An Integrated Approach to the Improvement of Stability Lobes

Sekar Mohan[1], Srinivas Jonnalagadda[1], Namcheol Kang[1] and Seung-Han Yang[1,#]
1 School of Mechanical Engineering, Kyungpook National University, 1370 Sankyuk-dong, Buk-gu, Deagu, South Korea, 702-701
Corresponding Author / E-mail: syang@knu.ac.kr, TEL: +82-53-950-6569, FAX: +82-53-950-6550

KEYWORDS : Stability lobe diagram, Regenerative chatter, Spurious loops, Trimming

Redundant spurious loops and overlapping segments in stability lobe diagrams cannot be removed easily by regular methods. A programmatic approach is presented to detect multiple spurious loops occurring within the lobes. They are identified by reorganizing the numerical lobe data into speed-depth corners for subsequent removal. The same principle is then extended to remove overlapping segments of adjacent lobes to obtain a final continuous stability map. Unlike existing methods, the proposed methodology requires no interface with additional editing software, and can also yield stability lobe diagrams more quickly. The methodology is presented with lobe diagrams constructed using milling and turning models.

Manuscript received: June 30, 2008 / Accepted: August 25, 2008

1. Introduction

The stability lobe diagram is a generally accepted tool for selecting a chatter-free combination of speed and depth of cut.[1] A stability lobe diagram consists of several lobes that are computed for an integer number of waves. These lobes are discontinuous because they overlap with their neighboring lobes and often have spurious loops within them (Fig. 1). The spurious loops and overlapping lobe regions must be removed to obtain a continuous map of spindle speed and allowable depth of cut.[2] This method distinctly identifies stability pockets of higher speed and depth of cut and can be used to increase the metal removal rate.

Attempts have been made to develop continuous lobe diagrams from both analytical equations and experimental results. The general approach to identify and trim loops that occur within stability lobes is to use a software tool with trimming capabilities. Schmitz[2] proposed a three-step methodology to identify loops with MATLAB and trim them with AUTOCAD. The problems with this approach are that

i) the approach does not consider the trimming of multiple loops.
ii) the spurious loops are recognized by identifying points manually, and the indices found are extended to other sections, which makes the approach suitable only for single loops with closely spaced modes.
iii) the approach requires editing with AutoCAD and thus imposes additional tasks such as
    a) scaling lobe data to visualize lobes properly and to accomplish the trimming.
    b) converting data to a suitable format for CAD data exchange, a process that results in the truncation of data.
    c) manual positioning of lobes within the view window while executing a script file for the trimming process.
    d) sorting data to arrange coordinates for subsequent plotting.

To overcome the above problems, this paper proposes an integrated approach to the automatic construction of a trimmed stability lobe free of spurious loops and overlapping lobes. In this approach, the lobe data are first organized into Speed-Depth (S-D) corners. Spurious loops are then detected based on the intersection of these S-D corners. Loops within lobes and overlapping segments between adjacent lobes are then removed without any user interaction. As the algorithms are implemented in the form of a single integrated routine and require no data transfer or graphical tools, the proposed approach saves considerable time without losing the critical points in the diagram; this results in a precise stability lobe diagram. The methodology is explained and illustrated with several examples.

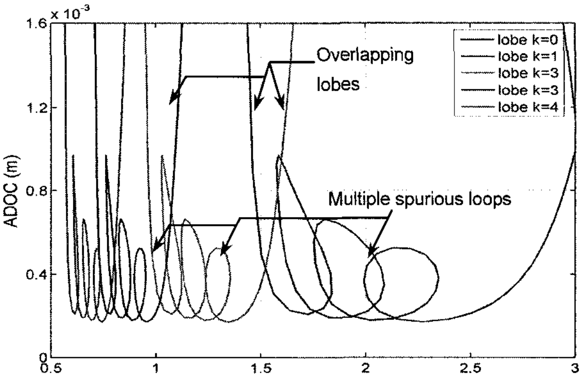


Fig. 1 A stability lobe diagram requiring trimming. ADOC is the axial depth of cut

2. Proposed Methodology

The approach consists of three stages: computation of stability data, identification of S-D corners and removal of loops, and removal of overlapping lobe segments. These are explained in the following sections.

2.1 Computing stability lobe data

The critical values of depth of cut (b) and operating speed (N) can be computed using analytical models.[3-5] By sweeping the frequency ω_c over a desired range ω_{max} at increments of $\Delta\omega_c$, a stability lobe for end milling or turning can be obtained for every k number of vibration waves. The discrete data points $[N, b]$ can then be plotted to form a stability lobe diagram.

2.2 Organizing lobes into speed-depth corners

The stability lobe diagram constructed with the above methods results in spurious loops and overlapping lobes that make the lobe diagram difficult to interpret. With the conventional approaches, recognition of loops and trimming become cumbersome because the loops become more clustered, the points that define loop sections may differ from one lobe to another, and the number of loops may vary.

