Data-Driven Parameter Estimation of Time Delay Dynamical Systems for Stability Prediction

by

James D. Turner

Department of Mechanical Engineering and Materials Science
Duke University

Date: _____

Approved:

_____
Brian P. Mann, Supervisor

_____
Lawrence N. Virgin

_____
Samuel C. Stanton

_____
Genevieve M. Lipp

Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Mechanical Engineering and Materials Science
in the Graduate School of Duke University
2021

Abstract

Data-Driven Parameter Estimation of Time Delay Dynamical Systems for Stability Prediction

by

James D. Turner

Department of Mechanical Engineering and Materials Science
Duke University

Date: _____

Approved:

_____
Brian P. Mann, Supervisor

_____
Lawrence N. Virgin

_____
Samuel C. Stanton

_____
Genevieve M. Lipp

An abstract of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Mechanical Engineering and Materials Science
in the Graduate School of Duke University
2021

Abstract

Subtractive machining operations such as milling, turning, and drilling are an essential part of many manufacturing processes. Unfortunately, under certain combinations of machine settings, the motion of the cutting tool can become unstable, due to feedback between consecutive passes of the tool. This phenomenon is known as chatter. Mathematical models, specifically delay differential equations (DDEs), can describe the motion of the cutting tool and predict this instability. While these models are useful, estimates of the models' parameters are necessary in order to apply them to real systems. Unfortunately, estimating the parameters directly can be time-consuming, expensive, and difficult. The objective of this research is to develop automated methods to estimate these parameters indirectly, from time series measurements of the tool's motion which can be collected in a few minutes with sensors attached to the machine. The estimated parameters can then be used to predict when chatter will occur so that the machine operator can select appropriate settings.

One way to estimate the parameters of a dynamics model is to match the characteristic multipliers (CMs) predicted by the model to CMs estimated from time series data. CMs describe the behavior, such as stability, of a dynamical system near a limit cycle. While existing CM estimation methods are available, practical challenges such as measurement noise, limited time series length, and repeated CMs can substantially reduce their accuracy. The first part of this dissertation presents improved methods for estimating CMs from time series. Numerical validation stud-

ies demonstrate that the improved methods consistently provide more accurate CM estimates than existing methods in a variety of scenarios.

The second part of this dissertation introduces improvements to CM matching and trajectory matching methods for estimating the parameters of DDEs from noisy time series data. For CM matching, it incorporates the empirical CM estimation improvements from the previous part, and it introduces a way to match multiple CM estimates for each time series. For trajectory matching, it describes how to handle multivariate observations and prior knowledge in a principled way; it uses the spectral element method to provide a convenient representation of the initial interval and reduce the computational cost of computing the objective function; and it fits multiple time series simultaneously. Simulation results demonstrate that these improved methods work well in practice, although CM matching has some limitations which are not a problem for the trajectory matching method.

The final part of this dissertation introduces a new approach to estimate the parameters of a DDE model for milling from noisy time series data, based on the trajectory matching approach described in the previous part. It extends models from the literature to more closely fit the time series data, and it describes a procedure to estimate the unknown parameters in stages, without having to solve a global optimization algorithm for all the parameters simultaneously. Additionally, it adapts the spectral element method to make predictions for this model. Experimental results using time series data collected on an instrumented milling machine demonstrate that the model and fitting procedure successfully estimate parameters for which the predicted stability boundaries approximate the true stability boundaries.

Contents

List of Tables

List of Figures

List of Acronyms

CM characteristic multiplier. iv, v, 1–9, 11, 13–16, 18, 19, 21, 22, 25, 26, 30, 31, 34, 38, 41–43, 45–49, 54–60, 65, 72, 92, 99–102, 109, 112, 130

DDE delay differential equation. iv, v, 2–5, 39–43, 45, 49–51, 53, 58, 62, 63, 65, 71, 80, 90, 93, 99, 100, 102, 115, 116, 120

KLT Karhunen–Loève transform. 46

LGL Legendre–Gauss–Lobatto. 44, 126, 128

MAP maximum a posteriori. 115, 119

ODE ordinary differential equation. 2, 9, 40, 45, 50, 70, 80, 115, 116

OLS ordinary least squares. 8, 25, 81, 83

OSPA optimal subpattern assignment. 49

TFEA time finite element analysis. 2, 3

TLS total least squares. 8, 25

Acknowledgements

I owe my deepest gratitude to those who provided support during my Ph.D. studies. My success would not have been possible without them.

First, I thank my advisor, Dr. Brian Mann, for his mentorship. He has always treated me with respect and friendship, and his experience, advice, and support have been essential. I especially admire his enthusiasm for learning and teaching and his care for his students. I also thank the members of my Ph.D. committees—Dr. Samuel Stanton, Dr. Lawrence Virgin, Dr. Genevieve Lipp, Dr. Robert Kielb, Dr. Andre Mazzoleni, and Dr. Kris Hauser—for their time and effort.

I especially appreciate my family's support. They have always been a stalwart source of strength and encouragement.

Many mentors have provided guidance throughout my research career, especially Dr. Ademola Ejire, Dr. James McMahon, Dr. Yong Zhu, Dr. Timothy Shafer, Dr. Michael Zavlanos, Dr. Jeffrey Eischen, and Dr. Lianne Cartee.

Additionally, I had the pleasure of collaborating and socializing with many great colleagues and friends at Duke. Among many others, I thank Dane Sequeira, Jared Little, Levi Manring, Xueshe Wang, and Michael Mazzoleni.

Finally, I thank the following sources of funding: the National Defense Science and Engineering Graduate Fellowship Program, the National Science Foundation, the Army Research Office, the U.S. Naval Research Laboratory, Duke University, and the American Society of Mechanical Engineers.

