



# A Boundary Auto-Location Algorithm for the Prediction of Milling Stability Lobe Diagram

Mingkai Zhang<sup>1</sup>, Xiaowei Tang<sup>1</sup>, Rong Yan<sup>1(✉)</sup>, Fangyu Peng<sup>2</sup>, Chen Chen<sup>1</sup>,  
Yuting Li<sup>1</sup>, and Haohao Zeng<sup>1</sup>

<sup>1</sup> School of Mechanical Science and Engineering,  
Huazhong University of Science and Technology, Wuhan 430074, China  
yanrong@hust.edu.cn

<sup>2</sup> State Key Lab of Digital Manufacturing Equipment and Technology,  
Huazhong University of Science and Technology, Wuhan 430074, China

**Abstract.** Chatter is known as a main factor that limits the machining quality and efficiency, and one universal solution is to predict occurrences of chatter via calculating the stability lobe diagram (SLD), such as time-domain methods, which are relatively time-consuming. Thus, based on time-domain methods, a boundary auto-location algorithm for the prediction of SLD in milling is proposed. In the proposed method, by setting a series of judgements based on the state of the transition matrix of the dynamic system, the calculation trajectory automatically surrounds the stability boundary line except isolated islands. Only the points on and around the stability boundary were calculated to draw SLD. The contrast simulations were conducted to verify the calculation efficiency of the given algorithm. And the results show that the computational time of the proposed method was cut down significantly than that of the traditional method.

**Keywords:** Milling process · Stability prediction · Boundary auto-location

## 1 Introduction

Prediction of stability lobe diagram is a vital method to avoid chatter that limits the quality and efficiency of manufacturing. However, the calculation of the stability prediction relate to plenty of parameters, such as modal parameters, cutting parameters and so on. It is time-consuming to calculate all of the points' stability of the parameters space, like the robotic milling. Therefore, a boundary auto-location algorithm is proposed to solve this issue in the prediction of SLD.

With respect to the prediction of cutting stability, so far there are two major methods: one is the frequency-domain method [1, 2], which is characterized by high-computational-efficiency and relatively low prediction accuracy [3]; another is time-domain method [4, 5], which is time-consuming and has higher prediction accuracy than the former. For the cutting system, the stability can be predicted by solving its feature equation in the former method and the time-periodic delayed-differential equation in the later method. Many efforts have been devoted to improving the prediction accuracy and efficiency in the previous literatures.

Altintas and Budak [1] proposed zero-order method that used the average of the Fourier series of the dynamic milling force to approximate the milling force variation, but is not capable in low radial immersion milling. Merdol and Altintas [2] solved this issue via developing multi-frequency method considering the harmonics of the tooth-passing frequencies. Insuperger et al. [4] firstly proposed semi-discretization method (SDM), in which the time-delayed term is discretized. And it is widely adopted to predict the stability of linear-variant time-delayed milling system in time domain. Then, some novel and improved algorithms were proposed to improve the computational efficiency, like full-discretization method (FDM) presented by Ding et al. [5], numerical integration method (NIM) [7], linear approximation of acceleration method (LAAM) [8], Runge-Kutta-based complete discretization method (RKCDM) presented by Li et al. [9], and so on [10, 11]. Also, Li et al. [12] proposed a fast-straight forward method to plot SLD using modal parameters of milling system, but with a relatively low accuracy. In this paper, based on the time-domain methods, a fast algorithm of SLD prediction was developed without consideration of the special SLD contour that called stability isolated islands mentioned in Ref. [13].

All of the time-domain methods focus on the stability calculation of certain cutting parameters, like revolution speed and cut depth which are generally used to represent two-dimension SLD. Therefore, in the traditional method, SLD is given by calculating stabilities of all the points in the discretized cutting parameters space. It is a huge work and time-consuming. In this paper, by setting a series of judgements based on the state of the transition matrix of the dynamic system, only the points on and around the stability boundary were automatically calculated to draw SLD.

The remaining part of this paper is organized as follows: Sect. 2 proposes the boundary auto-location algorithm for prediction of the SLD; Sect. 3 gives the comparisons of computational efficiency between the proposed method and traditional methods. And in Sect. 3 this paper is concluded.