In this integrated approach, the trimming of lobes starts with the identification of spurious loops by reorganizing the numerical lobe data into Speed-Depth (S-D) corners. A S-D corner is a term used for the combination of a decreasing segment of depth of cut followed by an increasing segment. The following procedure organizes every kth stability lobe obtained from procedure 2.1 into multiple S-D corners.

1. Create a dynamic array matrix to store S-D corners individually in separate sub-matrices. Initialize $n=1$ for the index of the S-D corner. Initialize $i=1$ for the index of the matrix $[N, b]_k$.
2. Compute the difference between b_i and b_{i+1}. If the difference is negative, append $[N_i, b_i]$ to the last row in the nth cell of the S-D corner matrix. Record the sign of difference with a *flag*.
3. If the *flag* goes from positive to negative, complete the assembly of the matrix in the nth cell of the S-D corner. Reinitialize the *flag*. Increment the index n. Make the last point of the previous S-D corner the first point of the current corner.
4. Continue steps 2 and 3 until $i+1<l$ (end of the kth lobe data); otherwise, store the assembled matrix in the nth cell of the S-D matrix. Steps 1–4 thus separate the nested S-D corners into individual matrices.
5. After executing the above steps until $k \le No\ of\ lobes$, the lobe data will be in the form

$$\left\{ \begin{bmatrix} N_{i=1} & \cdots & N_m \\ b_{i=1} & \cdots & b_m \end{bmatrix}'_{n=1} \cdots \begin{bmatrix} N_{i=1} & \cdots & N_m \\ b_{i=1} & \cdots & b_m \end{bmatrix}'_{\substack{speed-depth \\ corners}} \right\}_k .$$

Depending on the complexity of the stability lobe data, the size of the individual matrix m need not be same for all n S-D corners.

2.3 Automatic identification and removal of spurious loops

The above procedure of dividing a lobe into speed-depth corners enables the identification and removal of spurious loops numerically. This process eliminates the requirement to export lobe data to other editing (CAD) software to remove spurious loops and overlapping lobes. Spurious loops can occur under the following conditions:
(i) if the number of S-D corners found in a lobe is greater than 1 and
(ii) if an intersection is found between adjacent S-D corners.

Section 2.2 provides the necessary information for conditions (i) and (ii). If the number of S-D corners is greater than 1, the following procedure is executed to identify and remove spurious loops from the S-D corner information. This requires no manual operation as in conventional methods.

1. Let i and j be the variables that represent the index of the S-D corner matrices $[N, b]_n$ and $[N, b]_{n+1}$ of lobe k, respectively. Initialize i, j, and n to 1.
2. Check for the intersection of line segments. The first line is constructed with points $[N_i, b_i]_n$ and $[N_{i+1}, b_{i+1}]_n$ and the second with points $[N_i, b_i]_{n+1}$ and $[N_{i+1}, b_{i+1}]_{n+1}$. Since physical line segments are not used for trimming, analytical computation may result in an apparent intersection. This is checked by comparing the distance between the starting point of the line segment and the computed intersection point. The intersection routine can not only compute the intersection but it can also handle issues regarding straight lines and common (identical) points.
3. If the intersection is not apparent, remove the superfluous lobe data after the intersection point $[N_{int}, b_{int}]$ and update the endpoint of the nth S-D corner matrix with the intersection point. Similarly, remove the superfluous lobe data of the $(n+1)$th S-D corner matrix and update with the intersection point as the starting point. This has the effect of reducing the size of the related S-D corners $[N, b]$. Proceed to step 6.
4. If either the intersection is apparent or no intersection is found, increment i. Continue to execute the steps 2 and 3 as long as $(i+1) < m_n$.
5. Increment j and continue to execute steps 2–5 if $j+1 < m_{n+1}$.
6. Retain $[N, b]_n$. Increment n and reinitialize i and j to 1.
7. Continue to execute steps 2–6 until $(n+1) \le$ *max no. of speed-depth corners* determined in procedure 2.2.

2.4 Removal of overlapping segments

Overlapping stability lobes are a common problem in stability lobe diagrams. The overlapping segments between adjacent lobes can exist with a common intersection point that can result in a continuous lobe profile upon trimming, or an apparent intersection point that can result in a discontinuous lobe profile upon trimming. The procedure described in section 2.3 can be followed for trimming adjacent lobes with a common intersection point. However, for lobes with an apparent intersection, the following procedure is used.

1. Identify the location of the overlap by analyzing the endpoint of the kth lobe and the starting point of the $(k-1)$th lobe.
2. Based on the location of intersection points, form a virtual vertical line segment with the starting point at the $(k-1)$th lobe and b of the endpoint of the kth lobe.
3. Find the intersection between the vertical line segment and the increasing part of the kth lobe following the algorithm in section 2.3, and remove the overlapping segment defined by the intersection point.

