\_\max\_c}$ based on (9);
        end if
        Execute the Step 5 to Step 21 in Algorithm 1;
    else
        $\dot{u}_N = 0$;
        Execute the check back operations.
    end if
    if the velocity of $X$-axis or $Y$-axis falls into the
    predefined range then
        Find the smallest node $u_k$ at which the velocity of
        $X$-axis or $Y$-axis falls into the predefined range;
        $u_i = u_k$;
    else
        $u_i = u_i + \Delta u$;
    end if
end while
```

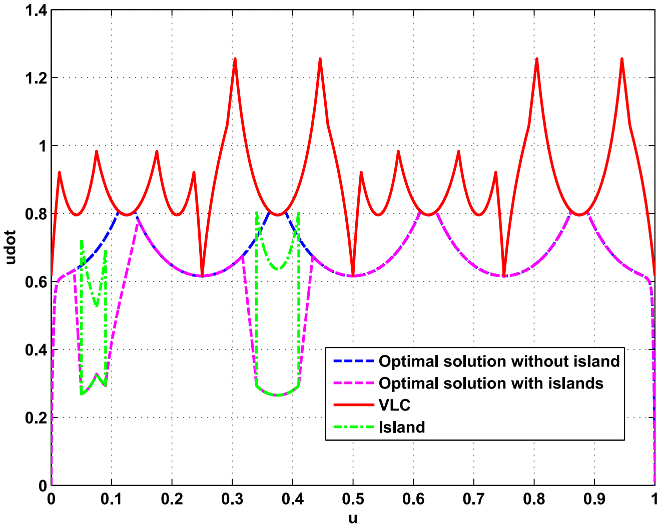


Fig. 1. Pseudovelocity and VLC of the Lissajous curve.

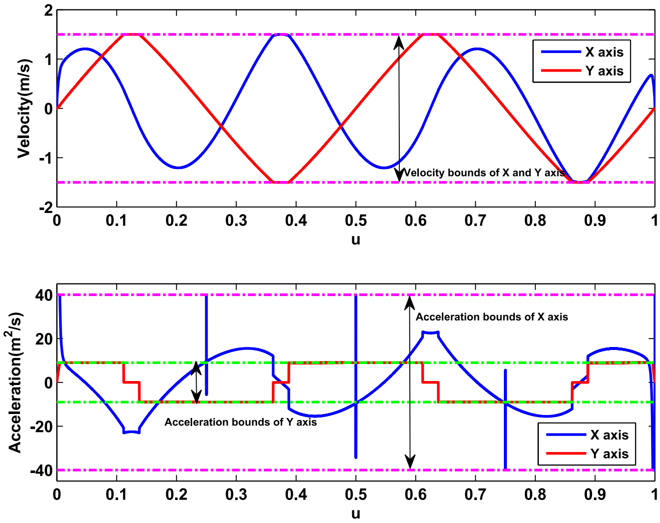


Fig. 2. Velocities and accelerations of the Lissajous curve without additional velocity-dependent constraints.

In this case, the selected Lissajous curve shown in Fig. 7 is described as $\mathbf{P}(u) = [0.1\{\cos\frac{\pi}{4} - \cos(6\pi u + \frac{\pi}{4})\}, 0.15\{1 - \cos(4\pi u)\}]$. To validate the capability of the proposed trajectory planning algorithm to incorporate velocity-dependent constraints, the following additional velocity dependent constraints are assumed as well:

$$\begin{cases} \mathbf{v} \notin [0.5, 0.8], & \text{if } u \in [0.05, 0.09] \\ \mathbf{v} \notin [0.5, 1.2], & \text{if } u \in [0.34, 0.41] \end{cases}. \tag{23}$$

With these additional velocity-dependent constraints, two infeasible islands appear in the phase graph of $u - \dot{u}$, as shown in Fig. 1.

Fig. 1 shows the pseudovelocity curve and the VLC. The optimized velocity and acceleration curves without and with the additional velocity-dependent constraints are shown in Figs. 2 and 3, respectively. As seen from Fig. 1, both of the optimal solutions with and without the additional velocity-dependent

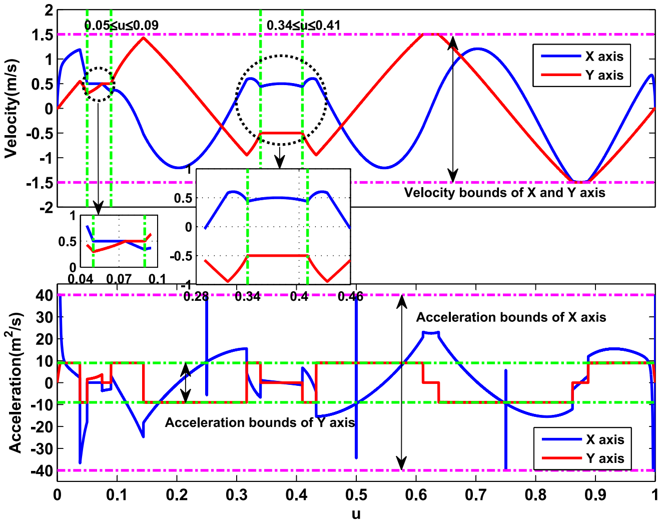


Fig. 3. Velocities and accelerations of the the Lissajous curve with additional velocity-dependent constraints.