1

Introduction

Subtractive machining operations such as milling, turning, and drilling are an essential part of many manufacturing processes. Under certain combinations of machine settings, the motion of the cutting tool can become unstable, due to feedback between consecutive passes of the tool. This phenomenon is known as chatter. Mathematical models can describe the motion of the cutting tool and predict this instability. While these models are useful, estimates of the models' parameters are necessary in order to apply them to real systems. Unfortunately, estimating the parameters directly can be time-consuming, expensive, and difficult. The objective of this research is to develop automated methods to estimate these parameters indirectly, via time series measurements of the tool's motion which can be collected in a few minutes with sensors attached to the machine. Towards this end, this dissertation presents improvements to empirical characteristic multiplier (CM) estimation techniques, improved methods for estimating parameters from time series measurements, and improved models and parameter estimation methods specifically for milling processes.

1.1 Background

CMs describe the asymptotic stability of dynamical systems, such as machining processes, about an equilibrium point or limit cycle oscillation. Given a model of a dynamical system, the stability can be predicted by computing the CMs from the model [1–11]; if the magnitude of any CM is greater than one, then the system will be unstable. Often, however, a model of the dynamics or values of the model's parameters may be unavailable. In these cases, CMs can be estimated from time series measurements of the system's state [11–19]. However, existing methods to estimate CMs from time series data have practical limitations. Improvements to these methods would enable CMs to be estimated more accurately despite noisy measurements. If CMs could be accurately estimated from time series, then parameter values of a dynamics model describing the system could be estimated by minimizing the difference between these CM estimates and the CMs predicted by the model [11].

Many machining processes can be described by delay differential equation (DDE) models [11, 20–27], where the time derivative of the system's state, such as the position and velocity of the cutting tool, is a function not only of the current time t and system state $\boldsymbol{x}(t)$ but also the system state $\boldsymbol{x}(t-\tau)$ at a fixed amount of time in the past:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left(t, \boldsymbol{x}(t), \boldsymbol{x}(t-\tau)\right) \tag{1.1}$$

For machining processes, the dependence on the delayed state $\boldsymbol{x}(t-\tau)$ is caused by the material removed on the previous pass of the cutting tool affecting the thickness of the chip removed on the current pass of the cutting tool, and therefore the cutting forces. This time-delayed feedback significantly complicates the analysis of DDEs compared to systems modeled by ordinary differential equations (ODEs).

A particularly useful class of methods for analyzing DDE models is semi-analytical methods based on discretizing time into elements. For example, time finite element

analysis (TFEA) has been used to predict the stability boundaries for milling systems [7, 8, 11]. A technique known as the spectral element method is similar but has faster convergence [10]. These techniques can be used to efficiently compute the CMs for DDEs and can also be adapted to efficiently compute trajectories described by DDE models. However, these analysis techniques require the values of the model parameters.

A variety of techniques have been explored for estimating the parameters of DDE models from time series data. Most of this research has focused on autonomous DDEs and involves estimating $\dot{\boldsymbol{x}}(t)$, $\boldsymbol{x}(t)$, and $\boldsymbol{x}(t-\tau)$ from the data, so that the relationship between them can be recovered by regression or other statistical techniques [28–36]. Other variations on this technique also exist [37–39], but they have many of the same limitations—requiring measurements or accurate numerical estimates of $\dot{\boldsymbol{x}}(t)$, requiring measurements or a reconstruction of the full state, and being developed only for autonomous DDEs. Two other strategies are of particular interest, since they do not require estimates of $\dot{\boldsymbol{x}}(t)$ and are suitable for nonautonomous DDEs: matching CMs and matching trajectories. The CM matching method uses optimization methods to find parameters for which the CMs computed using TFEA most closely match the CMs estimated from time series [11]. The trajectory matching method uses optimization methods to estimate the parameters, and initial conditions if necessary, which cause the trajectory predicted by the dynamics model to most closely match the time series data [40–44]. Improvements to these techniques would enable more accurate parameter estimation on a wider variety of systems. The literature regarding parameter estimation from time series data is especially sparse for milling operations, so this topic is particularly ripe for further research.

1.2 Research Contributions

The primary objective of this work is to investigate improved techniques for estimating the parameters of DDE models from time series data, especially for milling. The research contributions of this work are in three primary areas: improved methods for estimating CMs from time series data, improved methods for estimating model parameters for DDEs in general, and improved methods for estimating model parameters for milling in particular. The improvements to CM estimation allow for CMs to be estimated more accurately in the presence of observation noise and other challenges, which make them more useful for CM matching. The investigation into improved methods for DDE parameter estimation considers both CM matching and trajectory matching. For CM matching, it describes how to match multiple CM estimates per time series, and it incorporates the improved CM estimation techniques introduced in this work. For trajectory matching, the approach introduced in this work handles multivarate observations and prior knowledge about parameters in a principled way, uses the spectral element method to provide a convenient representation of the initial interval and to reduce the computational cost of computing the objective function, and fits multiple time series simultaneously instead of only a single time series. The investigation into parameter estimation for milling in particular introduces a new method to estimate the parameters for a two-degree-of-freedom milling model, combines and extends existing models to better match experimental data, adapts the spectral element method for this model, and evaluates the proposed methods on experimental data. In the end, the proposed methods for milling allow the operator to, in the span of a few minutes, collect data for a few test cuts using sensors attached to the machine, run an automated algorithm to estimate the parameters, and then automatically generate a stability chart which can be used to select appropriate machining settings.

1.3 Organization

This work is organized into three chapters, corresponding to the three primary contribution areas, followed by appendices with supplemental information. Chapter 2 introduces improvements for empirical CM estimation and compares the accuracy of CMs computed using various methods. Chapter 3 introduces improved CM matching and trajectory matching techniques for estimating the parameters of DDEs from time series data; it incorporates the improved CM estimation techniques introduced in Chapter 2. Chapter 4 introduces a method for estimating the parameters of an improved DDE model for milling. At its core, the approach is based on the trajectory matching method introduced in Chapter 3. Finally, the appendices provide supplemental information, including background on Floquet theory (Appendix A), how to describe the trajectory matching problem as a least squares problem (Appendix C), a way to guess the initial conditions for trajectory matching (Appendix D), and adapting the spectral element method for the milling model used in this work (Appendix E).

