INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2004; **61**:117–141 (DOI: 10.1002/nme.1061)

Updated semi-discretization method for periodic delay-differential equations with discrete delay

更新的离散时滞周期微分方程半离散化方法

Tamás Insperger*,†,‡,§ and Gábor Stépán¶,‖,**

Department of Applied Mechanics, Budapest University of Technology and Economics, Budapest H-1521, Hungary

将延迟项近似为两个相邻离散延迟状态值的加权和，确定了单个周期内的转移矩阵。
建立了不同时延比下阻尼时滞马休方程的稳定性图。

SUMMARY

单离散时滞周期系统

An updated version of the semi-discretization method is presented for periodic systems with a single discrete time delay. The delayed term is approximated as a weighted sum of two neighbouring discrete delayed state values and the transition matrix over a single period is determined. Stability charts are constructed for the damped and delayed Mathieu equation for different time-period/time-delay ratios. The convergence of the method is investigated by examples. Stability charts are constructed for 1 and 2 degree of freedom milling models. The codes of the algorithm are also attached in the appendix. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: time delay; periodic system; Floquet theory; linear stability

1. INTRODUCTION

时滞微分方程

Systems, where the rate of change of state depends not only on the present state of the system, but also on the past states are described by delay-differential equations (DDEs). One of the classical examples for delayed system is the predator–prey model of Volterra [1], where the growth rate of predators depends not only on the present quality of food (say, prey), but also on the past quantities (in the period of gestation, say). The first delay models in engineering were introduced by von Schlippe and Dietrich [2] for modelling wheel shimmy, and by

*Correspondence to: Tamás Insperger, Department of Applied Mechanics, Budapest University of Technology and Economics, Budapest H-1521, Hungary.
†E-mail: inspi@mm.bme.hu
‡Assistant Professor
§http://www.mm.bme.hu/~inspi
¶Professor, Head of Department of Applied Mechanics
‖E-mail: stepan@mm.bme.hu
**http://www.mm.bme.hu/~stepan

Contract/grant sponsor: Hungarian National Science Foundation; contract/grant number: OTKA F047318 and OTKA T043368
Contract/grant sponsor: Foundation for Hungarian Higher Education and Research

Received 4 September 2003
Revised 16 January 2004
Accepted 24 January 2004

Published online 21 July 2004

Minorsky [3] for ship stabilization. Time delay is also a key element of the explanation of machine tool chatter [4, 5], more and more sophisticated models have appeared for turning and milling applications in the last decade [6–14]. Delayed equations arise in several fields of science: in the digital control of robotics applications with information delay [15, 16], in neural network models, where the interactions of the neurons are delayed [17], or in delayed feedback control mechanism of human balancing [18, 19]. Since the publication of the first delayed models, several books have appeared summarizing the most important theorems of DDEs [20–23]. The stability of a linear DDE depends on its (infinite number of) characteristic roots: if and only if all of these roots are located in the left side of the complex plane, the system is asymptotically stable. Due to the infinite number of characteristic roots, the stability properties of delayed systems might be extremely complex [21, 24].

复平面

If a parameter of a system changes periodically in time, then the corresponding equation of motion is a time periodic differential equation. According to the Floquet theory, the stability of a linear periodic system depends on the characteristic multipliers: if all of these multipliers are located in the inside the unit disc of the complex plane, then the system is asymptotically stable [25].

单位圆

Recently, the investigation of periodic motion of DDEs and the associated variational systems (that are time periodic DDEs) has come into the focus of interest [26–29]. One of the most important motivations is the milling process analysis: if both the so-called tooth pass excitation effect and the regenerative effect are considered, then the variational system associated to the ideal periodic motion of the tool is a time periodic DDE. Several papers deal with the exploration of the dynamic behaviour of the milling process [6, 9, 11–14]. Parametric excitation is also used to avoid vibrations. A turning process can be stabilized by the periodic variation of the spindle speed [30–32], or by changing the system stiffness periodically [7]. In robotic applications, the robustness of feedback control can be increased by periodic variation of the gain parameters [16]. These systems are all governed by time periodic DDEs.

An analytical stability chart for the delayed Mathieu equation

$$\ddot{x}(t) + (\delta + \varepsilon \cos(t))x(t) = bx(t - 2\pi) \tag{1}$$

was given by Insperger and Stépán [33]. They also confirmed the analytical results by the numerical technique called semi-discretization [34]. Butcher *et al.* [35] have also checked the results regarding the stability chart of the delayed Mathieu equation, they used the shifted Chebyshev polynomial method [36].

The semi-discretization of delayed systems was introduced by Insperger and Stépán [37]. The point of the method is that only the delayed terms are discretized while the actual time domain terms are unchanged as opposed to the full-discretization techniques [38]. This results in a finite-dimensional discrete map approximation of the DDE. An extension of the semi-discretization method to delayed systems subjected to stochastic disturbance was developed by Elbeyli *et al.* [39].

In the paper [37], the semi-discretization method is explained for general delayed systems, where the delay can also be distributed along the past. The application of the method for systems with discrete time delays may lead to intolerable inaccuracy (or alternatively, intolerable computation time) at certain narrow parameter domains. One possible solution for this problem is presented in Reference [34] for the damped Mathieu equation with discrete time delay. There, the time delay τ was equal to the time period T, but the time step of the discretization with respect to the delay was $\Delta t = \tau/(m + \frac{1}{2})$, where m was an integer. Consequently, the time

period T was not an integer multiple of the time step Δt, therefore, the transition matrix $\mathbf{\Phi}$ over the time period $T = \tau = (m + \frac{1}{2})\Delta t$ could not be determined in accurate closed form. In order to avoid this problem, the solution was determined over the double period $2T = (2m + 1)\Delta t$ that resulted in a closed form expression for the square of the transition matrix: $\mathbf{\Phi}^2$. The eigenvalues of $\mathbf{\Phi}^2$ are the square of the characteristic multipliers: μ^2. The system was said to be stable if $|\mu^2| < 1$, which is equivalent to $|\mu| < 1$. Although, this method resulted in accurate stability charts, the computation time was doubled compared to the solution over a single period. Another problem was that during the double period analysis, only the square of the characteristic multipliers were obtained. This is perfect for stability prediction, but more information is needed for the subsequent bifurcation analysis. E.g., the cases of period one ($\mu = 1$) and period two ($\mu = -1$) instabilities cannot be distinguished, since both cases are obtained by the computation as $\mu^2 = 1$. The determination of the vibration frequencies are not trivial either since pseudo-roots occur when the complex $\sqrt{\mu^2}$ is calculated.

In the present paper, an updated version of the semi-discretization method is presented for time periodic DDEs of optional time period and time delay. The main point is that the delayed term in question is approximated as a weighted sum of the two neighbouring discrete delayed state values. With this technique, the time step Δt can still be chosen as an integer fraction of the time period T, the solution can be determined over a single period, that results in a closed form expression for the transition matrix: $\mathbf{\Phi}$. The updated method is presented here for DDEs with a single constant delay, however, the method can be generalized for DDEs with multiple time dependent delays and also for DDEs with distributed delays.

As a reference case, the damped and delayed Mathieu equation is investigated. Stability charts are determined for different ratios of the time delay and the time period including the case, when this ratio is irrational (e.g. $\sqrt{2}$). These charts are the combinations of the Hsu–Bhatt–Vyshnegradskii chart of the second order delay-differential equation [40] and the Strutt–Ince chart of the classical Mathieu equation [41, 42].

As examples for higher-order systems, stability charts are determined for 1 and 2 degree of freedom (DOF) milling models. The codes of the algorithm are also attached in the appendix.

2. UPDATED SEMI-DISCRETIZATION METHOD

The n-dimensional delay differential equation

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{x}(t - \tau), \quad \mathbf{A}(t + T) = \mathbf{A}(t), \quad \mathbf{B}(t + T) = \mathbf{B}(t) \tag{2}$$

is investigated, where the time delay τ and the time period T are optional positive values.

The first step of semi-discretization is the construction of the time interval division $[t_i, t_{i+1}]$ of length Δt, $i = 0, 1, \ldots$ so that $T = k\Delta t$, where k is an integer that can be considered as an approximation parameter regarding the time period.

Introduce the integer m so that

$$m = \text{int}\left(\frac{\tau + \Delta t/2}{\Delta t}\right) \tag{3}$$

where int is the function that rounds positive numbers towards zero (e.g. $\text{int}(4.82) = 4$). The integer m can be considered as an approximation parameter regarding the time delay.

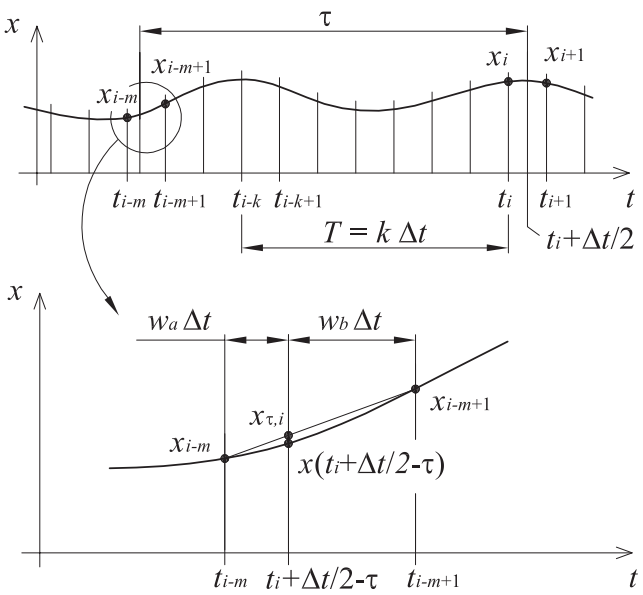

Figure 1. Approximation of the delayed term.