TABLE I
COMPUTATION TIME OF THE BFC ALGORITHM, LISSAJOUS CURVE (SET1:WITHOUT ADDITIONAL CONSTRAINTS; SET2: WITH ADDITIONAL CONSTRAINTS)

	$\Delta u = 0.01$	$\Delta u = 0.001$	$\Delta u = 0.0001$
Set1 (ms)	1	48	4741
Set2 (ms)	1	42	4144

constraints are bounded by the VLC all the time. Therefore, no velocity or acceleration constraints are violated. Furthermore, it is observed that the optimal solution to the case where the additional velocity-dependent constraints are considered can bypass the infeasible islands, and these conclusions can be further verified in Figs. 2 and 3. Note that in Fig. 3, the magnified regions indicate that the trajectory planned by Algorithm 2 can indeed satisfy the constraints shown in (23).

Moreover, the necessary condition for the optimality of the time optimal trajectory planning algorithm can be described as follows. *At least one axis operates on its capability limit at every moment* [26]. From Figs. 2 and 3, it is observed that at least one axis achieves its physical bound at each moment, which confirms the optimality of the resulting trajectories. Therefore, the proposed trajectory planning algorithm is feasible and time optimal.

Additionally, the results of the computation time for the Lissajous curve with and without the additional velocity-dependent constraints are shown in Table I. From Table I, the execution time of the proposed trajectory planning algorithm is quite short, and thus the proposed trajectory planning algorithm is very computationally efficient.

V. IMPLEMENTATION ISSUE OF GTCF

In the contouring tracking task, a specified geometric path is given. By employing the proposed trajectory planning algorithm, a reference trajectory with respect to time t can be obtained, which is then fed into the contouring tracking controller. In this paper, GTCF proposed in [8] will be utilized to calculate the contour error.

In GTCF [8], the contour is described in an algebraic equation form of $f(x,y)=0$. However, in many contouring tracking cases, the desired contour is specified by a parameterized form, i.e., $\mathbf{q}_v(v)=[x_v(v),y_v(v)]^T$. For these cases, the function f can be obtained based on the approach in [8]. Define a function $g_v(v)$ by

$$g_v(v)=g(x_v(v),y_v(v)),\forall v \tag{24}$$

where $g(x,y)$ is an arbitrary smooth function. Then the function f can be defined as

$$f(x,y)=g(x,y)-g_v(v_x(x)) \tag{25}$$

or

$$f(x,y)=g(x,y)-g_v(v_y(y)) \tag{26}$$

where $v_x(x)$ and $v_y(y)$ are the inverse function of $x_v(v)$ and $y_v(v)$, respectively. With the function $f(x,y)$, the curvilinear coordinate $r_c(x,y)$ in GTCF that characterizes the contouring error is defined as

$$r_c(x,y)=\frac{1}{\sqrt{f_x^2+f_y^2}}f(x,y). \tag{27}$$

Note that $r_c(x,y)=0$ represents the desired contour. The other orthogonal curvilinear coordinate $r_m(x,y)$ in GTCF that characterizes the motion on the desired contour is defined as

$$r_m(x,y)=s\left(v_x\left(x-\frac{f_x}{f_x^2+f_y^2}f(x,y)\right)\right) \tag{28}$$

or

$$r_m(x,y)=s\left(v_y\left(y-\frac{f_y}{f_x^2+f_y^2}f(x,y)\right)\right) \tag{29}$$

where s is the curve length function from the initial point $\mathbf{q}_v(0)$ to any point on the desired contour $\mathbf{q}_v(v)$, and is defined as

$$s(v)=\int_0^v \sqrt{x_v'(\tau)^2+y_v'(\tau)^2}d\tau. \tag{30}$$

As shown in [8], on the desired contour, the directional vector along the first curvilinear coordinate $r_c(x,y)$ is given by

$$\mathbf{i}_{r_c}(x,y)=\frac{1}{\sqrt{f_x^2+f_y^2}}\begin{bmatrix} f_x \\ f_y \end{bmatrix}, \tag{31}$$

which is exactly the unity normal vector of the desired contour. The unity tangent vector of the desired contour is given by

$$\mathbf{t}_{P_c}(x,y)=\frac{1}{\sqrt{f_x^2+f_y^2}}\begin{bmatrix} f_y \\ -f_x \end{bmatrix}. \tag{32}$$

It is shown in [8] that, on the desired contour, the direction of the curvilinear coordinate $r_m(x,y)$ is exactly along $\mathbf{t}_{P_c}$ (or negative). In fact, it is easy to verify that, with the parametrized form of (28) when $x_v(v)$ is a monotonic function

$$\mathbf{i}_{r_m}=\text{sign}(f_y)\text{sign}(x_v')\mathbf{t}_{P_c} \tag{33}$$

and with the parametrized form of (29) when $y_v(v)$ is a monotonic function

$$\mathbf{i}_{r_m} = -\text{sign}(f_x)\text{sign}(y_v')\mathbf{t}_{P_c}. \tag{34}$$

According to differential geometry, the tangent vector at a certain point of a contour described in a parameterized form $\mathbf{q}_v(v) = [x_v(v), y_v(v)]^T$ is given by

$$\mathbf{t}(v) = [x_v'(v), y_v'(v)]^T. \tag{35}$$

By the right-hand rule, the normal vector $\mathbf{n}(v)$ is defined by a 90° counterclockwise rotation of $\mathbf{t}(v)$, and thus is given by

$$\mathbf{n}(v) = [-y_v'(v), x_v'(v)]^T. \tag{36}$$

To make sure that signs of the contouring error and the tracking error along the desired contour are correctly defined, the directions of $\mathbf{i}_{r_c}$ and $\mathbf{i}_{r_m}$ should coincide with $\mathbf{n}(v)$ and $\mathbf{t}(v)$, respectively. Conditions to guarantee these are summarized in the following two theorems.