2

Empirical Characteristic Multiplier Estimation: A Comparison of Methods and Improvements[1]

2.1 Introduction

Characteristic multipliers (CMs) describe the behavior of a dynamical system close to a time-periodic steady-state response. In particular, if the magnitudes of all the system's CMs are less than one, then the steady-state response is asymptotically stable. If an accurate model of the system's dynamics is available, the CMs can be computed numerically or sometimes analytically [1–3]. CMs can be computed for models with time delays, as well [4–6, 9]. However, an accurate model is often unavailable for real, physical systems, so it is useful to be able to estimate CMs from experimental data. Floquet theory provides the basis for computing CMs from a time series of the system's states following an initial perturbation from the steady-state response. Briefly, the transformation of the perturbation over one period can be described by multiplication with a time-periodic transition matrix, and the eigenvalues of this matrix are the CMs of the system. Existing methods to estimate CMs

[1] This work has been published in the Journal of Sound and Vibration [45].

from time series using Floquet theory work in ideal cases, but each has limitations when applied in practice.

Estimating characteristic multipliers from experimental data has been explored in a variety of settings, including machining [11], cardiac dynamics [12], human and canine gaits [13, 14], an aeroelastic system [15], electric power systems [16], a forced spring–pendulum [17], an impacting pendulum [18], a geometrically nonlinear oscillator [18], and a torsionally-excited pendulum [19]. This chapter considers only deterministic systems, although CMs can also describe systems with some noise in the dynamics [46]. CMs can be used not only as absolute stability criteria, but also to determine how close the system is to the stability boundary and to qualitatively analyze how the dynamics of the system change with changing parameters. Numerically accurate CMs may also be useful for system identification or assessing approximate models [11, 19].

A variety of experimental challenges can occur when collecting the data necessary to estimate CMs. For example, measurement noise is usually unavoidable. The length of the time series may be limited if the applied perturbations are not under the control of the experimenter, so it may be impossible to directly measure the steady-state response of the system. Finally, it may be difficult to measure all of the system's state variables. To address this last issue, a variety of state space reconstruction methods have been proposed to construct a time series representation of the system from measurements of only a subset of the state variables [47, 48]. This constructed time series can then be used with the various CM estimation methods [11, 12, 17, 19]. As will be demonstrated, several existing CM estimation methods have limitations primarily related to measurement noise and short time series.

The most common method for estimating CMs from time series data is the difference method. In order to estimate the perturbation over time, the system is measured once per period at a constant phase, the steady-state response at that phase is ap-

proximated by the last measurement, and the perturbation is approximated by the difference between the measurements and the approximated steady-state response. Then, this series of perturbation estimates is used to compute the transition matrix by ordinary least squares (OLS). This method works well if the measurement noise is small relative to the perturbation, the time series is long enough for the perturbation to die out, and the characteristic multipliers are distinct. However, these conditions can be difficult to achieve in practice.

An alternative approach is to fit an affine transition model to the periodic state measurements so that the steady-state response does not need to be estimated separately [13, 14, 18]. However, since this method still uses only a single state measurement per period, it performs poorly when the measurement noise is large relative to the perturbation.

To address the noise issue while avoiding directly estimating the steady-state response, Little et al. proposed using a moving average of the system's states instead of explicitly estimating the perturbation [19]. They showed that the averaging process preserves the CMs, and the integral substantially reduces the effect of noise when the number of samples per period is large. However, this approach relies on the moving average of the steady-state response being zero, which is sometimes invalid for physical systems.

This chapter describes new approaches which address the limitations of existing methods. First, it shows that the moving integral method can be modified by using an affine transition model to avoid constraining the steady-state response. Second, it illustrates how the OLS approach used by existing methods introduces bias and how total least squares (TLS) resolves this issue. Finally, it investigates the numerical issues caused by repeated characteristic multipliers and describes a workaround. Changes to the traditional methods can be combined in various ways, as illustrated in fig. 2.1. All of the improvements can be combined to obtain a method which

is insensitive to noise, does not make assumptions about the steady-state response, works with short time series, and can handle cases with repeated CMs. Simulation results for a variety of cases show that this combined method produces more accurate CM estimates than existing methods in all the tested cases.

2.2 Existing Methods and Limitations

Consider a dynamical system

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), t) \tag{2.1}$$

with a T-periodic equilibrium response $\boldsymbol{x}_{\mathrm{p}}(t) = \boldsymbol{x}_{\mathrm{p}}(t+T)$. The asymptotic stability of the system about the periodic response is described by the behavior of a small perturbation from that response. Let the perturbation from the periodic equilibrium response be $\boldsymbol{\xi}(t) = \boldsymbol{x}(t) - \boldsymbol{x}_{\mathrm{p}}(t)$. Then, it can be shown that, after linearizing about the periodic response, the dynamics of the perturbation are given by a linear ordinary differential equation (ODE) with time-periodic coefficients,

$$\dot{\boldsymbol{\xi}}(t) = \boldsymbol{A}(t)\boldsymbol{\xi}(t) \tag{2.2}$$

where $\boldsymbol{A}(t) = \boldsymbol{\nabla}_{\boldsymbol{x}}^{\top} \boldsymbol{f}(\boldsymbol{x}_{\mathrm{p}}(t), t)$ is a T-periodic matrix of coefficients. Then, it can be shown that the transition over one period is

$$\boldsymbol{\xi}(t+T) = \boldsymbol{P}(t)\boldsymbol{\xi}(t) \tag{2.3}$$

where the transition matrix $\boldsymbol{P}(t)$ is periodic, $\boldsymbol{P}(t+T) = \boldsymbol{P}(t)$, and its eigenvalues are constant. The eigenvalues of $\boldsymbol{P}(t)$ are known as the CMs. The CMs are a useful property of the system because they describe the decay or growth in the perturbation over time; the system is asymptotically stable about $\boldsymbol{x}_{\mathrm{p}}(t)$ only if the magnitudes of all the characteristic multipliers are less than 1. See Appendix A for details.