Use the notation $\mathbf{x}(t_j) = \mathbf{x}_j$ for any integer j. In the ith interval, Equation (2) can be approximated as

$$\dot{\mathbf{x}}(t) = \mathbf{A}_i \mathbf{x}(t) + \mathbf{B}_i \mathbf{x}_{\tau,i} \tag{4}$$

where

$$\mathbf{A}_i = \frac{1}{\Delta t}\int_{t_i}^{t_{i+1}} \mathbf{A}(t)\,\mathrm{d}t, \quad \mathbf{B}_i = \frac{1}{\Delta t}\int_{t_i}^{t_{i+1}} \mathbf{B}(t)\,\mathrm{d}t \tag{5}$$

and $\mathbf{x}_\tau$ is the following approximation of the delayed term: 延迟项

$$\mathbf{x}(t-\tau) \approx \mathbf{x}(t_i + \Delta t/2 - \tau) \approx w_b \mathbf{x}_{i-m} + w_a \mathbf{x}_{i-m+1} = \mathbf{x}_{\tau,i} \tag{6}$$

where the weights of $\mathbf{x}_{i-m}$ and $\mathbf{x}_{i-m+1}$ are

$$w_b = \frac{\tau + \Delta t/2 - m\Delta t}{\Delta t} \tag{7}$$

$$w_a = \frac{m\Delta t + \Delta t/2 - \tau}{\Delta t} \tag{8}$$

诀窍
The trick in the approximation is that the delayed term is approximated as a weighted linear combination of the delayed discrete values $\mathbf{x}_{i-m}$ and $\mathbf{x}_{i-m+1}$ (see Figure 1). With the help of the usual Lagrange remainder term, an error estimation can be constructed in the usual way for this natural choice of weights. 拉格朗日剩余项

The solution of Equation (4) for the initial condition

$$\mathbf{x}(t_i) = \mathbf{x}_i \tag{9}$$

reads

$$\mathbf{x}(t) = \exp(\mathbf{A}_i(t - t_i))(\mathbf{x}_i + \mathbf{A}_i^{-1}\mathbf{B}_i\mathbf{x}_{\tau,i}) - \mathbf{A}_i^{-1}\mathbf{B}_i\mathbf{x}_{\tau,i} \tag{10}$$

Substituting $t = t_{i+1}$ and using Equation (6), $\mathbf{x}_{i+1} = \mathbf{x}(t_{i+1})$ is defined as

$$\mathbf{x}_{i+1} = \mathbf{P}_i\mathbf{x}_i + w_a\mathbf{R}_i\mathbf{x}_{i-m+1} + w_b\mathbf{R}_i\mathbf{x}_{i-m} \tag{11}$$

where

$$\begin{cases} \mathbf{P}_i = \exp(\mathbf{A}_i\Delta t) \\ \mathbf{R}_i = (\exp(\mathbf{A}_i\Delta t) - \mathbf{I})\mathbf{A}_i^{-1}\mathbf{B}_i \end{cases}$$

单位矩阵

Here, $\mathbf{I}$ denotes identity matrix.

Now, according to Equation (11), a discrete map can be defined

$$\mathbf{y}_{i+1} = \mathbf{C}_i\mathbf{y}_i \tag{12}$$

where the $n(m+1)$-dimensional vector is

$$\mathbf{y}_i = \mathrm{col}(\mathbf{x}_i \;\; \mathbf{x}_{i-1} \;\; \ldots \;\; \mathbf{x}_{i-m}) \tag{13}$$

and the coefficient matrix has the form

$$\mathbf{C}_i = \begin{pmatrix} \mathbf{P}_i & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & w_a\mathbf{R}_i & w_b\mathbf{R}_i \\ \mathbf{I} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{I} & \mathbf{0} \end{pmatrix} \tag{14}$$

The next step is to determine the transition matrix $\mathbf{\Phi}$ over the principal period $T = k\Delta t$. This serves a finite-dimensional approximation of the monodromy operator in the infinite-dimensional version of the Floquet Theory [22, 25]. The transition matrix gives the connection between $\mathbf{y}_0$ and $\mathbf{y}_k$ in the form

$$\mathbf{y}_k = \mathbf{\Phi}\mathbf{y}_0 \tag{15}$$

where $\mathbf{\Phi}$ is given by coupling the solutions

$$\mathbf{\Phi} = \mathbf{C}_{k-1}\mathbf{C}_{k-2}\ldots\mathbf{C}_1\mathbf{C}_0 \tag{16}$$

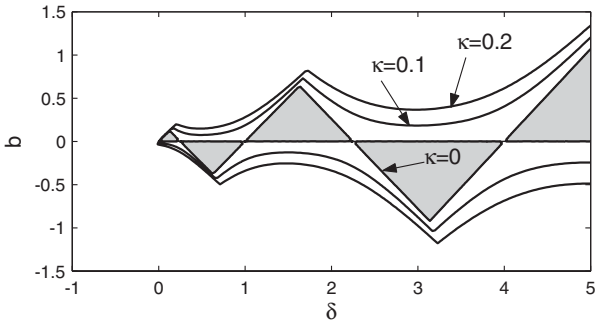

Figure 2. Hsu–Bhatt–Vyshnegradskii stability chart of Equation (17) with $\varepsilon = 0$ and $\tau = 2\pi$.

Note, that the integer k determines the number of matrices to be multiplied in Equation (16), and m determines the size of these matrices. For systems with large principal period (i.e. with large k) and with large delay (i.e. with large m), the determination of the transition matrix might be time consuming.

Now, the stability investigation is reduced to the problem, whether the eigenvalues of $\mathbf{\Phi}$ are in modulus less than one [43].

3. DAMPED DELAYED MATHIEU EQUATION 阻尼滞后马蒂厄方程

The equation 此公式缘何而来？？

$$\ddot{x}(t) + \kappa\dot{x}(t) + (\delta + \varepsilon\cos(2\pi t/T))x(t) = bx(t-\tau) \tag{17}$$

is investigated. The time period of the periodic coefficient is T, the time delay is τ.

For the autonomous case $\varepsilon = 0$ of Equation (17), the so-called Hsu–Bhatt–Vyshnegradskii stability chart is shown in Figure 2 [40]. For the undamped case ($\kappa = 0$), the domain of asymptotic stability is denoted by grey colour, and the stability boundaries for the cases $\kappa = 0.1$ and 0.2 are also presented.

For the case $b = 0$ of Equation (17), the Strutt–Ince stability chart is presented in Figure 3 for different dampings κ.

For the semi-discretization procedure, use the notation $x(t_j) = x_j$ for any integer j. In the ith semi-discretization interval, Equation (17) can be approximated as

$$\ddot{x}(t) + \kappa\dot{x}(t) + (\delta + \varepsilon c_i)x(t) = bx_{\tau,i} \tag{18}$$

where

$$c_i = \frac{1}{\Delta t}\int_{t_i}^{t_{i+1}} \cos(2\pi t/T)\,\mathrm{d}t \tag{19}$$

and

$$x_{\tau,i} = w_b x_{i-m} + w_a x_{i-m+1} \approx x(t_i + \Delta t/2 - \tau) \approx x(t-\tau) \tag{20}$$

where m, w_b and w_a are defined by Equations (3), (7) and (8), respectively.

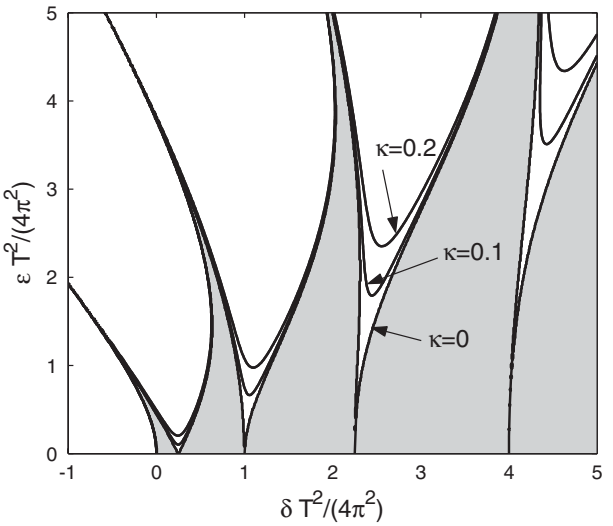

Figure 3. Strutt–Ince stability chart of Equation (17) with $b = 0$.

By Cauchy transformation, the following system is obtained:

$$\dot{\mathbf{u}}(t) = \mathbf{A}_i \mathbf{u}(t) + w_a \mathbf{B}\mathbf{u}_{i-m+1} + w_b \mathbf{B}\mathbf{u}_{i-m} \tag{21}$$

where

$$\mathbf{A}_i = \begin{pmatrix} 0 & 1 \\ -(\delta + \varepsilon c_i) & -\kappa \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 0 & 0 \\ b & 0 \end{pmatrix}, \quad \mathbf{u}(t) = \begin{pmatrix} x(t) \\ \dot{x}(t) \end{pmatrix} \tag{22}$$

and

$$\mathbf{u}_j = \mathbf{u}(t_j) = \begin{pmatrix} x(t_j) \\ \dot{x}(t_j) \end{pmatrix} = \begin{pmatrix} x_j \\ \dot{x}_j \end{pmatrix}$$

for any integer j. Introduce the notation

$$\mathbf{u}_{\tau,i} = w_a \mathbf{u}_{i-m+1} + w_b \mathbf{u}_{i-m} \tag{23}$$

Then the solution of Equation (21) for the initial condition

$$\mathbf{u}(t_i) = \mathbf{u}_i \tag{24}$$

reads

$$\mathbf{u}(t) = \exp(\mathbf{A}_i(t - t_i))(\mathbf{u}_i + \mathbf{A}_i^{-1}\mathbf{B}\mathbf{u}_{\tau,i}) - \mathbf{A}_i^{-1}\mathbf{B}\mathbf{u}_{\tau,i} \tag{25}$$

Substituting $t = t_{i+1}$ and using Equation (23), $\mathbf{u}_{i+1} = \mathbf{u}(t_{i+1})$ is defined as

$$\mathbf{u}_{i+1} = \mathbf{P}_i \mathbf{u}_i + w_a \mathbf{R}_i \mathbf{u}_{i-m+1} + w_b \mathbf{R}_i \mathbf{u}_{i-m} \tag{26}$$

where

$$\mathbf{P}_i = \exp(\mathbf{A}_i \Delta t)$$

$$\mathbf{R}_i = (\exp(\mathbf{A}_i \Delta t) - \mathbf{I})\mathbf{A}_i^{-1}\mathbf{B}$$

and $\mathbf{I}$ denotes identity matrix. Now, a discrete map can be constructed similarly to Equation (12).