Theorem 1: If $g(x, y)$ is a first-order function with respect to x and y, i.e., $g(x, y) = \alpha x + \beta y$, where α and β are constants, the direction of the curvilinear coordinate $r_m(x, y)$ in GTCF, $\mathbf{i}_{r_m}$, always coincides with that of the contour's tangent vector $\mathbf{t}(v) = [x_v'(v), y_v'(v)]^T$.

Proof: If $x_v(v)$ is a monotonic function, we can get the following representation of $f(x, y)$ based on (24) and (25)

$$f(x, y) = g(x, y) - g\left(x_v(v_x), y_v(v_x)\right) \tag{37}$$

where v_x, a function with respect to x, is the inverse function of $x_v(v)$. By chain rule, the partial derivatives of $f(x, y)$ is given by

$$\begin{aligned} f_x = g_x - g_x|_{x=x_v} \cdot \frac{dx}{dv}|_{v=v_x(x)} \cdot \frac{dv_x}{dx} - g_y|_{y=y_v} \\ \cdot \frac{dy}{dv}|_{v=v_x(x)} \cdot \frac{dv_x}{dx} \end{aligned} \tag{38}$$

$$\begin{aligned} f_y = g_y - g_x\Big|_{x=x_v} \cdot \frac{dx}{dv}\Big|_{v=v_x(x)} \cdot \frac{dv_x}{dy} - g_y\Big|_{y=y_v} \\ \cdot \frac{dy}{dv}\Big|_{v=v_x(x)} \cdot \frac{dv_x}{dy} \end{aligned} \tag{39}$$

where $x_v = x_v(v_x(x))$ and $y_v = y_v(v_x(x))$. It is evident that the following two equations hold

$$g_x = g_x|_{x=x_v} = \alpha \tag{40}$$

$$g_y = g_y|_{y=y_v} = \beta. \tag{41}$$

Furthermore, note the following two facts that

$$\frac{dx}{dv}|_{v=v_x(x)} \cdot \frac{dv_x}{dx} = 1 \tag{42}$$

$$\frac{dv_x}{dy} = 0. \tag{43}$$

Based on (40)–(43), (38) and (39) become

$$f_x = -g_y \cdot \frac{y_v'}{x_v'} \tag{44}$$

and

$$f_y = g_y \tag{45}$$

where $y_v' = \frac{dy}{dv}|_{v=v_x(x)}$, and $x_v' = \frac{dx}{dv}|_{v=v_x(x)}$. Substitute (44) and (45) into (33), we can get

$$\mathbf{i}_{r_m} = \frac{1}{\sqrt{(x_v')^2 + (y_v')^2}} \begin{bmatrix} x_v' \\ y_v' \end{bmatrix} = \frac{|g_y|}{|x_v'|\sqrt{f_x^2 + f_y^2}}\mathbf{t}(v) \tag{46}$$

which shows that the direction of the curvilinear coordinate $\mathbf{i}_{r_m}$ is coincident with that of the contour's tangent vector $\mathbf{t}(v)$.

If $y_v(v)$ is a monotonic function, by utilizing the same method above, the partial derivatives are computed as

$$f_x = g_x \tag{47}$$

$$f_y = -g_x \cdot \frac{x_v'}{y_v'} \tag{48}$$

and the direction of the curvilinear coordinate $\mathbf{i}_{r_m}$ is obtained as

$$\mathbf{i}_{r_m} = \frac{1}{\sqrt{(x_v')^2 + (y_v')^2}} \begin{bmatrix} x_v' \\ y_v' \end{bmatrix} = \frac{|g_x|}{|y_v'|\sqrt{f_x^2 + f_y^2}}\mathbf{t}(v). \tag{49}$$

Equations (46) and (49) show that the direction of the curvilinear coordinate $r_m(x, y)$ in GTCF, $\mathbf{i}_{r_m}$, is always coincident with that of the contour's tangent vector $\mathbf{t}(v) = [x_v'(v), y_v'(v)]^T$, which completes the proof. ■

Theorem 2: If $g(x, y)$ is a first-order function with respect to x and y, i.e., $g(x, y) = \alpha x + \beta y$, where α and β are constants, the direction of the curvilinear coordinate $r_c(x, y)$ in GTCF, $\mathbf{i}_{r_c}$, is the same as that of the contour's normal vector $\mathbf{n}(v) = [-y_v'(v), x_v'(v)]^T$ if one of the following conditions is satisfied.

1) Sign$(g_y) \cdot$ sign$(x_v'(v)) = 1$ if $x_v(v)$ is a monotonic function.
2) Sign$(g_x) \cdot$ sign$(y_v'(v)) = -1$ if $y_v(v)$ is a monotonic function.

Proof: If $x_v(v)$ is a monotonic function, then substitute (44) and (45) into (31), we get

$$\mathbf{i}_{r_c}(x, y) = \text{sign}(g_y)\text{sign}(x_v')\frac{|g_y|}{|x_v'|\sqrt{f_x^2 + f_y^2}} \begin{bmatrix} -y_v' \\ x_v' \end{bmatrix}. \tag{50}$$

Based on the condition sign$(g_y) \cdot$ sign$(x_v'(v)) = 1$, (50) becomes

$$\mathbf{i}_{r_c}(x, y) = \frac{1}{\sqrt{(x_v')^2 + (y_v')^2}} \begin{bmatrix} -y_v' \\ x_v' \end{bmatrix} = \frac{|g_y|}{|x_v'|\sqrt{f_x^2 + f_y^2}}\mathbf{n}(v) \tag{51}$$

which shows that the direction of the curvilinear coordinate $r_c(x, y)$ is coincident with the contour's normal vector $\mathbf{n}(v)$.