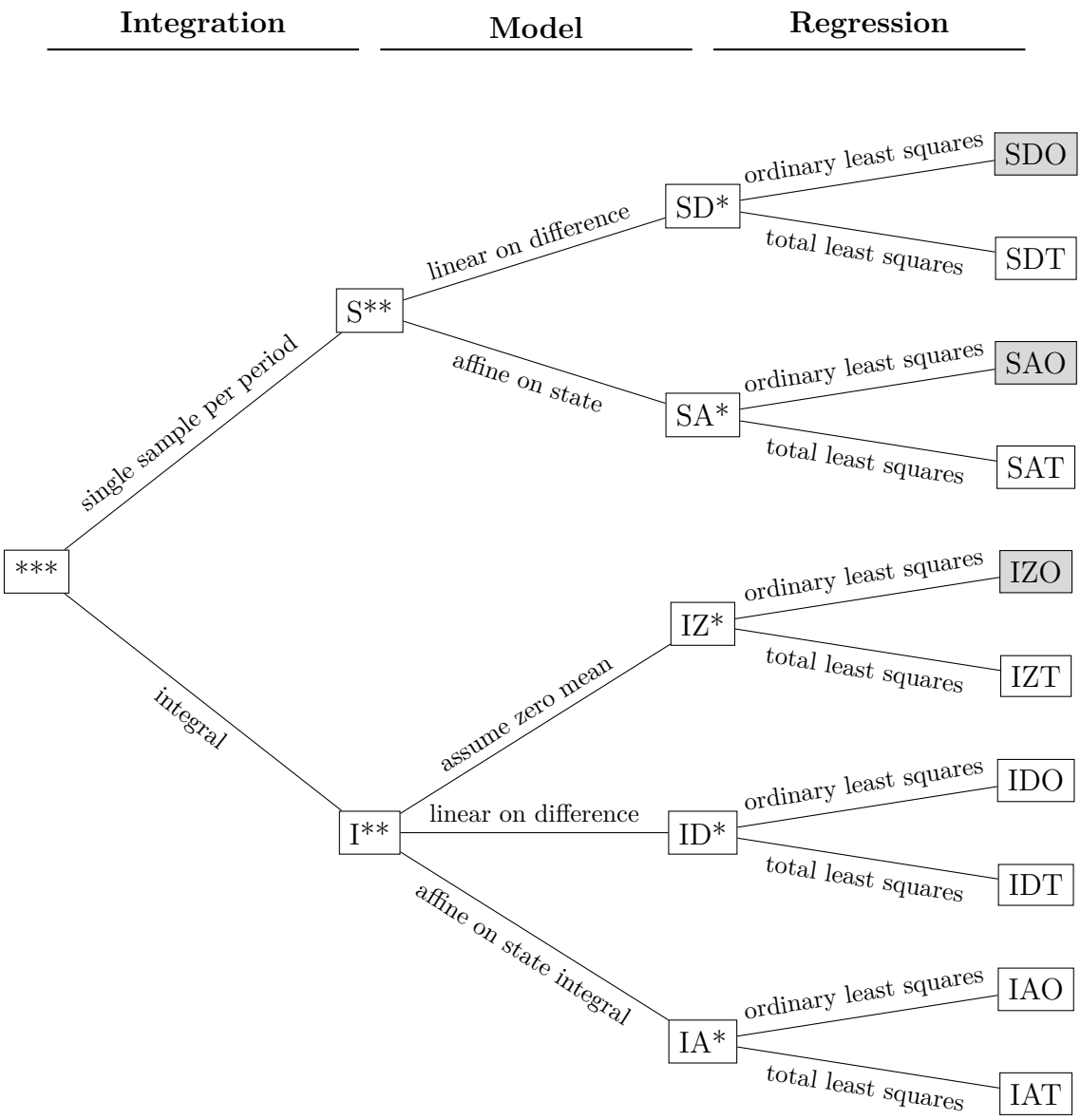


FIGURE 2.1: Classification of characteristic multiplier estimation methods for a single time series, and the corresponding acronyms used in this chapter. An asterisk represents any option; for example, "IA*" represents either method which integrates the raw data and uses the affine model, i.e. the IAO or IAT method. Whether or not multiple time series are used could be an additional level of the tree, but that is not shown here for the sake of conciseness. The methods in existing literature are indicated with a gray background.

2.2.1 Existing Methods

Since the characteristic multipliers are such an important descriptor of a system, it is useful to be able to estimate the CMs from experimental data. The CMs can be estimated from a time series of the system's response following an initial perturbation. Existing methods to estimate CMs this way work by estimating the transition matrix $\boldsymbol{P}(t_0)$ for a given phase t_0, or another analogous matrix, and then computing the eigenvalues of that matrix.

Difference Method with Ordinary Least Squares

The most commonly used method explicitly estimates the time series of the perturbation at a given phase by subtracting an estimate of the fixed point at that phase from the state measurements, i.e. $\boldsymbol{\xi}(t_0+jT) = \boldsymbol{x}(t_0+jT) - \boldsymbol{x}_{\mathrm{p}}(t_0)$. Then, this data is used to estimate the transition matrix at that phase, as follows. Let $\boldsymbol{\xi}_j = \boldsymbol{\xi}(t_0+jT)$, then by eq. (2.3),

$$\boldsymbol{\xi}_{j+1} = \boldsymbol{P}(t_0)\boldsymbol{\xi}_j \tag{2.4}$$

If m samples are stacked together into a single matrix, $\boldsymbol{\Xi}_j = \begin{bmatrix}\boldsymbol{\xi}_j & \boldsymbol{\xi}_{j+1} & \cdots & \boldsymbol{\xi}_{j+m-1}\end{bmatrix}$, then

$$\boldsymbol{\Xi}_1 = \boldsymbol{P}(t_0)\boldsymbol{\Xi}_0 \tag{2.5}$$

Then, eq. (2.5) is solved for $\boldsymbol{P}(t_0)$, and the eigenvalues of $\boldsymbol{P}(t_0)$ are the CM estimates. If the values of $\boldsymbol{\xi}_0, \ldots, \boldsymbol{\xi}_m$ were known perfectly, then it would be sufficient to use m equal to the number of state variables, and solve the equation as $\boldsymbol{P}(t_0) = \boldsymbol{\Xi}_1\boldsymbol{\Xi}_0^{-1}$. However, when conducting a physical experiment, only noisy estimates are available, so it is common to choose a large value for m and solve eq. (2.5) using ordinary least squares, $\boldsymbol{P}(t_0) = \boldsymbol{\Xi}_1\boldsymbol{\Xi}_0^{\top}\left(\boldsymbol{\Xi}_0\boldsymbol{\Xi}_0^{\top}\right)^{-1}$.