At this point, it should be noticed, that $\mathbf{u}_{i+1}$ depends on x_i, $\dot{x}_i$, x_{i-m+1} and x_{i-m}, but it does not depend on $\dot{x}_{i-m+1}$ and $\dot{x}_{i-m}$. The reason is that since $\dot{x}(t-\tau)$ does not appear in Equation (17), the 2nd column of matrix $\mathbf{B}$ is zero, consequently, the 2nd column of matrix $\mathbf{R}_i$ is also zero. This means that for the discrete map, the $(m+2)$-dimensional state vector

$$\mathbf{z}_i = \mathrm{col}(x_i \;\; \dot{x}_i \;\; x_{i-1} \;\; \ldots \;\; x_{i-m}) \tag{27}$$

should be defined instead of the $2(m+1)$-dimensional vector

$$\mathbf{y}_i = \mathrm{col}(\mathbf{u}_i \;\; \mathbf{u}_{i-1} \;\; \ldots \;\; \mathbf{u}_{i-m}) = \mathrm{col}(x_i \;\; \dot{x}_i \;\; x_{i-1} \;\; \dot{x}_{i-1} \;\; \ldots \;\; x_{i-m} \;\; \dot{x}_{i-m}) \tag{28}$$

defined by Equation (13). This trick makes the size of the resulting discrete map smaller:

$$\mathbf{z}_{i+1} = \mathbf{D}_i \mathbf{z}_i \tag{29}$$

where the $(m+2)$-dimensional coefficient matrix reads

$$\mathbf{D}_i = \begin{pmatrix} P_{i,11} & P_{i,12} & 0 & 0 & \ldots & 0 & w_a R_{i,11} & w_b R_{i,11} \\ P_{i,21} & P_{i,22} & 0 & 0 & \ldots & 0 & w_a R_{i,21} & w_b R_{i,21} \\ 1 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \ddots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & 1 & 0 \end{pmatrix} \tag{30}$$

Here, $P_{i,hj}$ and $R_{i,hj}$ are the elements of matrices $\mathbf{P}_i$ and $\mathbf{R}_i$ in the hth row and in the jth column, respectively.

The transition matrix $\mathbf{\Phi}$ over the principal period $T = k\Delta t$ is determined now by coupling Equation (29) for $i = 0, 1, \ldots, k-1$:

$$\mathbf{\Phi} = \mathbf{D}_{k-1}\mathbf{D}_{k-2}\ldots\mathbf{D}_1\mathbf{D}_0 \tag{31}$$

If the eigenvalues of $\mathbf{\Phi}$ are in modulus less than one, then the system is stable.

In Figure 4, stability charts of Equation (17) are shown for different parameters. The case when the time period and the time delay are integer multiple of each other (e.g. $T = 2\tau$), and the case when their ratio is irrational (e.g. $T = \sqrt{2}\tau$) are also considered. For all the charts in Figure 4, the time delay is $\tau = 2\pi$, while the time period and the amplitude ε are different. In Figure 4(a) and (b), the time period is $T = \pi = \tau/2$, in Figure 4(c) and (d), it is $T = \sqrt{2}\pi = \tau/\sqrt{2}$, in Figure 4(e) and (f), $T = 2\pi = \tau$, in Figure 4(g) and (h), $T = 2\sqrt{2}\pi = \sqrt{2}\tau$, and in Figure 4(i) and (j), $T = 4\pi = 2\tau$. In Figure 4(a), (c), (e), (g) and (i) (i.e. in the charts on the left side), the amplitude is $\varepsilon = 1$, while in Figure 4(b), (d), (f), (h) and (j) (i.e. in the charts on the right side), it is $\varepsilon = 2$. For the undamped case ($\kappa = 0$), the domain of asymptotic stability is denoted by grey colour. The stability boundaries for the cases $\kappa = 0.1$ and 0.2 are also presented. The charts were determined via point-by-point evaluation of the transition matrix over a 200×100 sized grid. The Matlab code for the stability analysis is given in Appendix A.

The number of discretization intervals is $k = 40$ for all charts in Figure 4.

For the case $T = \tau/2 = \pi$ (Figure 4(a) and (b)), Equation (3) results in $m = 80$. This means that the transition matrix $\mathbf{\Phi}$ is obtained by multiplication of 40 number of 82×82 sized matrices. For this case, Equations (7) and (8) result in the weights $w_b = \frac{1}{2}$ and $w_a = \frac{1}{2}$. The computation time for determining one stability chart over a 200×100 sized grid of parameters was 22.6 min on a 2 GHz Pentium 4 processor.

For the case $T = \tau/\sqrt{2} = \sqrt{2}\pi$ (Figure 4(c) and (d)), Equation (3) results in $m = 57$, that is, 40 number of 59×59 sized matrices is multiplied to obtain the transition matrix $\mathbf{\Phi}$. For this case, the weights are $w_b = 0.0685$ and $w_a = 0.9315$. In this case, the computation time was 10.4 min.

For the case $T = \tau = 2\pi$ (Figure 4(e) and (f)), Equation (3) results in $m = 40$, and 40 number of 42×42 sized matrices is multiplied to obtain the transition matrix $\mathbf{\Phi}$. For this case, the weights are $w_b = \frac{1}{2}$ and $w_a = \frac{1}{2}$, again, and the computation time was 6.6 min.

For the case $T = \sqrt{2}\tau = 2\sqrt{2}\pi$ (Figure 4(g) and (h)), Equation (3) results in $m = 28$. Here, 40 number of 30×30 sized matrices is multiplied to obtain the transition matrix $\mathbf{\Phi}$, and the weights are $w_b = 0.7843$ and $w_a = 0.2157$. Here, the computation time was 4.6 min.

Finally, for the case $T = 2\tau = 4\pi$ (Figure 4(i) and (j)), Equation (3) results in $m = 20$, and 40 number of 22×22 sized matrices is multiplied to obtain the transition matrix $\mathbf{\Phi}$ and again, the weights are $w_b = \frac{1}{2}$ and $w_a = \frac{1}{2}$. The computation time was 3.8 min.

For the case $T = \tau$ with $\kappa = 0$ (Figure 4(e) and (f)), the stability boundaries are shown to be straight lines with slopes $+1$, -1 and 0, while for increasing damping $\kappa > 0$, some lines of slope ± 1 remains, and curved stability boundaries arise. This is in accordance with the analytical result presented in References [33, 34]. Here, the same phenomena can be observed for the case $T = \tau/2$ (Figure 4(a) and (b)), but for the other cases $T = \tau/\sqrt{2}$, $T = \sqrt{2}\tau$ and $T = 2\tau$, the stability boundaries shows a complex structure.

In order to explain the complexity of the charts in Figure 4, Figures 2 and 3 should be analysed. For the undamped case $\kappa = 0$ in Figure 2, the stability boundaries intersects the $b = 0$ line at $\frac{1}{4}$, 1, $\frac{9}{4}$, 4, etc. In Figure 3, the $\kappa = 0$ stability boundaries intersects the $\varepsilon = 0$ line at π^2/T^2, $4\pi^2/T^2$, $9\pi^2/T^2$, etc. If the time period and the time delay are equal ($T = \tau = 2\pi$), then the intersection points are the same. This case is clear and the stability charts are known in closed form [33, 34]. If $T = \tau/2 = \pi$, then these special points in Figure 3 are 1, 4, 9, etc., that is, every intersection point of the Hsu–Bhatt–Vyshnegradskii chart is