If $y_v(v)$ is a monotonic function, by employing the same method, it is easy to obtain $\mathbf{i}_{r_c}$ as

$$\mathbf{i}_{r_c}(x, y) = -\text{sign}(g_x)\text{sign}(y_v')\frac{|g_x|}{|y_v'|\sqrt{f_x^2 + f_y^2}} \begin{bmatrix} -y_v' \\ x_v' \end{bmatrix}. \tag{52}$$

Based on the condition $\text{sign}(g_x) \cdot \text{sign}(y_v'(v)) = -1$, (52) becomes

$$\mathbf{i}_{r_c}(x,y) = \frac{1}{\sqrt{(x_v')^2 + (y_v')^2}} \begin{bmatrix} -y_v' \\ x_v' \end{bmatrix} = \frac{|g_x|}{|y_v'|\sqrt{f_x^2 + f_y^2}} \mathbf{n}(v). \tag{53}$$

Therefore, from (51) and (53), we can conclude that the direction of the curvilinear coordinate $r_c(x,y)$ in GTCF, $\mathbf{i}_{r_c}$, is the same as that of the contour's normal vector $\mathbf{n}(v) = [-y_v'(v), x_v'(v)]^T$ if the condition in Theorem 2 is satisfied. ■

Based on the discussions above, we need to calculate the parameter v based on the current position x or y. It is evident that the parameter v can be determined uniquely by x if $x_v(v)$ is a monotonic function in a certain interval of the parameter v, e.g., $[v_i, v_{i+1}]$, and the same fact goes for the *Y*-axis. However, note that $x_v(v)$ or $y_v(v)$ may not be a monotonic function all the time. Therefore, the procedure of calculating the parameter v should be switched between the *X*-axis and the *Y*-axis. The method of finding the switching points will be demonstrated.

In the practical implementation, given a specified trajectory $\mathbf{q}_v = [x_v(v), y_v(v)]^T$, $v \in [0,1]$, we first calculate the extremum points of $x_v(v)$ and $y_v(v)$. Denote

$$\mathbf{v}^x = [v_0^x, v_1^x, \ldots, v_i^x, \ldots, v_m^x] \tag{54}$$

and

$$\mathbf{v}^y = [v_0^y, v_1^y, \ldots, v_i^y, \ldots, v_n^y] \tag{55}$$

as the critical node sequences, and extremum values can be achieved at the nodes in the sequences $\mathbf{v}^x$ and $\mathbf{v}^y$ for $x_v(v)$ and $y_v(v)$, respectively. Furthermore, it is assumed that

$$v_0^x < v_1^x < \cdots < v_i^x < \cdots < v_m^x \tag{56}$$

and

$$v_0^y < v_1^y < \cdots < v_i^y < \cdots < v_n^y \tag{57}$$

for the sequences $\mathbf{v}^x$ and $\mathbf{v}^y$, respectively. It is obvious that $x_v(v)$ is a monotonic function between two adjacent critical nodes in the sequence $\mathbf{v}^x$, e.g., $[v_i^x, v_{i+1}^x]$, and the similar fact goes for $y_v(v)$. The switching points can be determined as follows.

Without loss of generality, we have the following assumptions.

1) $v_0^x \leqslant 0 < v_1^x$.
2) $v_0^y \leqslant 0 < v_1^y$.
3) $v_1^y < v_1^x$.

Therefore, the first switching point can be selected as v_1^s s.t.

$$v_1^y < v_1^s < v_1^x. \tag{58}$$

It is obvious that $x_v(v)$ is a monotonic function in the interval $[0, v_1^s]$, and at the switching point v_1^s, the procedure of calculating the parameter v is switched from the *X*-axis to the *Y*-axis. Next, it is assumed that

$$v_i^x < v_2^y < v_{i+1}^x \tag{59}$$

hence, the next switching point can be obtained as v_2^s s.t.

$$v_i^x < v_2^s < v_2^y. \tag{60}$$

The conclusion that $y_v(v)$ is a monotonic function in the interval $[v_1^s, v_2^s]$ is also evident, and the procedure of calculating the parameter v will be switched from the *Y*-axis to the *X*-axis again at the switching point v_2^s. Based on the method above, all the switching points of the geometric path can be obtained, and we describe these switching points in a sequence as follows:

$$\mathbf{v}^s = [0, v_1^s, v_2^s, \ldots, v_k^s] \tag{61}$$

where $v_k^s \geq 1$.

Algorithm 3: The implementation algorithm of the GTCF.

```
Input: $v_d$, $x$ and $y$;
Determine the two adjacent switching points in (61) s.t.
  $v_j^s \leq v_d \leq v_{j+1}^s$.
if $x_v(v)$ is a monotonic function in the interval
  $[v_j^s, v_{j+1}^s]$ then
    Calculate $v$ based on $v_x(x)$ and $x$;
    if $x_v'(v) > 0$ in the interval $[v_j^s, v_{j+1}^s]$ then
        $g(x,y) = x + y$ s.t. $\text{sign}(g_y) \cdot \text{sign}(x_v'(v)) = 1$
        based on the Theorem 2;
    else
        $g(x,y) = x - y$ s.t. $\text{sign}(g_y) \cdot \text{sign}(x_v'(v)) = 1$
        based on the Theorem 2;
    end if
    Calculate $f(x,y)$, $f_x$ and $f_y$ based on (25),
    (44) and (45);
else
    Calculate $v$ based on $v_y(y)$ and $y$;
    if $y_v'(v) > 0$ in the interval $[v_j^s, v_{j+1}^s]$ then
        $g(x,y) = -x + y$ s.t.