## 2 The Proposed Algorithm

### 2.1 The Stability Prediction Model in Milling Process

The milling system can be simplified as a two-DOF dynamic system, as shown in Fig. 1, the dynamic model is a time-periodic delayed-differential equation and can be expressed in the form of Eq. (1), as follow:

$$\mathbf{M} \begin{bmatrix} \ddot{x}(t) \\ \ddot{y}(t) \end{bmatrix} + \mathbf{C} \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \end{bmatrix} + \mathbf{K} \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \mathbf{K}_c(t) \begin{bmatrix} x(t) - x(t - T) \\ y(t) - y(t - T) \end{bmatrix} \quad (1)$$

where  $M$ ,  $C$ ,  $K$ ,  $K_c(t)$  is the mass, damping, stiffness matrices and the cutting force coefficient matrix of dynamic system respectively.  $x(t) - x(t - T)$ ,  $y(t) - y(t - T)$  represent the dynamic displacement of the cutter between the previous and current revolution on the  $X$ ,  $Y$  direction.  $T$  is the tooth-passing period.

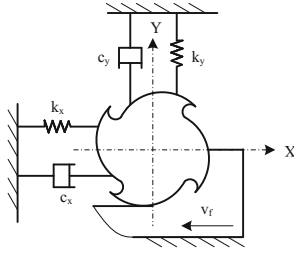


Fig. 1. The two-DOF dynamic model of milling system

By using one of the time-domain methods in the references, the Eq. (1) can be solved and the transition matrix  $\Phi$  of the dynamic system can be obtained. And then according to the Floquet theory, the system is judged as stable when the maximum absolute value of the eigenvalues of the transition matrix  $\Phi$  is less than 1. The following deduction process is developed based on this criterion.

### 2.2 The Algorithm for Locating the Original Point on the Stability Boundary

Firstly, it is necessary to evenly discretize the cutting parameters space of SLD. Here, the revolution speed and cut depth are used to configure the two-dimension SLD. The revolution speed and cut depth interval are divided into  $m$  and  $n$  discrete values as follows:

$$\begin{cases} S_i = \frac{i}{m}(S_{\max} - S_{\min}), i = 1, 2, \dots, m \\ d_j = \frac{j}{n}(d_{\max} - d_{\min}), j = 1, 2, \dots, n \end{cases} \quad (2)$$

where  $S, d$  respectively represent the revolution speed and cut depth.

Then, along the revolution speed incremental direction  $i = 1 \rightarrow m$ , the limitation value of cut depth  $d_{chatter}$  related to each discrete revolution speed  $S_i$  can be sequentially obtained via the algorithm in this paper.

---

**Algorithm I.** Locate the original point on the stability boundary

---

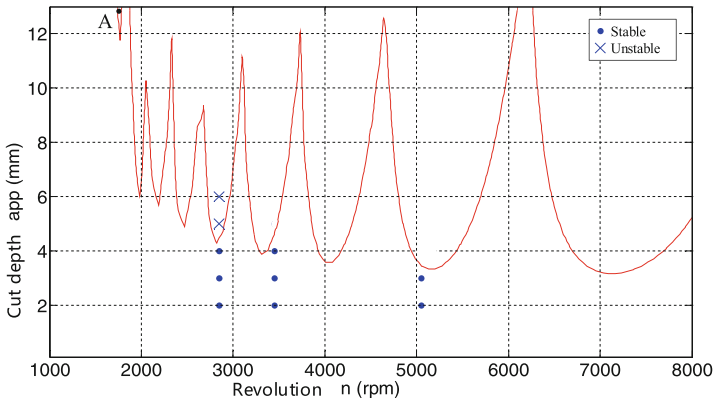
```

1: set the initial condition:  $i = 1, [a, b] = [1, n], j = [n/2]$ 
2: while  $1 \leq j \leq n$  do
3:   if  $E_\phi(S_i, d_j) = 1$  then
4:     return  $d_{chatter} = d_j$ 
5:   end algorithm
6:   else
7:     if  $j = a$  then
8:       if  $(E_\phi(S_i, d_a) - 1) \cdot (E_\phi(S_i, d_b) - 1) < 0$  then
9:         return  $d_{chatter} = d_a + (1 - E_\phi(S_i, d_a)) \cdot (d_b - d_a) / (E_\phi(S_i, d_b) - E_\phi(S_i, d_a))$ 
10:        end algorithm
11:       else do the next iteration under  $S_{i+1}$ 
12:       end if
13:     else
14:       if  $E_\phi(S_i, d_j) > 1$  then  $b = j$ 
15:       else  $a = j$ 
16:       end if
17:     end if
18:      $j = [(a + b) / 2]$ 
19:   end if
20: end while