The primary issues with this method are estimating the fixed point $\boldsymbol{x}_{\mathrm{p}}(t_0)$ and dealing with measurement noise. Typically, the last available state measurement

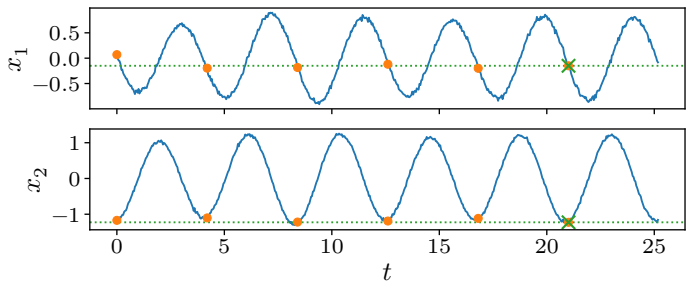


FIGURE 2.2: Illustration of the difference method for a linear oscillator. The full, noisy time series is the solid blue line. The samples used for the difference method are indicated by orange dots. The last sample, which is used as the estimate of the fixed point, is indicated with a green ×, and its state value is shown with a horizontal dotted line.

at the correct phase is used as the estimate of the fixed point in the hope that most of the perturbation has died out by that time, but this is often not a good approximation. With respect to noise, since only a single sample per period is used in the calculation, the number of samples used in the calculation is small, and even a small amount of noise can have a large impact on the results.

Affine Transition Model with Ordinary Least Squares

Instead of estimating the unknown fixed point from only the last period, as in the difference method, an alternative approach is to estimate the fixed point as part of fitting a transition model to all the data [13, 14, 18]. It turns out that, with some algebra, the problem can be changed from estimating $\boldsymbol{P}(t_0)$ using estimates of the perturbation $\boldsymbol{\xi}$ to estimating $\boldsymbol{P}(t_0)$ and a vector $\boldsymbol{b}$ using the state $\boldsymbol{x}$. Let $\boldsymbol{x}_j = \boldsymbol{x}(t_0 + jT)$, then starting from eq. (2.4), it can be shown that, for $j = 0, 1, ...,$

$$\boldsymbol{x}_{j+1} = \boldsymbol{P}(t_0)\boldsymbol{x}_j + \boldsymbol{b}(t_0) \tag{2.6}$$

where

$$\boldsymbol{b}(t_0) = \left(\boldsymbol{I} - \boldsymbol{P}(t_0)\right) \boldsymbol{x}_{\mathrm{p}}(t_0) \tag{2.7}$$

Now, instead of fitting eq. (2.4) (where the parameters to fit are $\boldsymbol{P}(t_0)$) to the available pairs of estimated perturbations $\left(\boldsymbol{\xi}_j, \boldsymbol{\xi}_{j+1}\right)$, eq. (2.6) can be fit (where the

parameters to fit are $\boldsymbol{P}(t_0)$ and $\boldsymbol{b}(t_0)$) to the available pairs of state values $\left(\boldsymbol{x}_j, \boldsymbol{x}_{j+1}\right)$. With this approach, an estimate of $\boldsymbol{x}_{\mathrm{p}}(t_0)$ is not needed in order to estimate $\boldsymbol{P}(t_0)$, so the steady-state behavior of the system does not need to be observed. However, this method still suffers from noise since it uses only a single sample per period.

Moving Integral with Ordinary Least Squares

An approach to reduce the effect of measurement noise is to numerically integrate the state measurements [19]. Then, these integrated values can be used instead of perturbation estimates to estimate the characteristic multipliers. It was shown that if the integral of the steady-state response is zero, then the CMs for a transition matrix using the integrated state values are the same as the CMs for a transition matrix using the perturbation.

In other words, if the numerical integral of a constant integer number of periods starting at period j of the response $\boldsymbol{x}(t)$ is denoted $\bar{\boldsymbol{x}}_j$, and the numerical integral of one period of $\boldsymbol{x}_{\mathrm{p}}(t)$ is $\mathbf{0}$, then it can be shown that there exists a matrix $\bar{\boldsymbol{P}}(t_0)$ such that

$$\bar{\boldsymbol{x}}_{j+1} = \bar{\boldsymbol{P}}(t_0)\bar{\boldsymbol{x}}_j \tag{2.8}$$

where the eigenvalues of $\bar{\boldsymbol{P}}(t_0)$ are equal to the eigenvalues of $\boldsymbol{P}(t_0)$. So, after computing the values of $\bar{\boldsymbol{x}}_j$, the approach is analogous to the difference method—stack together samples $\bar{\boldsymbol{x}}_j$ into matrices and solve for $\bar{\boldsymbol{P}}(t_0)$ using ordinary least squares, and then the eigenvalues of $\bar{\boldsymbol{P}}(t_0)$ are the CM estimates.

This method avoids the need to explicitly estimate the fixed point by assuming that the mean of the steady-state response $\boldsymbol{x}_{\mathrm{p}}(t)$ is zero. Additionally, by integrating over many data points, this approach can substantially reduce the effect of measurement noise on the estimated CMs. When the assumption that the numerical mean of steady-state response is zero is valid, this method provides much better results

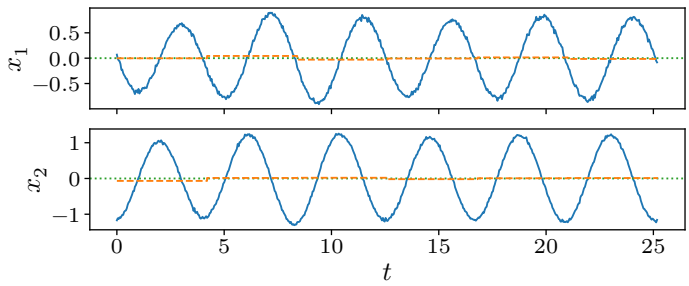


FIGURE 2.3: Illustration of the moving integral method with an integration window of one period. The full, noisy time series is the solid blue line. The integral of each period is indicated by the solid orange line. Zero is used as the estimate of the integral of the fixed point; this value is indicated with a horizontal dotted line.

than the difference method. Unfortunately, this assumption is often violated for real systems.