        $\text{sign}(g_x) \cdot \text{sign}(y_v'(v)) = -1$ based on the
        Theorem 2;
    else
        $g(x,y) = x + y$ s.t. $\text{sign}(g_x) \cdot \text{sign}(y_v'(v)) = -1$
        based on the Theorem 2;
    end if
    Calculate $f(x,y)$, $f_x$ and $f_y$ based on (26),
    (47) and (48);
end if
```

Based on the definitions above, the detailed implementation of GTCF is illustrated in Algorithm 3. The inputs of Algorithm 3 are the parameter v_d of the desired trajectory, the current actual positions x and y, while the outputs are $f(x,y)$, f_x and f_y. Furthermore, in the implementation, the α and β in Theorem 2 are selected as $\alpha = \pm 1$ and $\beta \pm 1$.

VI. Experimental Validation

In this section, the Lissajous curve used in Section IV is selected as the case study in the experimental contouring tracking task. It is evident that the selected curve cannot be described with an algebraic equation analytically since the contour has self-intersection points. Therefore, the method developed in Section V will be applied in the contouring tracking experiment of this case.

A. Experimental Setup and Control Schemes

The contouring tracking experiment is conducted on an industrial biaxial gantry by Rockwell Automation. The two axes are powered by Anorad LC-50-200 iron core linear motors and mounted orthogonally with the X-axis on top of the Y-axis. The encoders of the two axes have the same measurement resolution of $0.5\,\mu$m. The voltage limits of the servo drivers for the X-axis and the Y-axis are $10\,$V.

In the experiment, an adaptive robust contouring controller (ARC) is utilized to address the parameter uncertainties and modeling uncertainties existing in the biaxial gantry system. Several similar control strategies based on adaptive and robust control have also been applied in many other mechatronic systems such as linear motor-driven systems [27], [28], magnetically levitated planar motor systems [29], hydraulic servo systems [30], [31], active suspension systems [32], [33] and teleoperation robots [34], which all show very successful results and skillful specific designs. The controller structure is the same as that of [8] except that the cogging forces are compensated and the friction coefficients are estimated online by the adaptive law in [6].

B. Performance Index

The performance index is as follows: 1) $\varepsilon_{\text{rms}} = \sqrt{\frac{1}{T}\int_0^T |\varepsilon|^2}$, the root-mean-squared value of the contouring error; 2) $\varepsilon_M = \max|\varepsilon|$, the maximum absolute value of the contouring error; and 3) the total experimental contouring tracking time t_c are employed to measure the performance of each control scheme.

C. Experimental Results

The Lissajous curve, which is formulated in Section IV and shown in Fig. 7, is selected to verify the the effectiveness of the proposed BFC algorithm in the contour tracking application. In the experiments, the bounds of the X-axis and the Y-axis are set as $v_x = 1\,$m/s, $v_y = 1\,$m/s, $a_x = 30\,$m/s^2 and $a_y = 5\,$m/s^2. The GTCF-based contour tracking algorithm is compared with the LTCF-based contour tracking algorithm shown in [4], and [6]. The following four contour control schemes are compared.

C1). GTCF-based ARC algorithm with $0.6\,$m/s constant feed rate on the desired contour.

C2). The same GTCF-based ARC algorithm with trajectory generated by the proposed BFC algorithm.

C3). LTCF-based ARC algorithm with $0.6\,$m/s constant feed rate on the desired contour.

C4). The same LTCF-based ARC algorithm with trajectory generated by the proposed BFC algorithm.

The actual contouring errors in the experiments are calculated by utilizing an offline numerical method. The contouring errors and the control inputs of C1, C2, C3, and C4 are shown in Figs. 4–6, respectively. Furthermore, the actual contour tracking results of C1, C2, C3, and C4 are shown in the Fig. 7, and the enlarged area of 'A' in Fig. 7 is shown in Fig. 8. Moreover, the detailed performance indexes are listed in Table II.

As seen from the Fig. 4 and the Table II, compared with the constant feed rate cases (C1 and C3), smaller contouring tracking errors and shorter contouring tracking times can be achieved in the trajectory planning cases (C2 and C4). Furthermore, from Figs. 4 and 8, it is observed that large contouring errors occur at the points with high curvature in C1 and C3, which is due to the controller saturation shown in Figs. 5 and 6. On the other hand, with trajectory planning, a low feed rate is utilized in the neighbourhood of the sharp corner points to avoid the controller saturation, and thus higher contouring tracking performances

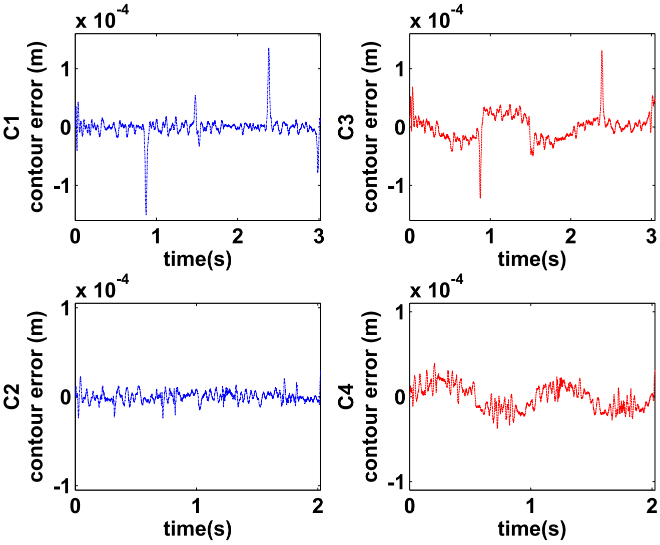


Fig. 4. Contouring tracking errors of C1, C2, C3, and C4.