```

---

For the proposed boundary auto-location algorithm, there is an initial condition: the original point on the stability boundary should be determined, like the point A shown in the Fig. 2. To locate this original point, the Algorithm I is also developed based on the bisection method.



**Fig. 2.** The original point A on the stability boundary

As shown in the Fig. 2, SLD is divided into stable and unstable zone by the stability boundary, and this discrimination method could be formulated as the follow:

$$\begin{cases} E_\phi(S_i, d_j) \geq 1, & \text{system is unstable} \\ E_\phi(S_i, d_j) < 1, & \text{system is stable} \end{cases} \quad (3)$$

where  $E_\phi(S_i, d_j)$  refers to the maximum absolute value of the eigenvalues of the transition matrix  $\Phi$  under the cutting parameters of  $(S_i, d_j)$ .

Note that  $E_\phi(S_i, d_j)$  of all the points on the stability boundary are equal to 1. Therefore, the boundary point  $(S_i, d_{chatter})$  of current discrete revolution speed  $S_i$  can be located rapidly based on the bisection method.

This bisection method iterates from  $i = 1$  to  $i = m$  until finding the first boundary point, which one would be identified as the original point A for the next algorithm. Each iteration start with an initial subscript interval of cut depth  $[a, b] = [1, n]$ , and then  $a$  and  $b$  would be updated corresponding to the value of  $E_\phi(S_i, d_j)$  as follow:

$$\begin{cases} b = j, & E_\phi(S_i, d_j) > 1 \\ a = j, & E_\phi(S_i, d_j) < 1 \end{cases} \quad (4)$$

Where  $j = \lfloor (a + b)/2 \rfloor$ , and the bracket  $\lfloor \ ]$  is a sign operator which means rounding down the number.

There exist the boundary point among interval  $[a, b]$  when  $(E_\phi(S_i, d_a) - 1) \cdot (E_\phi(S_i, d_b) - 1) \leq 0$ . And then the cutting parameter  $(S_i, d_{chatter})$  of the boundary point can be approximated by some interpolation methods, such as Lagrange Approximation. Here, the boundary point is given by Eq. (5) as follow:

$$\begin{aligned} d_{chatter} &= d_a + (1 - E_\phi(S_i, d_a)) \cdot (d_b - d_a) / (E_\phi(S_i, d_b) - E_\phi(S_i, d_a)), \\ \text{if } (E_\phi(S_i, d_a) - 1) \cdot (E_\phi(S_i, d_b) - 1) &\leq 0 \end{aligned} \quad (5)$$

The detailed algorithm can be summarized as Algorithm I. Meanwhile, it is worthy to be mentioned, in general, if there exist one boundary point among subscript interval  $[a, b]$ , the relationship that  $b = a + 1$  is satisfied. This attribute would be useful for the boundary auto-location algorithm.