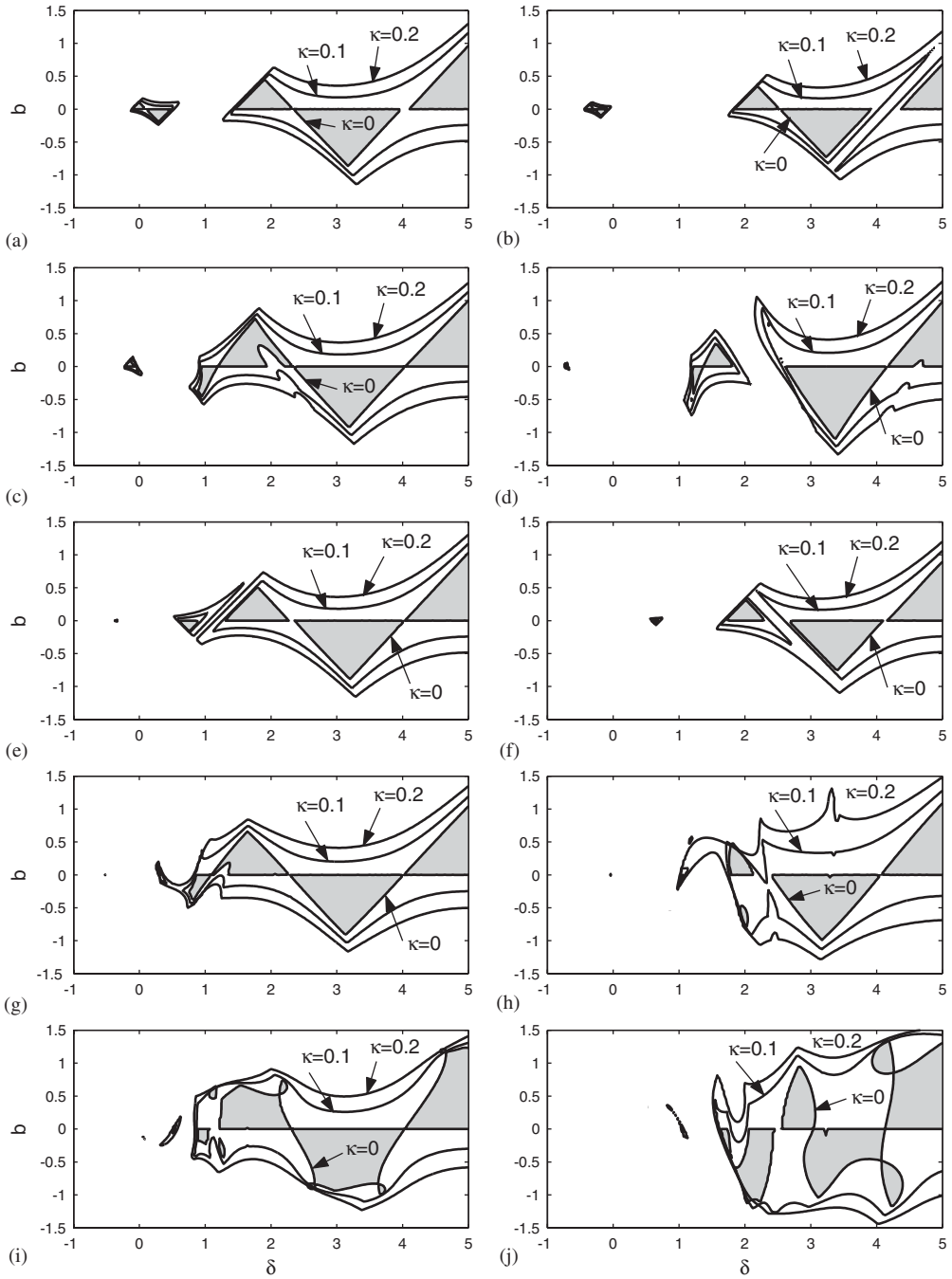

Figure 4. Stability charts of Equation (17) with $\tau = 2\pi$ and $T = \pi$ (a, b), $T = \sqrt{2}\pi$ (c, d), $T = 2\pi$ (e, f), $T = 2\sqrt{2}\pi$ (g, h), $T = 4\pi$ (i, j) and $\varepsilon = 1$ (a, c, e, g, i), $\varepsilon = 2$ (b, d, f, h, j).

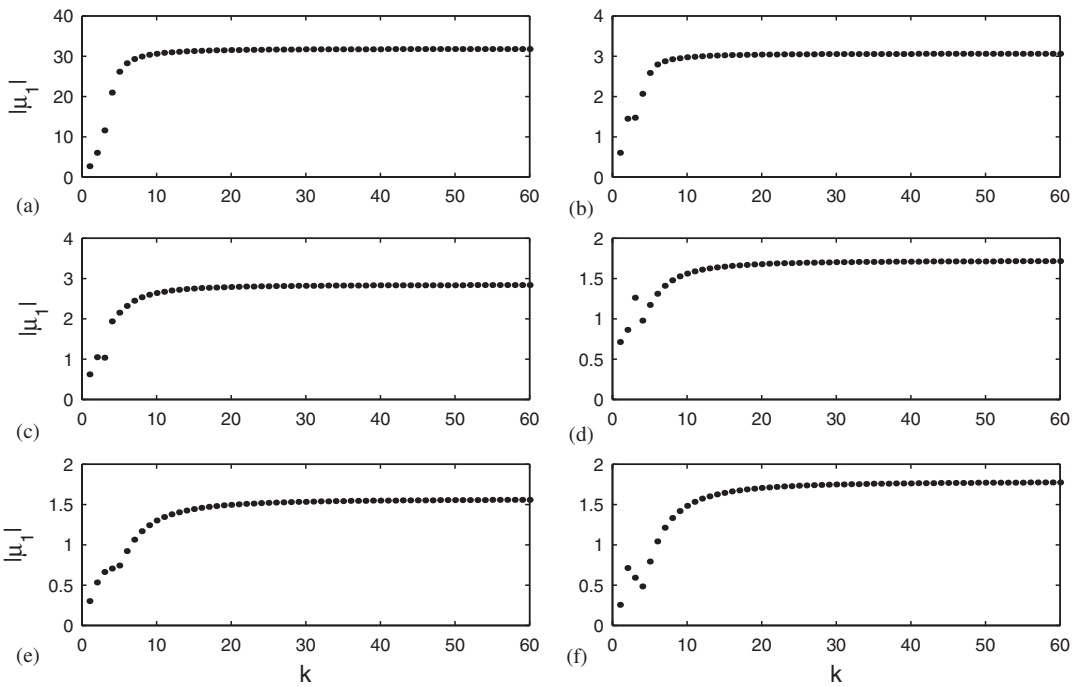

Figure 5. Modulus of critical eigenvalues for Equation (17) with $b = -1.5$, $\kappa = 0.2$, $\varepsilon = 2$, $\tau = T = 2\pi$ and $\delta = 0$ (a), 1 (b), 2 (c), 3 (d), 4 (e), 5 (f) as function of the number k of the discretization intervals.

an intersection point of the Strutt–Ince chart. This explains the linear stability boundaries in Figure 4(a) and (b). For the case $T = \tau/\sqrt{2} = \sqrt{2}\pi$ in Figure 4(c) and (d), the special points for the corresponding Strutt–Ince chart are $\frac{1}{2}$, 2, $\frac{9}{2}$, etc. These intersection points differs from those of the corresponding Hsu–Bhatt–Vyshnegradskii chart, and the resulted stability charts in Figure 4(c) and (d) show a kind of split structure around $\delta = \frac{1}{2}$, 2, $\frac{9}{2}$, etc. at the range $b \approx 0$. Similarly, for the other cases, the intersection points of the corresponding Hsu–Bhatt–Vyshnegradskii and Strutt–Ince charts are not the same, and the combined stability charts have a kind of intersecting structure.

For some cases, the convergence of the critical eigenvalues can be observed in Figure 5. Here, the modulus of the critical eigenvalues $|\mu_1|$ for Equation (17) are presented as function of the number k of the discretization intervals. In Figure 5(a)–(f), the parameter δ is 0, 1, 2, 3, 4 and 5, respectively. The other parameters are $b = -1.5$, $\kappa = 0.2$, $\varepsilon = 2$ and $\tau = T = 2\pi$. The plots show, how the eigenvalues converge to a fixed value as k increase. The difference between the critical values $|\mu_1|$ for $k = 40$ and 60 are 0.1261% for the case (a) in Figure 5, 0.1040% for case (b), 0.2582% for case (c), 0.3306% for case (d), 0.6269% for case (e) and 0.6174% for case (f).

4. 1 DOF MILLING EQUATION

The governing equation of motion of a 1 DOF milling model reads [44]

$$\ddot{x}(t) + 2\zeta\omega_n\dot{x}(t) + \omega_n^2 x(t) = -\frac{wh(t)}{m_t}(x(t) - x(t-\tau)) \tag{32}$$

where ω_n is the angular natural frequency, ζ is the relative damping, w is the depth of cut, m_t is the modal mass of the tool. The delayed term $x(t-\tau)$ arise due to the regenerative effect. The time delay is equal to the tooth passing period. The specific cutting force coefficient $h(t)$ is determined by the technological parameters

$$h(t) = \sum_{j=1}^{N} g(\phi_j(t))\sin(\phi_j(t))(K_t\cos(\phi_j(t)) + K_n\sin(\phi_j(t))) \tag{33}$$

where N is the number of teeth, K_t and K_n are the tangential and the normal linearized cutting force coefficients, respectively, and $\phi_j(t)$ is the angular position of tooth j defined as

$$\phi_j(t) = (2\pi\Omega/60)t + j2\pi/N \tag{34}$$

where Ω is the spindle speed in rpm. The function $g(\phi_j(t))$ is a screen function, it is equal 1, if the tooth j is in the cut, and it is equal to 0, if tooth j is out of cut:

$$g(\phi_j(t)) = \begin{cases} 1 & \text{if } \phi_{st} < \phi_j(t) < \phi_{ex} \\ 0 & \text{otherwise} \end{cases} \tag{35}$$

where ϕ_{st} and ϕ_{ex} is the start and exit angles of the tooth j. For up-milling, $\phi_{st} = 0$ and $\phi_{ex} = \arccos(1 - 2a/D)$, for down-milling, $\phi_{st} = \arccos(2a/D - 1)$ and $\phi_{ex} = \pi$, where a/D is the radial depth of cut ratio.

Due to the tooth pass excitation effect, $h(t)$ is time periodic with the tooth passing period τ, that is, for milling processes, the time delay is equal to the time period. Consequently, Equations (3), (7) and (8) result in the approximation parameter $m = k$ and the weights $w_b = \frac{1}{2}$ and $w_a = \frac{1}{2}$. For the spindle speed Ω (rpm), the time delay is $\tau = T = 60/(N\Omega)$ (s), and the time step is $\Delta t = T/k = 60/(kN\Omega)$ (s).