Note that the overlap that occurs in the decreasing part of the $(k-1)$th lobe can be avoided by increasing the frequency range ω_{max}.

3. Results and Discussion

The above steps were implemented in MATLAB, and a user-friendly program was developed to obtain a continuous map of the stability lobes. After computing the stability conditions, data for each lobe were processed to identify spurious loops. Loops are characterized by closed profiles formed with open S-D corners.

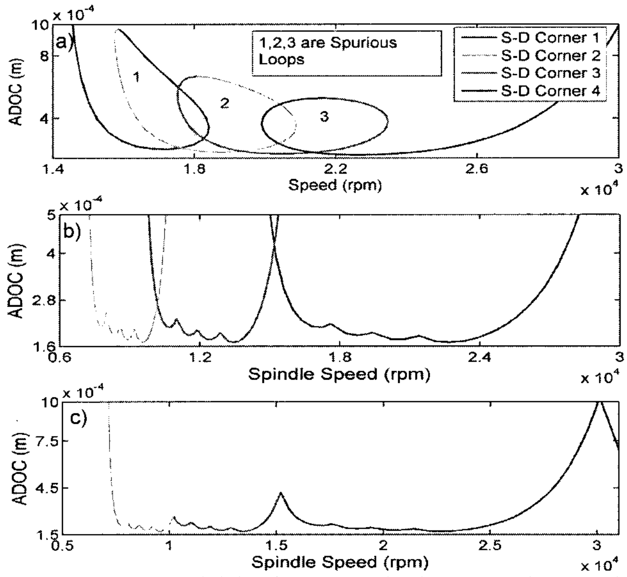


Fig. 2 Lobe after (a) dividing into speed-depth corners, (b) removing spurious loops, and (c) removing overlapping segments

When applied to improving stability lobes in Fig. 1, the program first splits them into multiple S-D corners. For simplicity, only the data for one lobe are shown in Fig. 2(a). Figure 2(b) shows lobes with spurious loops removed according to the method described in section 2.3. Figure 2(c) shows the stability lobe after the removal of the overlapping segments. The sequence of first removing loops and then trimming lobes must be followed to avoid multiple intersections.

In the above lobe diagram, all the S-D corners that occur inside the lobe intersect to form spurious loop(s). This is not always the case as shown in Fig. 3(a). Nonintersecting S-D corners may also coexist, which will result in a mix of loop and no-loop conditions. Figure 3(a) shows lobes with three S-D corners. Since the corners of Lobes 2 and 3 (shown in magenta and cyan) do not intersect to form a closed profile, no spurious loops will occur. However in Lobe 1, S-D corners 2 and 3 intersect to form a spurious loop. The proposed approach correctly detects these spurious loops and removes them as shown in Fig. 3(b). In addition, selecting the minimum allowable depth when the adjacent lobe segments overlap is also necessary. The proper depth of cuts at positions A and B corresponding to lower depth of cuts are identified and retained with this method.

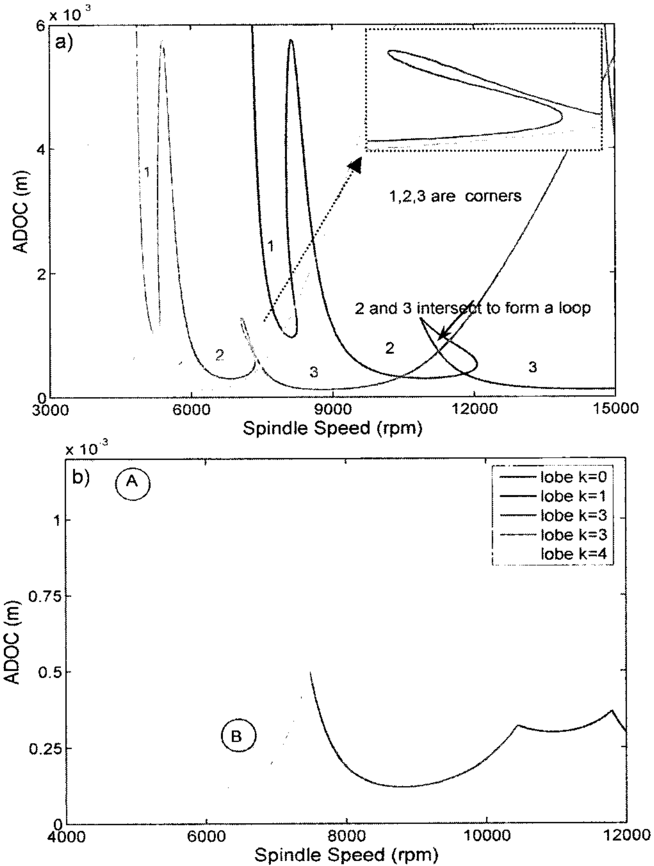
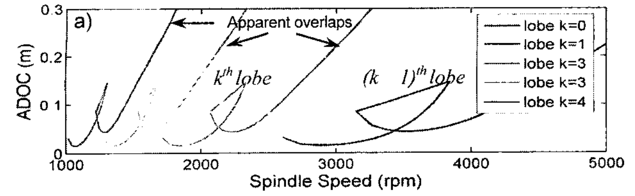