### 2.3 The Boundary Auto-Location Algorithm

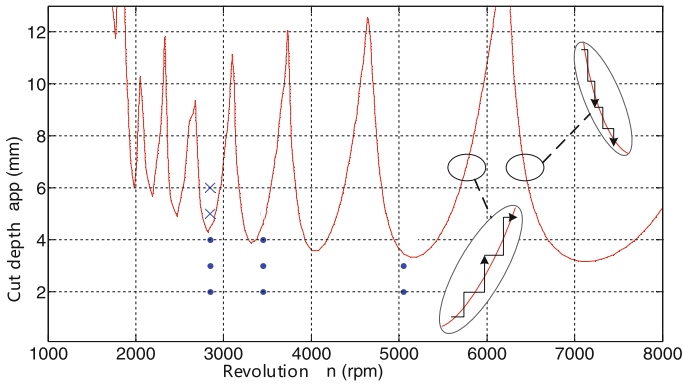
As can be seen in Fig. 2, generally, the stability boundary is a contour line, which is known for its continuity and equivalence. Therefore, once one boundary point has been founded, like the aforementioned point A  $(S_i, d_{chatter})$ , the next boundary point  $(S_{i+1}, d_{chatter})$  can be determined in the neighborhood of it. Based on this it is unnecessary to calculate all the points among  $[d_1, d_n]$  to determine the next  $d_{chatter}$ , only the points on and around the latest stability boundary point are needed.

Assume that, under the current discrete revolution speed  $S_i$ , the  $d_{chatter}$  locate in the subscript interval  $[a, b]$  with  $b = a + 1$ , such as point A. And the next iteration should

start from  $j = b$  or  $j = a$  under the next discrete revolution speed  $S_{i+1}$ , in other words, directly calculating the value of the  $E_\phi(S_{i+1}, d_b)$  or  $E_\phi(S_{i+1}, d_a)$ . Then same as the Algorithm I above, the Eq. (5) also can be used to determine the stability boundary point. But different from the update forms of the parameters  $a, b$  and  $j$  that used in Eq. (4), here, those parameters are updated in the form of Eq. (6).

$$\begin{cases} b = j, & E_\phi(S_i, d_j) > 1 \\ a = j, & E_\phi(S_i, d_j) < 1 \end{cases} \quad (6)$$

In this method, the calculation process would be proceeding along the incremental direction of cut depth when the current point  $(S_i, d_j)$  is judged as stable and along the decremental direction in the unstable zone. Only a few steps are performed for this process to locate the new stability point, because the attributes of continuity and equivalence of the contour line. The whole trajectory of the calculation process closely surround the stability boundary, just like automatically keep following the tangential direction of the boundary line as can be seen in Fig. 3. The method of this part can be summarized as Algorithm II. Note that, in a special case where the system is stable in whole given cut depth interval  $(d_{min}, d_{max})$  under current revolution speed, the algorithm would proceed along  $d_{max}$  until entering into the unstable zone.



**Fig. 3.** Diagram of calculation process of the boundary auto-location algorithm

---

**Algorithm II.** Boundary auto-location algorithm for the SLD prediction

---

- 1: input the parameters of the latest boundary point:  $S_{i-1}, d_j, [a, b]$
  - 2: while  $1 \leq j \leq n$  do
  - 3: if  $(E_\phi(S_i, d_a) - 1) \cdot (E_\phi(S_i, d_b) - 1) \leq 0$  then
  - 4: return  $d_{chatter} = d_a + (1 - E_\phi(S_i, d_a)) \cdot (d_b - d_a) / (E_\phi(S_i, d_b) - E_\phi(S_i, d_a))$
  - 5: do the next iteration under  $S_{i+1}$
  - 6: else
  - 7: if  $E_\phi(S_i, d_j) > 1$  then  $b = a, a = a - 1$
  - 8: else  $a = b, b = b + 1$
  - 9: end if
  - 10:  $j = b$
  - 11: end if
  - 12: end while
  - 13: do the next iteration under  $S_{i+1}$
- 

### 3 Simulations

The experimental platform is Intel(R) Core(TM) i5-4460, CPU@3.20 GHz, RAM 8 GB, and all of the algorithms are written and executed in MATLAB R2012a. The transition matrix  $\Phi$  of the dynamic system is calculated via using the 2nd-FDM, NIM, and LAAM in the literatures [6], [7], and [8], respectively. The computational efficiency of the proposed method is studied and compared with that of the traditional method. Besides, the SDM in literature [4] is used as a criterion-reference to validate the accuracy of the calculation results.