Equation (32) is similar to the damped delayed Mathieu equation (17), the only difference is that the delayed term has a time periodic coefficient. The discretized equation has the form

$$\ddot{x}(t) + 2\zeta\omega_n\dot{x}(t) + \left(\omega_n^2 + \frac{wh_i}{m_t}\right)x(t) = \frac{wh_i}{m_t}x_{\tau,i}, \quad t \in [t_i, t_{i+1}] \tag{36}$$

where

$$h_i = \frac{1}{\Delta t}\int_{t_i}^{t_{i+1}} h(t)\,\mathrm{d}t \tag{37}$$

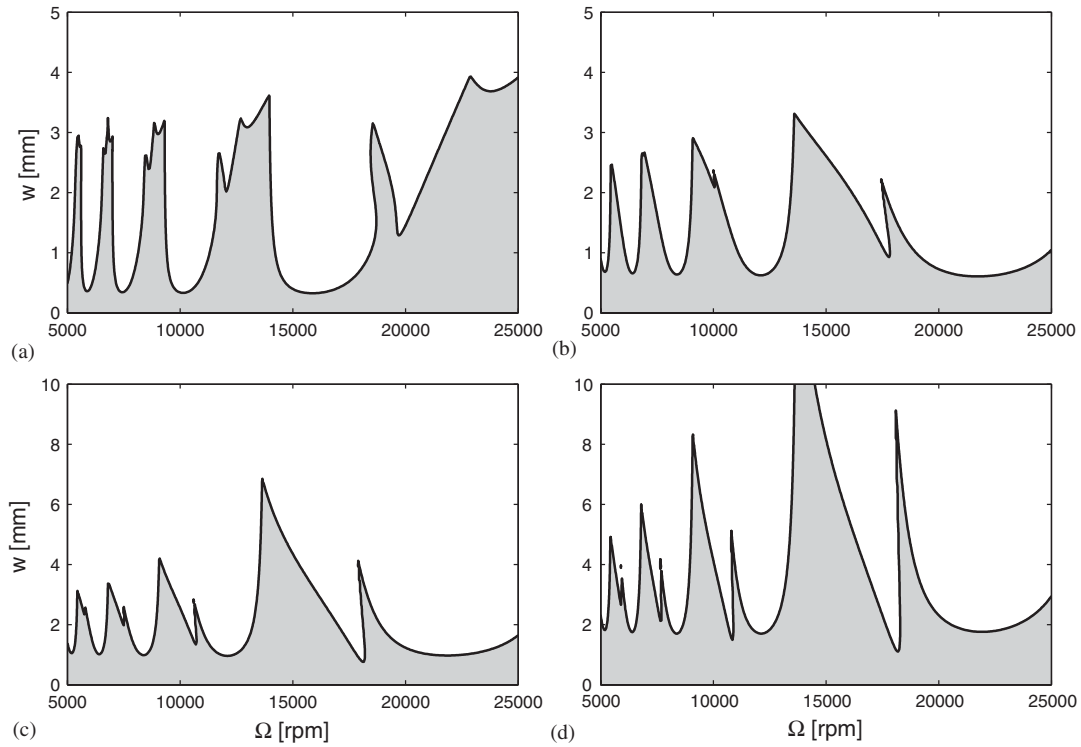

Figure 6. Stability charts for 1 DOF down-milling model with radial depth of cut ratio $a/D = 1$ (a), $a/D = 0.5$ (b), $a/D = 0.1$ (c), $a/D = 0.05$ (d).

and $x_{\tau,i}$ is defined as in Equation (20). From that point, the same discretization steps should be done as for the damped delayed Mathieu equation (17) with the only difference that

$$\mathbf{A}_i = \begin{pmatrix} 0 & 1 \\ -\left(\omega_n^2 + \dfrac{wh_i}{m_t}\right) & -2\zeta\omega_n \end{pmatrix}, \quad \mathbf{B}_i = \begin{pmatrix} 0 & 0 \\ \dfrac{wh_i}{m_t} & 0 \end{pmatrix}, \quad \mathbf{u}(t) = \begin{pmatrix} x(t) \\ \dot{x}(t) \end{pmatrix} \tag{38}$$

instead of (22). The Matlab code for the stability analysis is in given Appendix B.

Stability charts are determined for down-milling operations of radial depth of cut ratios $a/D = 1,\ 0.5,\ 0.1,\ 0.05$ with a two fluted miller ($N = 2$). The other parameters are: natural frequency $f_n = 922\,\text{Hz}$, the angular natural frequency is $\omega_n = 5793\,\text{rad/s}$, the relative damping is $\zeta = 0.011$, the modal mass is $m_t = 0.03993\,\text{kg}$, the tangential and the normal linearized cutting force coefficients are $K_t = 6 \times 10^8\,\text{N/m}^2$ and $K_n = 2 \times 10^8\,\text{N/m}^2$, similarly as in [11, 44]. The approximation parameter for the semi-discretization is $m = 40$. The computation time for determining one stability chart over a 400×200 sized grid of parameters was 22.1 min on a 2 GHz Pentium 4 processor.

The stability charts for various radial depth of cut ratios are shown in Figure 6. The stable, chatter-free domains are denoted by grey colour.

5. 2 DOF MILLING EQUATION

The governing equation of motion of a 2 DOF milling model with a symmetric tool reads [44]

$$\begin{pmatrix} \ddot{x}(t) \\ \ddot{y}(t) \end{pmatrix} + \begin{pmatrix} 2\zeta\omega_n & 0 \\ 0 & 2\zeta\omega_n \end{pmatrix} \begin{pmatrix} \dot{x}(t) \\ \dot{y}(t) \end{pmatrix} + \begin{pmatrix} \omega_n^2 + \dfrac{wh_{xx}(t)}{m_t} & \dfrac{wh_{xy}(t)}{m_t} \\ \dfrac{wh_{yx}(t)}{m_t} & \omega_n^2 + \dfrac{wh_{yy(t)}}{m_t} \end{pmatrix} \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$$

$$= \begin{pmatrix} \dfrac{wh_{xx}(t)}{m_t} & \dfrac{wh_{xy}(t)}{m_t} \\ \dfrac{wh_{yx}(t)}{m_t} & \dfrac{wh_{yy(t)}}{m_t} \end{pmatrix} \begin{pmatrix} x(t-\tau) \\ y(t-\tau) \end{pmatrix} \tag{39}$$

where the angular natural frequency ω_n, the relative damping ζ and the modal mass m_t of the tool are considered to be equal to both x and y directions corresponding to the symmetric tool assumption. All the parameters are the same as for Equation (32) of the 1 DOF model.

In Equation (39), $h_{xx}(t)$, $h_{xy}(t)$, $h_{yx}(t)$ and $h_{yy}(t)$ are four projections of the specific cutting force coefficient defined as

$$h_{xx}(t) = \sum_{j=1}^{N} g(\phi_j(t)) \sin(\phi_j(t))(K_t \cos(\phi_j(t)) + K_n \sin(\phi_j(t))) \tag{40}$$

$$h_{xy}(t) = \sum_{j=1}^{N} g(\phi_j(t)) \cos(\phi_j(t))(K_t \cos(\phi_j(t)) + K_n \sin(\phi_j(t))) \tag{41}$$

$$h_{yx}(t) = \sum_{j=1}^{N} g(\phi_j(t)) \sin(\phi_j(t))(-K_t \sin(\phi_j(t)) + K_n \cos(\phi_j(t))) \tag{42}$$

$$h_{yy}(t) = \sum_{j=1}^{N} g(\phi_j(t)) \cos(\phi_j(t))(-K_t \sin(\phi_j(t)) + K_n \cos(\phi_j(t))) \tag{43}$$

Note, that the $h_{xx}(t)$ in Equation (40) is identical to $h(t)$ in Equation (33) of the 1 DOF case. The angular position $\phi_j(t)$ of tooth j and the function $g(\phi_j(t))$ are defined by Equations (34) and (35), respectively.

In the ith semi-discretization interval, Equation (39) can be approximated as

$$\begin{pmatrix} \ddot{x}(t) \\ \ddot{y}(t) \end{pmatrix} + \begin{pmatrix} 2\zeta\omega_n & 0 \\ 0 & 2\zeta\omega_n \end{pmatrix} \begin{pmatrix} \dot{x}(t) \\ \dot{y}(t) \end{pmatrix} + \begin{pmatrix} \omega_n^2 + \dfrac{wh_{xxi}}{m_t} & \dfrac{wh_{xyi}}{m_t} \\ \dfrac{wh_{yxi}}{m_t} & \omega_n^2 + \dfrac{wh_{yyi}}{m_t} \end{pmatrix} \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$$

$$= \begin{pmatrix} \dfrac{wh_{xxi}}{m_t} & \dfrac{wh_{xyi}}{m_t} \\ \dfrac{wh_{yxi}}{m_t} & \dfrac{wh_{yyi}}{m_t} \end{pmatrix} \begin{pmatrix} x_{\tau,i} \\ y_{\tau,i} \end{pmatrix} \tag{44}$$

By Cauchy transformation, Equation (44) is written in the form

$$\dot{\mathbf{u}}(t) = \mathbf{A}_i \mathbf{u}(t) + w_a \mathbf{B}_i \mathbf{u}_{i-m+1} + w_b \mathbf{B} \mathbf{u}_{i-m} \tag{45}$$

where

$$\mathbf{A}_i = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\omega_n^2 - \dfrac{wh_{xxi}}{m_t} & -\dfrac{wh_{xyi}}{m_t} & -2\zeta\omega_n & 0 \\ -\dfrac{wh_{yxi}}{m_t} & -\omega_n^2 - \dfrac{wh_{yyi}}{m_t} & 0 & -2\zeta\omega_n \end{pmatrix} \tag{46}$$

$$\mathbf{B}_i = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \dfrac{wh_{xxi}}{m_t} & \dfrac{wh_{xyi}}{m_t} & 0 & 0 \\ \dfrac{wh_{yxi}}{m_t} & \dfrac{wh_{yyi}}{m_t} & 0 & 0 \end{pmatrix} \tag{47}$$

$$\mathbf{u}(t) = \begin{pmatrix} x(t) \\ y(t) \\ \dot{x}(t) \\ \dot{y}(t) \end{pmatrix} \quad \text{and} \quad \mathbf{u}_j = \mathbf{u}(t_j) = \begin{pmatrix} x(t_j) \\ y(t_j) \\ \dot{x}(t_j) \\ \dot{y}(t_j) \end{pmatrix} = \begin{pmatrix} x_j \\ y_j \\ \dot{x}_j \\ \dot{y}_j \end{pmatrix} \tag{48}$$

for any integer j, and $w_a = w_b = \frac{1}{2}$, since the time delay is equal to the time period.