Fig. 3 Stability lobe for milling[4] (a) speed-depth corners and (b) after removing loops and overlapping segments

The stability lobe diagram[1] shown in Fig. 4(a) has lobes that do not intersect but do contain apparent overlaps. Trimming these is desirable to obtain a continuous map of stability conditions. When no intersection is found between lobes, the program invokes the procedure described in section 2.4 to remove the apparent overlapping segments. Figure 4(b) shows the continuous stability map obtained after removing the apparent overlaps.



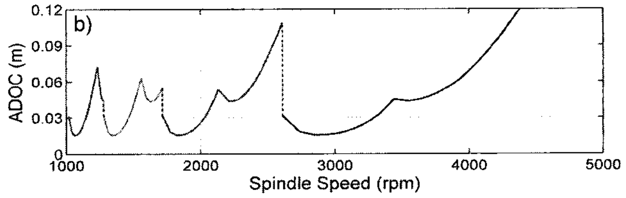


Fig. 4 (a) Nonintersecting adjacent lobes and (b) trimmed lobes

The proposed approach can be applied to generating continuous stability lobe diagrams for turning. Figures 5(a) and (b) show the stability lobe diagram before and after trimming. The machining and modal data for constructing the lobes were obtained from Chen.[5]

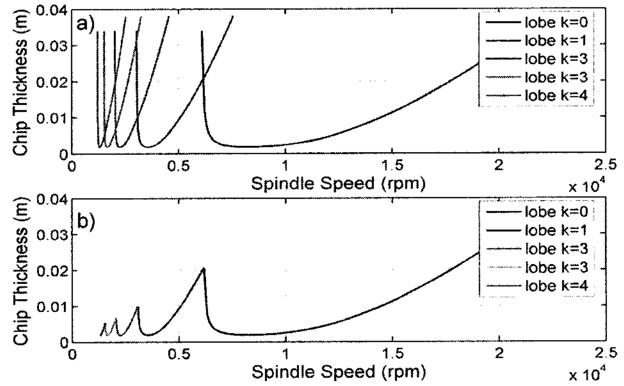


Fig. 5 Stability lobe for turning showing (a) the original lobe diagram with overlapping segments and (b) the final diagram obtained

The time taken to execute the routine is as short as 1 s using a 500-MHz Pentium processor. This integrated approach is completely automatic, highly efficient, and fast. This approach can be applied for offline analysis of feedrate profiles of parametric interpolators.[6]

4. Conclusions

This paper has presented an integrated programmatic approach to preparing continuous stability lobe diagrams. The proposed approach can automatically detect multiple spurious loops by reorganizing each lobe into speed-depth corners. Since this method requires no user interaction or data transfer, it produces precise and clear stability diagrams quickly and automatically. Hence this approach can be used to predict favorable cutting conditions for milling and turning processes in real time. Moreover, this methodology could be extended to construct a three-dimensional continuous stability lobe surface.

REFERENCES

1. Tlusty, J., "Manufacturing Processes and Equipment," Prentice-Hall, pp. 559-594, 1999.
2. Schmitz, T. L., "Automatic Trimming of Machining Stability Lobes," Intl. Journal of Machine Tools and Manufacture, Vol. 42, No. 13, pp. 801-813, 2002.
3. Altintas, Y. and Budak, E., "Analytical Prediction of Stability Lobes in Milling," CIRP, Vol. 44, No. 1, pp. 357-362, 1995.
4. Tekeli, A. and Budak, E., "Maximization of Chatter-Free Material Removal Rate in End Milling using Analytical Methods," Machining Science and Tech., Vol. 9, No. 22, pp. 147-167, 2005.
5. Chen, C. K. and Tsao, Y. M., "A Stability Analysis of Turning a Tailstock Supported Flexible Work-Piece," Intl. Journal of Machine Tools and Manufacture, Vol. 46, No. 11, pp. 18-25, 2006.
6. Sekar, M., Kweon S. H., Lee, D. M. and Yang, S. H., "Parametric NURBS Curve Interpolators: A Review," Intl Journal of Precision Engineering and Manufacturing, Vol. 9, No. 2, pp. 84-92, 2008.