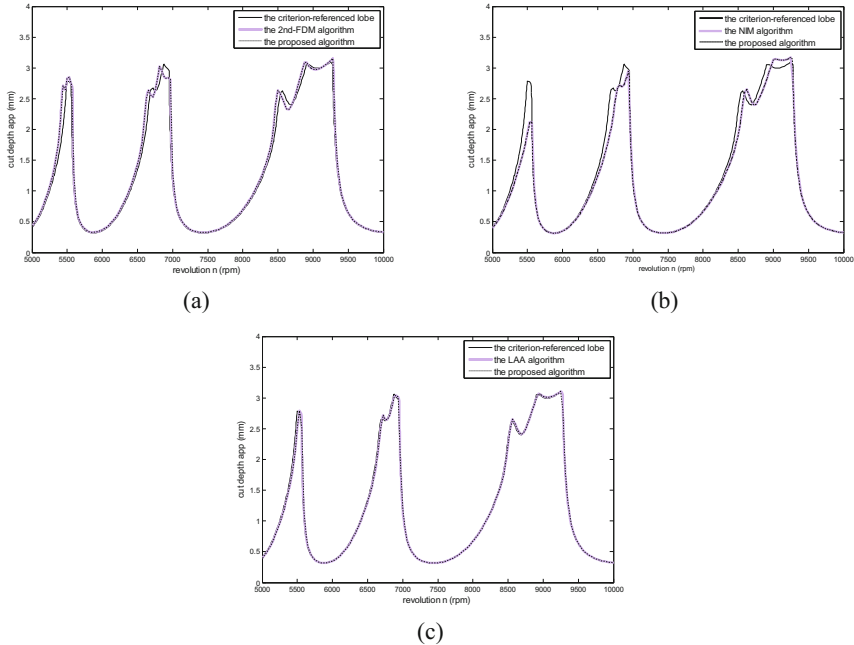
Here, SLD of single-DOF milling model is represented by parameters space with cut depth and revolution speed under  $0 \text{ mm} \leq d \leq 4 \text{ mm}$  and  $5000 \text{ rpm} \leq S \leq 10000 \text{ rpm}$ . The quantity of discretization are  $m = 160$  and  $n = 80$ . Besides, the tooth-passing period  $T$ , which is also an important parameter and proportional to the prediction accuracy in time-domain method to solve Eq. (1), is divided into 40 time intervals in the first three methods. And the tooth-passing period  $T$  in the last method is divided into 200 time intervals as a criterion-reference.

The other cutting parameters, including tool, cutting force coefficients, etc., and dynamic parameters, including modal mass, natural frequency, etc., are same as the benchmark data adopted in the references [4, 6–8], as can be seen in Table 1.

**Table 1.** The cutting and dynamic parameters for single-DOF milling model.

Radial immersion ratio	Tangential cutting force coefficient	Normal cutting force coefficient	Natural frequency	Relative damping	Modal mass
1	$6 \times 10^8 \text{ N/m}^2$	$2 \times 10^8 \text{ N/m}^2$	922 Hz	0.011	0.03993 kg

The calculation results and comparisons of SLD for single-DOF milling model based on different time-domain methods are shown as Fig. 4. Because the transition matrix  $\Phi$  of every point of SLD is same in each case, there is minimal difference of the stability boundary between the proposed fast algorithm and the traditional algorithm. And the tiny difference here is caused by the calculation method of contour line in MATLAB, which is used in later. The boundary point of the former is determined by Eq. (5), while the interpolation method, like Eq. (5), is also used in revolution speed interval to determine the boundary point of the later.



**Fig. 4.** SLD of the fast algorithm based on: (a) 2nd-FDM (b) NIM (c) LAAM

Meanwhile, in the terms of prediction accuracy the LAAM has the highest agreement with the criterion-referenced lobe in Fig. 4(c). The 2nd FDM in Fig. 4(a) is inferior to the former, and the NIM in Fig. 4(b) is lowest.

The comparisons of the computational time of the fast algorithm based on different method are shown in Table 2. There are three results of each case are given on account of the fluctuations of computer performance. And the computational efficiency is represented by the average of those three times. Compared with the traditional method that calculates all the points of SLD, the proposed algorithm enhance the computational time by 18.7 times in the 2nd-FDM-based case, 17.4 times in the NIM-based case and 19.5 times in the NIM-based case.