2.2.2 Limitations

The previously-described methods all work well for certain cases. They give good results if the measurement noise is small relative to the size of the perturbation, the characteristic multipliers are distinct, and the fixed point can be accurately estimated or the mean of the steady-state response is zero. However, those conditions are often violated for real systems, so they are important limitations when attempting to apply the existing methods in practice.

Estimating or Constraining the Steady-State Response

The first limitation is that the existing difference and moving integral methods either require a good estimate of the steady-state response or require it to have zero mean. These requirements can be difficult to meet in practice, which limits the applicability of the methods.

The usual way to estimate the steady-state response for the difference method is to assume that the last period of data approximates the steady-state response. However, when estimating CMs to describe the properties of a dynamical system,

some of the most important cases to estimate accurately are when the system is near the stability boundary; in these cases, a characteristic multiplier is near 1 and the perturbation dies out slowly. In these cases, a very long time series may be necessary for the last period to be a good approximation of the steady-state response. Collecting a long time series is not always practical when many parameter combinations need to be tested or when the timing of the initial perturbations is not under the control of the experimenter. Additionally, even if a long time series can be collected, measurement noise will influence the estimate of the steady-state response.

These limitations can be avoided by using an affine transition model on the state, which avoids the need to explicitly estimate the fixed point. Since the fixed point is estimated implicitly from the full time series instead of just using the last period, the perturbation does not need to die out by the end of the time series, and the effect of noise in the last data point is reduced. However, this idea has not been explored in combination with the moving integral method, which also reduces the effect of noise.

For the moving integral method as described in the literature [19], requiring the steady-state response to have zero mean is problematic because many real systems do not meet this criterion. A simple example is a forced harmonic oscillator where the forcing has nonzero mean. The experimenter could try to estimate the shift necessary to make the steady-state response have zero mean, but that brings back the problem of accurately estimating the steady-state response.

To estimate CMs in practice, it would be useful to have a method that would allow use of the moving integral in order to reduce noise, but not require directly estimating the steady-state response or constraining it to have zero mean.

Ordinary Least Squares is Biased

The existing methods solve for the transition matrix using ordinary least squares with their transition models—eqs. (2.4), (2.6), and (2.8)—and estimates of $\boldsymbol{\xi}$, $\boldsymbol{x}$,

or $\bar{\boldsymbol{x}}$. The problem is that noise is present in estimates of both the independent variables, i.e. $\boldsymbol{\xi}_j$, $\boldsymbol{x}_j$ or $\bar{\boldsymbol{x}}_j$, and the dependent variables, i.e. $\boldsymbol{\xi}_{j+1}$, $\boldsymbol{x}_{j+1}$, or $\bar{\boldsymbol{x}}_{j+1}$, but ordinary least squares assumes zero noise in the independent variables. For example, for the case of eq. (2.4), if $\tilde{\boldsymbol{\xi}}_j = \boldsymbol{\xi}_j + \boldsymbol{\varepsilon}_j$ is the estimate of $\boldsymbol{\xi}_j$ with error $\boldsymbol{\varepsilon}_j$, ordinary least squares assumes a regression model of the form

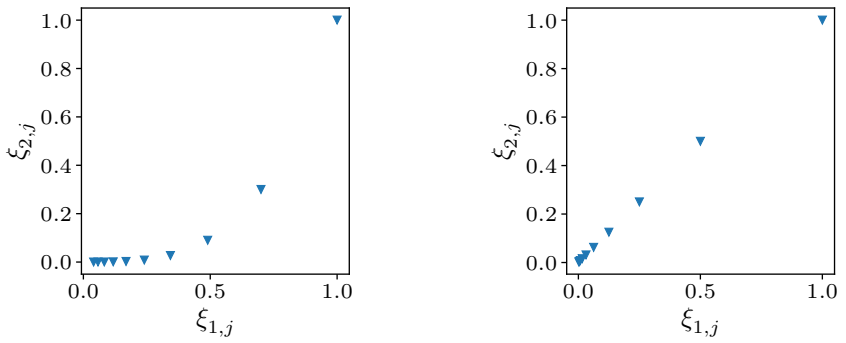
$$\tilde{\boldsymbol{\xi}}_{j+1} - \boldsymbol{\varepsilon}_{j+1} = \boldsymbol{P}(t_0)\boldsymbol{\xi}_j \tag{2.9}$$

which is not appropriate because $\boldsymbol{\xi}_j$ is not known exactly; only an estimate $\tilde{\boldsymbol{\xi}}_j$ is known. Since the existing methods use estimates of the independent variables in place of true values, the result of ordinary least squares is biased; this is known as regression dilution or attenuation bias. The resulting bias can produce surprisingly large errors in the characteristic multiplier estimates.

Similar Characteristic Multipliers Cause Numerical Issues

When multiple characteristic multipliers have similar values, the data matrices become ill-conditioned. For example, fig. 2.4 illustrates the progression of the perturbation sampled at a given phase in a single time series for a system with two state variables. When the CMs are distinct as in fig. 2.4a, the collection of perturbation vectors has rank two, but in the repeated CMs case shown in fig. 2.4b, all the perturbation vectors lie on a line passing through the origin, so they have rank one. More generally, the data matrices become rank-deficient whenever any characteristic multipliers are repeated. In these cases, insufficient information is available to estimate the transition matrix accurately.