For the initial condition $\mathbf{u}(t_i) = \mathbf{u}_i$, $\mathbf{u}_{i+1}$ is determined as (similarly to Equation (26))

$$\mathbf{u}_{i+1} = \mathbf{P}_i \mathbf{u}_i + w_a \mathbf{R}_i \mathbf{u}_{i-m+1} + w_b \mathbf{R}_i \mathbf{u}_{i-m} \tag{49}$$

where

$$\mathbf{P}_i = \exp(\mathbf{A}_i \Delta t)$$

$$\mathbf{R}_i = (\exp(\mathbf{A}_i \Delta t) - \mathbf{I})\mathbf{A}_i^{-1} \mathbf{B}_i$$

Note, that $\dot{x}(t-\tau)$ and $\dot{y}(t-\tau)$ does not appear in Equation (39). Consequently, $\mathbf{u}_{i+1}$ depends on x_i, y_i, $\dot{x}_i$, $\dot{y}_i$, x_{i-m+1}, y_{i-m+1}, x_{i-m} and y_{i-m}, but it does not depend on $\dot{x}_{i-m+1}$, $\dot{y}_{i-m+1}$, $\dot{x}_{i-m}$ and $\dot{y}_{i-m}$. Therefore, the 3rd and the 4th columns of matrices $\mathbf{B}_i$ and $\mathbf{R}_i$ are zeros. This leads to the $(2m+4)$-dimensional state vector

$$\mathbf{z}_i = \mathrm{col}(x_i \;\; y_i \;\; \dot{x}_i \;\; \dot{y}_i \;\; x_{i-1} \;\; y_{i-1} \;\; \ldots \;\; x_{i-m} \;\; y_{i-m}) \tag{50}$$

instead of the $(4m+4)$-dimensional vector defined by Equation (13).

The resulted discrete map reads

$$\mathbf{z}_{i+1} = \mathbf{D}_i \mathbf{z}_i \tag{51}$$

where the $(2m+4)$-dimensional coefficient matrix is

$$\mathbf{D}_i = \begin{pmatrix} P_{i,11} & P_{i,12} & P_{i,13} & P_{i,14} & 0 & \dots & 0 & w_a R_{i,11} & w_a R_{i,12} & w_b R_{i,11} & w_b R_{i,12} \\ P_{i,21} & P_{i,22} & P_{i,23} & P_{i,24} & 0 & \dots & 0 & w_a R_{i,21} & w_a R_{i,22} & w_b R_{i,21} & w_b R_{i,22} \\ P_{i,31} & P_{i,32} & P_{i,33} & P_{i,34} & 0 & \dots & 0 & w_a R_{i,31} & w_a R_{i,32} & w_b R_{i,31} & w_b R_{i,32} \\ P_{i,41} & P_{i,42} & P_{i,43} & P_{i,44} & 0 & \dots & 0 & w_a R_{i,41} & w_a R_{i,42} & w_b R_{i,41} & w_b R_{i,42} \\ 1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \dots & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \tag{52}$$

Here, $P_{i,hj}$ and $R_{i,hj}$ are the elements of matrices $\mathbf{P}_i$ and $\mathbf{R}_i$ in the hth row and in the jth column, respectively.

The $(2m+4)$-dimensional transition matrix $\mathbf{\Phi}$ is determined now by coupling Equation (51) for $i = 0, 1, \dots, k-1$:

$$\mathbf{\Phi} = \mathbf{D}_{k-1}\mathbf{D}_{k-2}\dots\mathbf{D}_1\mathbf{D}_0 \tag{53}$$

If the eigenvalues of $\mathbf{\Phi}$ are in modulus less than one, then the system is stable. The Matlab code for the stability analysis is in Appendix C.

Stability charts are determined for down-milling operations of radial depth of cut ratios $a/D = 1, 0.5, 0.1, 0.05$ with a two fluted miller ($N = 2$). A symmetric tool was assumed with the same parameters as for the 1 DOF case: $f_n = 922$ Hz, $\omega_n = 5793$ rad/s, $\zeta = 0.011$, $m_t = 0.03993$ kg, $K_t = 6 \times 10^8$ N/m^2 and $K_n = 2 \times 10^8$ N/m^2, similarly as in References [11, 44]. The stability charts for various radial depth of cut ratios are shown in Figure 7. The stable, chatter-free domains are denoted by grey colour. The approximation parameter for the semi-discretization is $m = 40$. The computation time for determining one stability chart over a 400×200 sized grid of parameters was 39.3 min on a 2 GHz Pentium 4 processor.

These stability charts with the same parameters were also determined in Reference [44] by using time finite element method [13] and time-marching simulation [11]. The charts in Figure 7 are in good agreement with those presented in Reference [44].

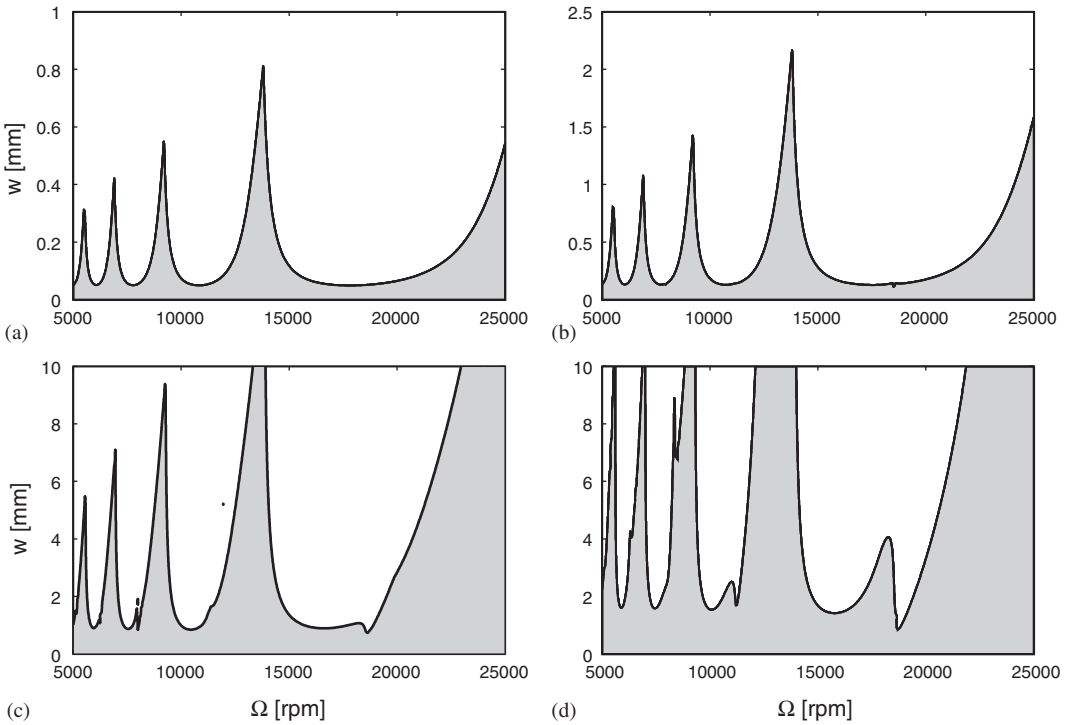

Figure 7. Stability charts for 2 DOF down-milling model with radial depth of cut ratio $a/D = 1$ (a), $a/D = 0.5$ (b), $a/D = 0.1$ (c), $a/D = 0.05$ (d).

6. CONCLUSIONS

An updated version of the semi-discretization method [34, 37], was presented for time periodic DDEs with optional ratio of the time delay and the time period. The problem in the method presented in Reference [34] was that the time period was not an integer multiple of the time step, and only the square of the Floquet transition matrix was determined that resulted double computation time and problems of identifying the critical characteristic roots.

The point of the updated semi-discretization method is that the delayed term is approximated as a weighted sum of the two neighbouring discrete delayed state values. With this technique, the time step is chosen as an integer fraction of the time period, and the Floquet transition matrix is determined over a single period.

Stability charts were determined for the damped and delayed Mathieu equation (17) for time-period/time-delay ratios $T/\tau = \frac{1}{2}$, $T/\tau = \frac{1}{\sqrt{2}}$, $T/\tau = 1$, $T/\tau = \sqrt{2}$ and $T/\tau = 2$. The case $T/\tau = 1$ was considered as a reference case, since for this case, the stability charts are known exactly and are comparable to the results of other publications [33–35]. For the cases $T/\tau \neq 1$, the structure of the charts were explained by referring to the corresponding autonomous and undelayed cases of the Hsu–Bhatt–Vyshnegradskii and the Strutt–Ince stability charts.

Stability charts were determined for 1 and 2 DOF milling processes. The charts for the 2 DOF case was compared to charts obtained by different methods [44].

Matlab codes for the stability analysis are given in the appendices in order to provide a full understanding of the method.