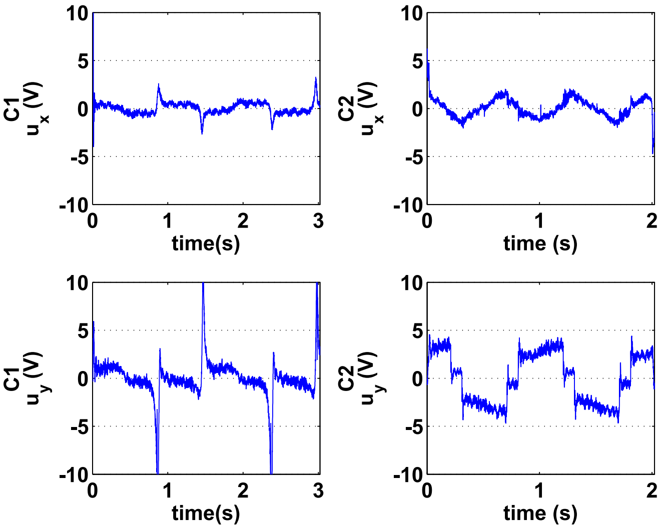


Fig. 5. Control inputs of C1 (left column) and C2 (right column).

TABLE II
EXPERIMENTAL RESULTS OF THE LISSAJOUS CURVE

	$\varepsilon_{\text{rms}}(\mu\text{ m})$	$\varepsilon_M\ (\mu\text{m})$	$t_c\,(s)$
C1	18.3	150.3	3.02
C2	5.9	30.5	2.02
C3	23.0	130.5	3.04
C4	14.3	38.9	2.02

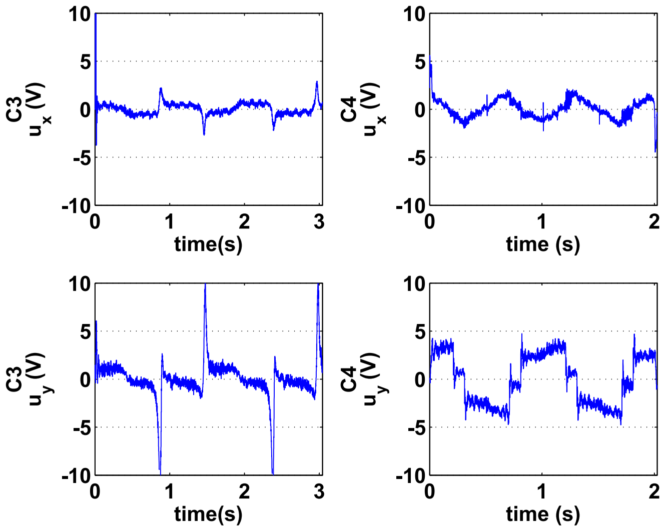


Fig. 6. Control inputs of C3 (left column) and C4 (right column).

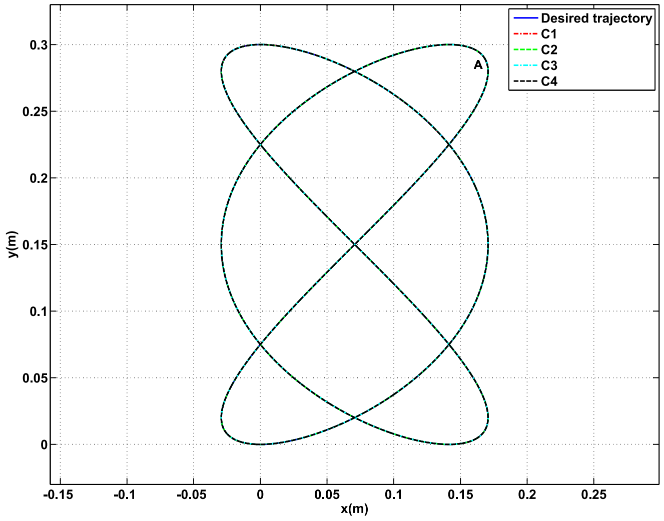


Fig. 7. Contour tracking results of C1, C2, C3, and C4.

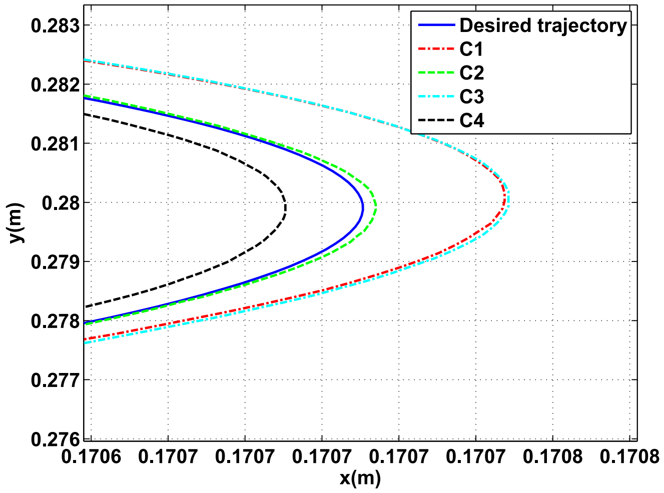


Fig. 8. Enlarged graph of the 'A' area.