**Table 2.** The comparisons of the computational time.

Algorithm		2nd-FDM-based		NIM-based		LAA-based	
		Convention	This paper	Convention	This paper	Convention	This paper
Time(s)	1	28.14648	1.49503	13.28893	0.75696	6.98059	0.35498
	2	28.19411	1.50136	13.29482	0.76778	6.97950	0.35864
	3	28.26346	1.51737	13.31026	0.77051	7.00041	0.35882
Average		28.20135	1.50459	13.29802	0.76508	6.98833	0.35748

Therefore, the proposed boundary auto-location algorithm significantly improved the computational efficiency of the SLD prediction with same accuracy as the time-domain methods.

## 4 Conclusions

This paper proposes a boundary auto-location algorithm for the prediction of milling stability lobe diagram without consideration of stability isolated islands. In the proposed method, the calculation trajectory automatically follows the tangential direction of the stability boundary line via setting the judgements based on the state of the transition matrix of the dynamic system and Floquet theory. Only the points on and around the stability boundary were calculated, thereby avoiding to calculate all of the points on SLD. The high-computational-efficiency of the proposed algorithm is validated by the simulations, in which the computational time is enhanced significantly than that of the traditional method.

**Acknowledgements.** This research is supported by National Natural Science Foundation of China under Grant no. 51625502, Innovation Group of Hubei under Grant no. 2017CFA003, Natural Science Foundation of Jiangsu under Grant no. BK20161473.

## References

1. Altintas, Y., Budak, E.: Analytical prediction of stability lobes in milling. *CIRP* **44**(1), 357–362 (1995)
2. Merdol, S., Altintas, Y.: Multi frequency solution of chatter stability for low immersion milling. *ASME J. Manuf. Sci. Eng.* **126**(3), 459–466 (2004)
3. Bayly, P., Mann, B., Schmitz, T., et al.: Effects of radial immersion and cutting direction on chatter instability in end-milling. In: 2002 ASME International Mechanical Engineering Congress and Exposition, pp. 351–363. American Society of Mechanical Engineers, New Orleans (2002)
4. Insuperger, T., Stépán, G.: Updated semi-discretization method for periodic delay-differential equations with discrete delay. *Int. J. Numer. Methods Eng.* **61**(1), 117–141 (2004)
5. Ding, Y., Zhu, L.M., Zhang, X.J., et al.: A full-discretization method for prediction of milling stability. *Int. J. Mach. Tools Manuf.* **50**(5), 502–509 (2010)
6. Ding, Y., Zhu, L.M., Zhang, X.J., et al.: Second-order full-discretization method for milling stability prediction. *Int. J. Mach. Tools Manuf.* **50**, 926–932 (2010)

7. Ding, Y., Zhu, L.M., Zhang, X.J., et al.: Numerical integration method for prediction of milling stability. *ASME J. Manuf. Sci. Eng.* **133**(3), 031005 (2011)
8. Huang, T., Zhang, X.M., Zhang, X.J., Ding, H.: An efficient linear approximation of acceleration method for milling stability prediction. *Int. J. Mach. Tools Manuf.* **74**, 56–64 (2013)
9. Li, Z., Yang, Z., Peng, Y., et al.: Prediction of chatter stability for milling process using Runge-Kutta-based complete discretization method. *Int. J. Adv. Manuf. Technol.* **1**, 1–10 (2015)
10. Zhang, Z., Li, H., Meng, G., et al.: A novel approach for the prediction of the milling stability based on the Simpson method. *Int. J. Mach. Tools Manuf.* **99**, 43–47 (2015)
11. Tang, X., Peng, F., Rong, Y., et al.: Accurate and efficient prediction of milling stability with updated full-discretization method. *Int. J. Adv. Manuf. Technol.* **88**(9–12), 2357–2368 (2017)
12. Li, Z., Wang, Z., Shi, X.: Fast prediction of chatter stability lobe diagram for milling process using frequency response function or modal parameters. *Int. J. Adv. Manuf. Technol.* **89**(9–12), 2603–2612 (2017)
13. Patel, B.R., Mann, B.P., Young, K.A.: Uncharted islands of chatter instability in milling. *Int. J. Mach. Tools Manuf.* **48**(1), 124–134 (2008)