APPENDIX A: MATLAB CODE FOR THE DELAYED MATHIEU EQUATION

```
clear
% parameters
epsilon = 1;                                  % amplitude
kappa = 0;                                    % damping
stx = 200;                                    % steps of δ
sty = 100;                                    % steps of b
b_st = −1.5;                                  % starting value for b
b_fi = 1.5;                                   % final value for b
delta_st = −1;                                % starting value for δ
delta_fi = 5;                                 % final value for δ
% computational parameters
k = 40;                          % number of discretization intervals over one period
int k = 20;                      % number of integration interval in each step
T = 2*pi;                                     % time period
dt = T/k;                                     % discretization interval length
ddt = dt/int k;                               % integration interval length
tau = 2*pi;                                   % time delay
m = floor((tau + dt/2)/dt);
wa = (m*dt + dt/2 − tau)/dt;
wb = (tau + dt/2 − m*dt)/dt;
D = zeros(m + 2, m + 2);                      % matrix D
d = ones(m + 1, 1);
d(1 : 2) = 0;
D = D + diag(d, −1);
D(3, 1) = 1;
for i = 1 : k
    c(i) = sum(cos(((i − 1)*2*pi/k) : ddt : ((i − 1)*2*pi/k + ddt*(int k − 1))))/int k;
end                                           % discrete values of c_i
% start of computation
for y = 1 : sty + 1                           % loop for b
    b = b_st + (y − 1)*(b_fi − b_st)/sty;
    for x = 1 : stx + 1                       % loop for δ
        delta = delta_st + (x − 1)*(delta_fi − delta_st)/stx;
        Fi = eye(m + 2, m + 2);
% construct transition matrix Fi
        for i = 1 : k
            A = zeros(2, 2);                  % matrix A_i
            A(1, 2) = 1;
            A(2, 1) = −delta − epsilon*c(i);
            A(2, 2) = −kappa;
```



```
        B = zeros(2, 2);                          % matrix B
        B(2, 1) = b;
        P = expm(A*dt);                           % matrix P_i
        R = (expm(A*dt) − eye(2))*inv(A)*B;       % matrix R_i
        D(1 : 2, 1 : 2) = P;
        D(1 : 2, m + 1) = wa*R(1 : 2, 1 : 1);
        D(1 : 2, m + 2) = wb*R(1 : 2, 1 : 1);
        Fi = D*Fi;                                % Floquet transition matrix Φ
    end
    delta_m(x, y) = delta;                        % matrix of δ
    b_m(x, y) = b;                                % matrix of b
    ei_m(x, y) = max(abs(eig(Fi)));               % matrix of eigenvalues
  end
  sty + 1 − y                                     % counter
end
figure
contour(delta_m,b_m,ei_m,[1, 1],'k')
```

APPENDIX B: MATLAB CODE FOR THE 1 DOF MILLING ANALYSIS

```
clear
% parameters
N = 2;                                            % number of teeth
Kt = 6e8;                                         % tangential cutting force coefficient (N/m^2)
Kn = 2e8;                                         % normal cutting force coefficient (N/m^2)
w0 = 922*2*pi;                                    % angular natural frequency (rad/s)
zeta = 0.011;                                     % relative damping (1)
m_t = 0.03993;                                    % mass (kg)
aD = 0.05;                                        % radial depth of cut
up_or_down = −1;                                  % 1: up-milling, −1: down-milling
if up_or_down == 1                                % up-milling
    fist = 0;                                     % start angle
    fiex = acos(1 − 2*aD);                        % exit angle
elseif up_or_down == −1                           % down-milling
    fist = acos(2*aD − 1);                        % start angle
    fiex = pi;                                    % exit angle
end
stx = 400;                                        % steps of spindle speed
sty = 200;                                        % steps of depth of cut
w_st = 0e − 3;                                    % starting depth of cut (m)
w_fi = 10e − 3;                                   % final depth of cut (m)
o_st = 5e3;                                       % starting spindle speed (rpm)
```



```
o_fi = 25e3;                                  % final spindle speed (rpm)
% computational parameters
k = 40;                                       % number of discretization interval over one
                                                period
int k = 20;                                   % number of numerical integration steps for
                                                Equation (37)
m = k;                                        % since time delay = time period
wa = 1/2;                                     % since time delay = time period
wb = 1/2;                                     % since time delay = time period
D = zeros(m + 2, m + 2);                      % matrix D
d = ones(m + 1, 1);
d(1 : 2) = 0;
D = D + diag(d, −1);
D(3, 1) = 1;
% numerical integration of specific cutting force coefficient according to Equation (37)
for i = 1 : k
    dtr = 2*pi/N/k;                           % Δt, if τ = 2π/N
    h_i(i) = 0;
    for j = 1 : N                             % loop for tooth j
        for h = 1 : int k                     % loop for numerical integration of h_i
            fi(h) = i*dtr + ( j − 1)*2*pi/N + h*dtr/int k;
            if (fi(h) >= fist)*(fi(h) <= fiex)
                g(h) = 1;                     % tooth is in the cut
            else
                g(h) = 0;                     % tooth is out of cut
            end
        end
        h_i(i) = h_i(i) + sum(g.*(Kt.* cos(fi) + Kn.* sin(fi)).* sin(fi))/int k;
    end
end
% start of computation
for x = 1 : stx + 1                           % loop for spindle speeds
    o = o_st + (x − 1)*(o_fi − o_st)/stx;     % spindle speed
    tau = 60/o/N;                             % time delay
    dt = tau/(m);                             % time step
    for y = 1 : sty + 1                       % loop for depth of cuts
        w = w_st + (y − 1)*(w_fi − w_st)/sty; % depth of cut
% construct transition matrix Fi
        Fi = eye(m + 2, m + 2);
        for i = 1 : m
            A = zeros(2, 2);                  % matrix A_i
            A(1, 2) = 1;
            A(2, 1) = −w0^2 − h_i(i)*w/m_t;
            A(2, 2) = −2*zeta*w0;
            B = zeros(2, 2);                  % matrix B_i
```



```
            B(2, 1) = h_i(i)*w/m_t;
            P = expm(A*dt);                          % matrix P_i
            R = (expm(A*dt) − eye(2))*inv(A)*B;      % matrix R_i
            D(1 : 2, 1 : 2) = P;
            D(1 : 2, m + 1) = wa*R(1 : 2, 1 : 1);
            D(1 : 2, m + 2) = wb*R(1 : 2, 1 : 1);
            Fi = D*Fi;                               % transition matrix Φ
        end
        ss(x, y) = o;                                % matrix of spindle speeds
        dc(x, y) = w;                                % matrix of depth of cuts
        ei(x, y) = max(abs(eig(Fi)));                % matrix of eigenvalues
    end
    stx + 1 − x
end
figure
contour(ss,dc,ei,[1, 1],'k')
```

APPENDIX C: MATLAB CODE FOR THE 2 DOF MILLING ANALYSIS

```
clear
% parameters
N = 2;                                   % number of teeth
Kt = 6e8;                                % tangential cutting force coefficient
                                           (N/m^2)
Kn = 2e8;                                % normal cutting force coefficient
                                           (N/m^2)
w0x = 922*2*pi;                          % angular natural frequency x (rad/s)
zetax = 0.011;                           % relative damping x (1)
w0y = 922*2*pi;                          % angular natural frequency y (rad/s)
zetay = 0.011;                           % relative damping y (1)
m_tx = 0.03993;                          % mass x (kg)
m_ty = 0.03993;                          % mass y (kg)
aD = 0.05;                               % radial depth of cut
up_or_down = −1;                         % 1: up-milling, −1: down-milling
if up_or_down == 1                       % up-milling
    fist = 0;                            % start angle
    fiex = acos(1 − 2*aD);               % exit anlge
elseif up_or_down == −1                  % down-milling
    fist = acos(2*aD − 1);               % start angle
    fiex = pi;                           % exit angle
end
stx = 400;                               % steps of spindle speed
sty = 200;                               % steps of depth of cut
w_st = 0e − 3;                           % starting depth of cut (m)
```



```
w_fi = 10e − 3;                                  % final depth of cut (m)
o_st = 5e3;                                      % starting spindle speed (rpm)
o_fi = 25e3;                                     % final spindle speed (rpm)
% computational parameters
k = 40;                                          % number of discretization interval over one
                                                   period
int k = 20;                                      % number of numerical integration steps for
                                                   Equation (37)
m = k;                                           % since time delay = time period
wa = 1/2;                                        % since time delay = time period
wb = 1/2;                                        % since time delay = time period
D = zeros(2*m + 4, 2*m + 4);                     % matrix D
d = ones(2*m + 2, 1);
d(1 : 4) = 0;
D = D + diag(d, −2);
D(5, 1) = 1;
D(6, 2) = 1;
% numerical integration of specific cutting force coefficient according to Equations (40)–(43)
for i = 1 : k
    dtr = 2*pi/N/k;                              % Δt, if τ = 2π/N
    hxx(i) = 0;
    hxy(i) = 0;
    hyx(i) = 0;
    hyy(i) = 0;
    for j = 1 : N                                % loop for tooth j
        for h = 1 : int k                        % loop for numerical integration of h_i
            fi(h) = i*dtr + ( j − 1)*2*pi/N + h*dtr/int k;
            if (fi(h) >= fist)*(fi(h) <= fiex)
                g(h) = 1;                        % tooth is in the cut
            else
                g(h) = 0;                        % tooth is out of cut
            end
        end
        hxx(i) = hxx(i) + sum(g.*(Kt.* cos(fi) + Kn.* sin(fi)).* sin(fi))/intk;
        hxy(i) = hxy(i) + sum(g.*(Kt.* cos(fi) + Kn.* sin(fi)).* cos(fi))/int k;
        hyx(i) = hyx(i) + sum(g.*(−Kt* sin(fi) + Kn.* cos(fi)).* sin(fi))/int k;
        hyy(i) = hyy(i) + sum(g.*(−Kt* sin(fi) + Kn.* cos(fi)).* cos(fi))/int k;
    end
end
% start of computation
for x = 1 : stx + 1                              % loop for spindle speeds
    o = o_st + (x − 1)*(o_fi − o_st)/stx;        % spindle speed
    tau = 60/o/N;                                % time delay
    dt = tau/(m);                                % time step
    for y = 1 : sty + 1                          % loop for depth of cuts
```



```
        w = w_st + (y − 1)*(w_fi − w_st)/sty;          % depth of cut
% construct transition matrix Fi
        Fi = eye(2*m + 4, 2*m + 4);
        for i = 1 : m
            A = zeros(4, 4);                           % matrix A_i
            A(1, 3) = 1;
            A(2, 4) = 1;
            A(3, 1) = −w0x^2 − hxx(i)*w/m_tx;
            A(3, 2) = −hxy(i)*w/m_tx;
            A(3, 3) = −2*zetax*w0x;
            A(4, 1) = −hyx(i)*w/m_ty;
            A(4, 2) = −w0y^2 − hyy(i)*w/m_ty;
            A(4, 4) = −2*zetay*w0y;
            B = zeros(4, 4);                           % matrix B_i
            B(3, 1) = hxx(i)*w/m_tx;
            B(3, 2) = hxy(i)*w/m_tx;
            B(4, 1) = hyx(i)*w/m_ty;
            B(4, 2) = hyy(i)*w/m_ty;
            P = expm(A*dt);                            % matrix P_i
            R = (expm(A*dt) − eye(4))*inv(A)*B;        % matrix R_i
            D(1 : 4, 1 : 4) = P;
            D(1 : 4, (2*m + 1) : (2*m + 2)) = wa*R(1 : 4, 1 : 2);
            D(1 : 4, (2*m + 3) : (2*m + 4)) = wb*R(1 : 4, 1 : 2);
            Fi = D*Fi;                                 % transition matrix Φ
        end
        ss(x, y) = o;                                  % matrix of spindle speeds
        dc(x, y) = w;                                  % matrix of depth of cuts
        ei(x, y) = max(abs(eig(Fi)));                  % matrix of eigenvalues
    end
    stx + 1 − x
end
figure
contour(ss,dc,ei,[1, 1],'k')
```