are obtained in C2 and C4. Furthermore, from the Fig. 4 and the Table II, it is evident that the GTCF cases (C1 and C2) achieve better contouring tracking performances from the viewpoint of the overall contouring tracking task in comparison with the LTCF cases (C3 and C4).

Based on the discussions above, it is seen that with the proposed trajectory planning scheme, a higher contouring tracking accuracy and a shorter contouring tracking time are achieved simultaneously. Therefore, the proposed trajectory planning approach can make an excellent trade-off between the accuracy and productivity in the contouring tracking task.

VII. Conclusion

In this paper, a novel BFC algorithm is proposed to address the minimum time trajectory planning problem. The procedure of computing the optimal solution of the time optimal trajectory planning problem is presented in detail. The proposed BFC algorithm is very computationally efficient because the optimal solution at each node can be obtained by several analytical equations directly. An extended BFC algorithm is also presented to incorporate some additional velocity-dependent constraints. Computation results confirm the optimality and the high computational efficiency of the BFC algorithm. Moreover, the computation results of planning a Lissajous curve validate the effectiveness of the extended BFC algorithm. Several implementation issues of the GTCF [8] based contouring controls on tracking complicated contours are also discussed. The experimental results confirm that the proposed trajectory planning approach indeed enables the simultaneous achievement of excellent contouring tracking accuracy and high productivity in practice.

References

[1] Y. Koren, "Cross-coupled biaxial computer control for manufacturing systems," *J. Dyn. Syst., Meas. Control*, vol. 102, no. 4, pp. 265–272, 1980.

[2] S.-S. Yeh and P.-L. Hsu, "Estimation of the contouring error vector for the cross-coupled control design," *IEEE/ASME Trans. Mechatronics*, vol. 7, no. 1, pp. 44–51, Mar. 2002.

[3] J. Yang and Z. Li, "A novel contour error estimation for position loop-based cross-coupled control," *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 4, pp. 643–655, Aug. 2011.

[4] G.-C. Chiu and M. Tomizuka, "Contouring control of machine tool feed drive systems: a task coordinate frame approach," *IEEE Trans. Control Syst. Technol.*, vol. 9, no. 1, pp. 130–139, Jan. 2001.

[5] M.-Y. Cheng and C.-C. Lee, "Motion controller design for contour-following tasks based on real-time contour error estimation," *IEEE Trans. Ind. Electron.*, vol. 54, no. 3, pp. 1686–1695, Jun. 2007.

[6] C. Hu, B. Yao, and Q. Wang, "Coordinated adaptive robust contouring control of an industrial biaxial precision gantry with cogging force compensations," *IEEE Trans. Ind. Electron.*, vol. 57, no. 5, pp. 1746–1754, May 2010.

[7] Y. Lou, H. Meng, J. Yang, Z. Li, J. Gao, and X. Chen, "Task polar coordinate frame-based contouring control of biaxial systems," *IEEE Trans. Ind. Electron.*, vol. 61, no. 7, pp. 3490–3501, Jul. 2014.

[8] B. Yao, C. Hu, and Q. Wang, "An orthogonal global task coordinate frame for contouring control of biaxial systems," *IEEE/ASME Trans. Mechatronics*, vol. 17, no. 4, pp. 622–634, Aug. 2012.

[9] C. Hu, Z. Hu, Y. Zhu, and Z. Wang, "Advanced GTCF based LARC contouring motion control on an industrial X-Y linear-motor-driven stage with experimental investigation," *IEEE Trans. Ind. Electron.*, vol. PP, no. 99, pp. 1–1, 2016, doi= 10.1109/TIE.2016.2574298.

[10] J. Dong and J. Stori, "A generalized time-optimal bidirectional scan algorithm for constrained feed-rate optimization," *J. Dyn. Syst. Meas. Control*, vol. 128, no. 2, pp. 379–390, 2006.

[11] M. Yuan, B. Yao, D. Gao, X. Zhu, and Q. Wang, "A novel algorithm for time optimal trajectory planning," in *Proc. ASME Dyn. Syst. Control Conf*, 2014, p. V001T02A003.

[12] J. E. Bobrow, S. Dubowsky, and J. Gibson, "Time-optimal control of robotic manipulators along specified paths," *Int. J. Robot. Res.*, vol. 4, no. 3, pp. 3–17, 1985.

[13] S. Liu, D. Sun, and C. Zhu, "Coordinated motion planning for multiple mobile robots along designed paths with formation requirement," *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 6, pp. 1021–1031, Dec. 2011.

[14] S. Liu and D. Sun, "Minimizing energy consumption of wheeled mobile robots via optimal motion planning," *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 2, pp. 401–411, Apr. 2014.

[15] F. Pfeiffer and R. Johanni, "A concept for manipulator trajectory planning," *IEEE J. Robot. Autom.*, vol. 3, no. 2, pp. 115–123, Apr. 1987.

[16] K. G. Shin and N. D. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Trans. Autom. Control*, vol. 30, no. 6, pp. 531–541, Jun. 1985.

[17] J.-J. Slotine and H. S. Yang, "Improving the efficiency of time-optimal path-following algorithms," *IEEE Trans. Robot. Autom.*, vol. 5, no. 1, pp. 118–124, Feb. 1989.

[18] K. G. Shin and N. D. McKay, "A dynamic programming approach to trajectory planning of robotic manipulators," *IEEE Trans. Autom. Control*, vol. 31, no. 6, pp. 491–500, Jun. 1986.