This can be shown as follows. Let d be the number of state variables. Assume



(a) Distinct CMs (b) Repeated CMs

FIGURE 2.4: Effect of repeated characteristic multipliers on the rank of the perturbation time series for an example system with two state variables. The data points represent stroboscopically-sampled perturbations $\boldsymbol{\xi}_j = \begin{bmatrix}\xi_{1,j} & \xi_{2,j}\end{bmatrix}^\top$, $j = 0, 1, \ldots$. The horizontal axis is the perturbation $\xi_{1,j}$ in the first state variable, and the vertical axis is the perturbation $\xi_{2,j}$ in the second state variable. The points in fig. 2.4b lie on a straight line passing through the origin, while those in fig. 2.4a do not. In other words, the set of perturbation vectors in fig. 2.4b has rank 1, while the set of perturbation vectors in fig. 2.4a has rank 2.

that $\boldsymbol{P}(t_0)$ is diagonalizable, i.e. $\boldsymbol{P}(t_0)$ can be expressed as

$$\boldsymbol{P}(t_0) = \boldsymbol{V}(t_0)\boldsymbol{M}(\boldsymbol{V}(t_0))^{-1} \tag{2.10}$$

$$= \begin{bmatrix}\boldsymbol{v}_1(t_0) & \cdots & \boldsymbol{v}_d(t_0)\end{bmatrix} \begin{bmatrix}\mu_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mu_d\end{bmatrix} \begin{bmatrix}\boldsymbol{v}_1(t_0) & \cdots & \boldsymbol{v}_d(t_0)\end{bmatrix}^{-1} \tag{2.11}$$

where $\boldsymbol{v}_1(t_0), \ldots, \boldsymbol{v}_d(t_0)$ are linearly independent eigenvectors and $\mu_1, \ldots, \mu_d$ are the eigenvalues. Without loss of generality, let

$$\boldsymbol{\xi}_j = \boldsymbol{V}(t_0)\boldsymbol{s}_j = s_{1,j}\boldsymbol{v}_1 + \cdots + s_{d,j}\boldsymbol{v}_d \quad \boldsymbol{s}_j = \begin{bmatrix}s_{1,j} \\ \vdots \\ s_{d,j}\end{bmatrix} \quad \boldsymbol{S}_j = \begin{bmatrix}\boldsymbol{s}_j & \cdots & \boldsymbol{s}_{j+m-1}\end{bmatrix} \tag{2.12}$$

so that

$$\boldsymbol{\Xi}_j = \boldsymbol{V}(t_0)\boldsymbol{S}_j \tag{2.13}$$

Note that since $\boldsymbol{V}(t_0)$ is a full-rank square matrix, $\text{rank}(\boldsymbol{\Xi}_j) = \text{rank}(\boldsymbol{S}_j)$. Rearrang-

ing and substituting,

$$\boldsymbol{s}_{j+1} = (\boldsymbol{V}(t_0))^{-1}\boldsymbol{\xi}_{j+1} = (\boldsymbol{V}(t_0))^{-1}\boldsymbol{P}(t_0)\boldsymbol{\xi}_j = (\boldsymbol{V}(t_0))^{-1}\boldsymbol{V}(t_0)\boldsymbol{M}(\boldsymbol{V}(t_0))^{-1}\boldsymbol{\xi}_j = \boldsymbol{M}\boldsymbol{s}_j \tag{2.14}$$

Then, it can be shown that

$$\boldsymbol{s}_{j+i} = \boldsymbol{M}^i\boldsymbol{s}_j = \begin{bmatrix} \mu_1^i & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mu_d^i \end{bmatrix} \begin{bmatrix} s_{1,j} \\ \vdots \\ s_{d,j} \end{bmatrix} = \begin{bmatrix} \mu_1^i s_{1,j} \\ \vdots \\ \mu_d^i s_{d,j} \end{bmatrix} \tag{2.15}$$

Substituting,

$$\boldsymbol{S}_j = \begin{bmatrix} \mu_1^0 s_{1,j} & \cdots & \mu_1^{m-1} s_{1,j} \\ \vdots & \ddots & \vdots \\ \mu_d^0 s_{d,j} & \cdots & \mu_d^{m-1} s_{d,j} \end{bmatrix} = \begin{bmatrix} s_{1,j} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & s_{d,j} \end{bmatrix} \begin{bmatrix} \mu_1^0 & \cdots & \mu_1^{m-1} \\ \vdots & \ddots & \vdots \\ \mu_d^0 & \cdots & \mu_d^{m-1} \end{bmatrix} \tag{2.16}$$

Let

$$\boldsymbol{U}_j = \begin{bmatrix} s_{1,j} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & s_{d,j} \end{bmatrix} \qquad \boldsymbol{Z} = \begin{bmatrix} \mu_1^0 & \cdots & \mu_1^{m-1} \\ \vdots & \ddots & \vdots \\ \mu_d^0 & \cdots & \mu_d^{m-1} \end{bmatrix} \tag{2.17}$$

then

$$\boldsymbol{S}_j = \boldsymbol{U}_j\boldsymbol{Z} \tag{2.18}$$

If and only if all $s_{1,j}, \ldots, s_{d,j}$ are nonzero, the $d \times d$ diagonal matrix $\boldsymbol{U}_j$ is full-rank, and $\text{rank}\left(\boldsymbol{S}_j\right) = \text{rank}\left(\boldsymbol{Z}\right)$. If any of them are zero, then $\text{rank}\left(\boldsymbol{S}_j\right) \leq \text{rank}\left(\boldsymbol{U}_j\right) < d$. The $d \times m$ matrix $\boldsymbol{Z}$ is a Vandermonde matrix, which can be shown to have rank d if and only if $\mu_1, \ldots, \mu_d$ are distinct (for the case $m \geq d$).

In summary, $\text{rank}\left(\boldsymbol{\Xi}_0\right) = \text{rank}\left(\boldsymbol{S}_0\right) = d$ if and only if $s_{1,0}, \ldots, s_{d,0}$ are nonzero and $\mu_1, \ldots, \mu_d$ are distinct. So, the data matrix is rank-deficient in the case of repeated eigenvalues or an initial transient that doesn't contain all the eigenvectors. In this case, it is not possible to estimate all of the CMs with a single time series. For example, when using ordinary least squares, the matrix $\boldsymbol{\Xi}_0{\boldsymbol{\Xi}_0}^\top$ is non-invertible in this case since $\text{rank}\left(\boldsymbol{\Xi}_0{\boldsymbol{\Xi}_0}^\top\right) = \text{rank}\left(\boldsymbol{\Xi}_0\right)$.