ACKNOWLEDGEMENTS

This research was supported in part by the Magyary Zoltán Postdoctoral Fellowship of Foundation for Hungarian Higher Education and Research and by the Hungarian National Science Foundation under Grant No. OTKA F047318 and OTKA T043368.

REFERENCES

1. Volterra V. Sur la Theorie Mathematique des Phenomenes Hereditares. *Journal de Mathématiques Pures et Appliqués* 1928; **7**:149–192.
2. von Schlippe B, Dietrich R. Shimmying of a pneumatic wheel, Lilienthal-Gesellschaft für Luftfahrtforschung. *Bericht* 1941; **140**:125–160 (translated for the *AAF* in 1947 by Meyer & Company).

3. Minorsky N. Selfexcited oscillations in dynamical systems possessing retarded actions. *Journal of Applied Mechanics* 1942; **9**:65–71.
4. Tlusty J, Polacek A, Danek C, Spacek J. *Selbsterregte Schwingungen an Werkzeugmaschinen*. VEB Verlag Technik: Berlin, 1962.
5. Tobias S.A. *Machine Tool Vibration*. Blackie: London, 1965.
6. Altintas Y, Engin S, Budak E. Analytical stability prediction and design of variable pitch cutters. *Journal of Manufacturing Science and Engineering* 1999; **121**:173–178.
7. Seagalman DJ, Butcher EA. Suppression of regenerative chatter via impedance modulation. *Journal of Vibration and Control* 2000; **6**:243–256.
8. Stépán G. Modelling non-linear regenerative effects in metal cutting. *Philosophical Transactions of the Royal Society* 2001; **359**:739–757.
9. Zhao MX, Balachandran B. Dynamics and stability of milling process. *International Journal of Solids and Structures* 2001; **38**(10–13):2233–2248.
10. Gouskov AM, Voronov SA, Paris H, Batzer SA. Nonlinear dynamics of a machining system with two interdependent delays. *Communications in Nonlinear Science and Numerical Simulation* 2002; **7**(4):207–221.
11. Schmitz T, Medicus K, Dutterer B. Exploring once-per-revolution audio signal variance as a chatter indicator. *Machining Science and Technology* 2002; **6**(2):207–225.
12. Insperger T, Stépán G, Bayly PV, Mann BP. Multiple chatter frequencies in milling processes. *Journal of Sound and Vibration* 2003; **262**(2):333–345.
13. Bayly PV, Halley JE, Mann BP, Davies MA. Stability of interrupted cutting by temporal finite element analysis. *Journal of Manufacturing Science and Engineering* 2003; **125**(2):220–225.
14. Szalai R, Stépán G. Stability boundaries of high-speed milling corresponding to period doubling are essentially closed curves. *Proceedings of ASME International Mechanical Engineering Conference and Exposition*, Washington, D.C., 2003, paper no. IMECE2003-42122 (CD-ROM).
15. Craig JJ. *Introduction to Robot Mechanics and Control*. Addison-Wesley: Reading, MA, 1986.
16. Insperger T, Stépán G. Stability improvements of robot control by periodic variation of the gain parameters. In *Proceedings of the 11th World Congress in Mechanism and Machine Science*, Tian Huang (ed.), Tianjin, China, 2003. China Machinery Press: China, 2004; 1816–1820.
17. Campbell SA, Ruan S, Wei J. Qualitative analysis of a neural network model with multiple time delays. *International Journal of Bifurcation and Chaos* 1999; **9**(8):1585–1595.
18. Foss J, Moss F, Milton J. Noise, multistability, and delayed recurrent loops. *Physical Review E* 1997; **55**(4):4536–4543.
19. Stépán G, Kollár LE. Balancing with reflex delay. *Mathematical and Computer Modelling* 2000; **31**:199–205.
20. Kolmanovskii VB, Nosov VR. *Stability of Functional Differential Equations*. Academic Press: London, 1986.
21. Stépán G. *Retarded Dynamical Systems*. Longman: Harlow, 1989.
22. Hale JK, Lunel SMV. *Introduction to Functional Differential Equations*. Springer: New York, 1993.
23. Diekmann O, van Gils SA, Lunel SMV, Walther H-O. *Delay Equations*. Springer: New York, 1995.
24. Olgac N, Sipahi R. An exact method for the stability analysis of time delayed LTI systems. *IEEE Transactions on Automatic Control* 2002; **47**(5):793–797.
25. Farkas M. *Periodic Motions*. Springer: New York, 1994.
26. Simmendinger C, Pelster A, Wunderlin A. Analytical approach for the Floquet theory of delay differential equations. *Physical Review E* 1999; **59**:5344–5353.
27. Engelborghs K, Luzyanina T, in 't Hout KJ, Roose D. Collocation methods for the computation of periodic solutions of delay differential equations. *SIAM Journal on Scientific Computing* 2000; **22**(5):1593–1609.
28. Gilsinn DE. Approximating limit cycles of a Van der Pol equation with delay. *Proceedings of Dynamical Systems and Applications*, Atlanta, GA, vol. 4, 2004.
29. Gilsinn DE. Estimating critical Hopf bifurcation parameters for a second-order delay differential equation with application to machine tool chatter. *Nonlinear Dynamics* 2002; **30**(2):103–154.
30. Sexton JS, Milne RD, Stone BJ. A stability analysis of single point machining with varying spindle speed. *Applied Mathematical Modelling* 1977; **1**:310–318.
31. Jayaram S, Kapoor SG, DeVor RE. Analytical stability analysis of variable spindle speed machining. *Journal of Manufacturing Science and Engineering* 2000; **122**(3):391–397.
32. Insperger T, Schmitz TL, Burns TJ, Stépán G. Comparison of analytical and numerical simulations for variable spindle speed turning. *Proceedings of ASME International Mechanical Engineering Conference and Exposition*, Washington, D.C., 2003, paper no. IMECE2003-41809 (CD-ROM).

33. Insperger T, Stépán G. Stability chart for the delayed Mathieu equation. *Proceedings of The Royal Society, Mathematical Physical and Engineering Sciences* 2002; **458**(2024):1989–1998.
34. Insperger T, Stépán G. Stability of the damped Mathieu equation with time delay. *Journal of Dynamic System, Measurement and Control* 2003; **125**(2):166–171.
35. Butcher EA, Ma H, Bueler E, Averina V, Szabó Zs. Stability of time-periodic delay-differential equations via Chebyshev polynomials. *International Journal for Numerical Methods in Engineering* 2004; **59**(7):895–922.
36. Sinha SC, Wu DH. An efficient computational scheme for the analysis of periodic systems. *Journal of Sound and Vibration* 1991; **151**:91–117.
37. Insperger T, Stépán G. Semi-discretization method for delayed systems. *International Journal for Numerical Methods in Engineering* 2002; **55**(5):503–518.
38. Győri I, Hartung F, Turi J. Numerical approximations for a class of differential equations with time- and state-dependent delays. *Applied Mathematics Letters* 1995; **8**(6):19–24.
39. Elbeyli O, Sun JQ, Ünal G. A semi-discretization method for delayed stochastic systems. *Communication in Nonlinear Science and Numerical Simulation*, in press.
40. Hsu CS, Bhatt SJ. Stability charts for second-order dynamical systems with time lag. *Journal of Applied Mechanics* 1966; **33E**(1):119–124.
41. van der Pol F, Strutt MJO. On the stability of the solutions of Mathieu's equation. *Philosophical Magazine, and Journal of Science* 1928; **5**:18–38.
42. Ince EL. *Ordinary Differential Equations*. Dover Publications, Inc.: New York, 1956.
43. Lakshmikantham V, Trigiante D. *Theory of Difference Equations, Numerical Methods and Applications*. Academic Press: London, 1988.
44. Bayly PV, Mann BP, Schmitz TL, Peters DA, Stépán G, Insperger T. Effects of radial immersion and cutting direction on chatter instability in end-milling. *Proceedings of ASME International Mechanical Engineering Conference and Exposition, New Orleans, LA*, 2002, paper no. IMECE2002-34116 (CD-ROM).