[19] D. Costantinescu and E. Croft, "Smooth and time-optimal trajectory planning for industrial manipulators along specified paths," *J. Robot. Syst.*, vol. 17, no. 5, pp. 233–249, 2000.

[20] D. Constantinescu, "Smooth time optimal trajectory planning for industrial manipulators," M.S. thesis, *Dept. Mech. Eng.,* Univ. British Columbia, Vancouver, BC, Canada, 1998.

[21] Q. Zhang, S. Li, and J. Guo, "Smooth time-optimal tool trajectory generation for CNC manufacturing systems," *J. Manuf. Syst.*, vol. 31, no. 3, pp. 280–287, 2012.

[22] D. Renton and M. Elbestawi, "High speed servo control of multi-axis machine tools," *Int. J. Mach. Tools Manuf.*, vol. 40, no. 4, pp. 539–559, 2000.

[23] J. Dong and J. Stori, "Optimal feed-rate scheduling for high-speed contouring," *J. Manuf. Sci. Eng.*, vol. 129, no. 1, pp. 63–76, 2007.

[24] L. Lu, B. Yao, and W. Lin, "A two-loop contour tracking control for biaxial servo systems with constraints and uncertainties," in *Proc. Amer. Control Conf.*, 2013, pp. 6468–6473.

[25] M. Yuan, B. Yao, and X. Zhu, "An efficient computation algorithm for time optimal trajectory planning with physical constraints," in *Proc. IEEE Conf. Inform. Autom.*, 2015, pp. 3100–3105.

[26] J. Dong, P. Ferreira, and J. Stori, "Feed-rate optimization with jerk constraints for generating minimum-time trajectories," *Int. J. Mach. Tools Manuf.*, vol. 47, no. 12, pp. 1941–1955, 2007.

[27] Z. Chen, B. Yao, and Q. Wang, "Mu-synthesis-based adaptive robust control of linear motor driven stages with high-frequency dynamics: A case study," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 3, pp. 1482–1490, Jun. 2015.

[28] Z. Chen, B. Yao, and Q. Wang, "Accurate motion control of linear motors with adaptive robust compensation of nonlinear electromagnetic field effect," *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 3, pp. 1122–1129, Jun. 2013.

[29] C. Hu, Z. Wang, Y. Zhu, M. Zhang, and H. Liu, "Performance oriented precision LARC tracking motion control of a magnetically levitated planar motor with comparative experiments," *IEEE Trans. Ind. Electron.*, vol. PP, no. 99, pp. 1–1, 2016, doi= 10.1109/TIE.2016.2538743.

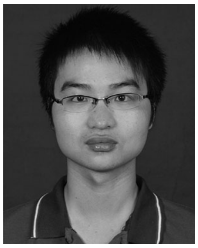
[30] J. Yao, Z. Jiao, D. Ma, and L. Yan, "High-accuracy tracking control of hydraulic rotary actuators with modeling uncertainties," *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 2, pp. 633–641, Apr. 2014.

[31] J. Yao, Z. Jiao, and D. Ma, "Extended-state-observer-based output feedback nonlinear robust control of hydraulic systems with backstepping," *IEEE Trans. Ind. Electron.*, vol. 61, no. 11, pp. 6285–6293, Nov. 2014.

[32] W. Sun, H. Gao, and O. Kaynak, "Finite frequency control for vehicle active suspension systems," *IEEE Trans. Control Syst. Technol.*, vol. 19, no. 2, pp. 416–422, Mar. 2011.

[33] W. Sun, H. Gao, and O. Kaynak, "Vibration isolation for active suspensions with performance constraints and actuator saturation," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 2, pp. 675–683, Apr. 2015.

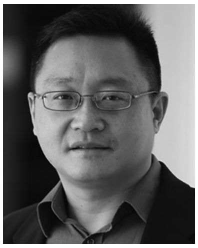
[34] Z. Chen, Y.-J. Pan, and J. Gu, "Integrated adaptive robust control for multilateral teleoperation systems under arbitrary time delays," *Int. J. Robust Nonlinear Control*, vol. 26, no. 12, pp. 2708–2728, 2016.



Mingxing Yuan received the B.Eng. degree in aircraft manufacturing engineering from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2013. He is currently working toward the Ph.D degree at the College of Mechanical Engineering, Zhejiang University, Hangzhou, China.

Zheng Chen received the B.Eng. and Ph.D. degrees in mechatronic control engineering from Zhejiang University, Hangzhou, China, in 2007 and 2012, respectively.

From 2013 to 2015, he was a Post-Doctoral Researcher in the Department of Mechanical Engineering, Dalhousie University, Halifax, NS, Canada. Since 2015, he has been an Associated Professor with Ocean College, Zhejiang University.



Bin Yao received the B.Eng. degree in applied mechanics from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 1987; the M.Eng. degree in electrical engineering from Nanyang Technological University, Singapore, in 1992; and the Ph.D. degree in mechanical engineering from the University of California, Berkeley, CA, USA, in 1996.

Since 1996, he has been with the School of Mechanical Engineering, Purdue University, West Lafayette, IN, USA, and was promoted to the rank of Professor in 2007.

Dr. Yao was honored as a Kuang-piu Professor in 2005 and a Changjiang Chair Professor at Zhejiang University by the Ministry of Education of China in 2010.

Xiaocong Zhu received the B.Eng. and Ph.D degrees in mechanical engineering from Zhejiang University, Hangzhou, China, in 2002 and 2007, respectively.

She is currently an Associated Professor with the College of Mechanical Engineering, Zhejiang University. Her research interests include pneumatic servo control, nonlinear control theory and applications, mechatronic control, and so on.