Repeated characteristic multipliers can occur fairly frequently. For example, the linear oscillator

$$\ddot{x}(t) + 2\zeta\omega_{\mathrm{n}}\dot{x}(t) + \omega_{\mathrm{n}}^2 x(t) = \Gamma \sin \Omega t \tag{2.19}$$

has characteristic multipliers $\mu_{1,2} = \mathrm{e}^{-\zeta\omega_{\mathrm{n}}T}\left(\cos\omega_{\mathrm{d}}T \pm \mathrm{i}\sin\omega_{\mathrm{d}}T\right)$ where $\omega_{\mathrm{d}} = \omega_{\mathrm{n}}\sqrt{1-\zeta^2}$, $T = 2\pi/\Omega$. The CMs are repeated whenever $\Omega = 2\omega_{\mathrm{d}}/k$, where $k = 1, 2, ...$, so there are infinitely many forcing frequencies which result in repeated CMs.

2.3 Methods

This section introduces solutions for all of the limitations described in the previous section of the existing methods. The limitation of constraining the steady-state response for the moving integral method can be resolved with an affine transition model; the bias of ordinary least squares can be resolved by using total least squares; and the case of repeated characteristic multipliers can be handled by collecting multiple time series.

2.3.1 Moving Integral with Nonzero Mean

The moving integral method as described in the literature assumes that the integral of the steady-state response is zero [19]. It would be useful to extend the method to systems where this assumption is not met. It is also worth investigating how the approximate nature of numerical integration compared to the true integral affects the accuracy of the method in order to determine, for example, if numerically integrating with a small number of samples causes errors in the characteristic multiplier estimates.

Typical numerical integration rules, such as Riemann sums, trapezoidal rule, and Simpson's rule can be expressed as a linear combination of samples,

$$\bar{\boldsymbol{x}}(t) = \sum_{\tau\in\mathcal{T}} \kappa(\tau)\boldsymbol{x}(t+\tau) \tag{2.20}$$

where $\mathcal{T}$ is a set of time offsets and κ is a scalar coefficient function. For example, $\bar{\boldsymbol{x}}(t)$ is a left Riemann sum of $\boldsymbol{x}(t)$ over one period if $\mathcal{T} = \{iT/n \mid i = 0, 1, \ldots, n-1\}$ and $\kappa(\tau) = T/n$, but different choices of $\mathcal{T}$ and κ can be used to express a wide variety of other numerical integration rules. Unlike in [19], the set of time offsets does not need to correspond to an integer number of periods; it can be arbitrary.

As described in Appendix A.4, by linearizing about the periodic solution, applying Floquet's theorem, and performing some manipulation, the transition over one period can be expressed as

$$\bar{\boldsymbol{\xi}}(t+T) = \bar{\boldsymbol{P}}(t)\bar{\boldsymbol{\xi}}(t) \tag{2.21}$$

where $\bar{\boldsymbol{x}}_{\mathrm{p}}(t) = \sum_{\tau\in\mathcal{T}} \kappa(\tau)\boldsymbol{x}_{\mathrm{p}}(t+\tau)$, $\bar{\boldsymbol{\xi}}(t) = \sum_{\tau\in\mathcal{T}} \kappa(\tau)\boldsymbol{\xi}(t+\tau) = \bar{\boldsymbol{x}}(t) - \bar{\boldsymbol{x}}_{\mathrm{p}}(t)$, and $\bar{\boldsymbol{P}}(t)$ has the same eigenvalues as $\boldsymbol{P}(t)$.

Analogously to eq. (2.4), by defining $\bar{\boldsymbol{\xi}}_j = \bar{\boldsymbol{\xi}}(t_0 + jT)$ for some initial time t_0, this can be written as

$$\bar{\boldsymbol{\xi}}_{j+1} = \bar{\boldsymbol{P}}(t_0)\bar{\boldsymbol{\xi}}_j \tag{2.22}$$

From this equation, it's clear why Little et al. [19] constrained the integral of the steady-state response to be zero; eq. (2.8) can be derived from eq. (2.22) only if $\bar{\boldsymbol{x}}_{\mathrm{p}}(t_0) = \boldsymbol{0}$. In contrast, eq. (2.22) is valid even in the nonzero case. Similarly, eq. (2.22) avoids any issues related to numerical integration error due to a finite number of samples, since it is valid for any choice of $\mathcal{T}$ and κ, even $|\mathcal{T}| = 1$.

Analogous to eq. (2.6), with some algebra, the transition can be written in terms of state (integrals) instead of perturbation (integrals),

$$\bar{\boldsymbol{x}}_{j+1} = \bar{\boldsymbol{P}}(t_0)\bar{\boldsymbol{x}}_j + \bar{\boldsymbol{b}}(t_0) \tag{2.23}$$

where $\bar{\boldsymbol{b}}(t_0) = \left(\boldsymbol{I} - \bar{\boldsymbol{P}}(t_0)\right)\bar{\boldsymbol{x}}_{\mathrm{p}}(t_0)$.

In summary, the transition models after numerical integration, eqs. (2.22) and (2.23), are in the same form as the original transition models, eqs. (2.4) and (2.6), and the

eigenvalues of the new transition matrix $\bar{\boldsymbol{P}}(t_0)$ are equal to the original characteristic multipliers. The difference method can be applied by subtracting the last numerical integral from the other numerical integrals and then fitting eq. (2.22). Alternatively, a better approach is to use the affine transition model, eq. (2.23), to implicitly estimate the fixed point while fitting the transition matrix. In other words, numerical integrals of samples of the state can be treated analogously to samples of the state for the purpose of CM estimation.

2.3.2 Total Least Squares

As described in Section 2.2.2, ordinary least squares produces a biased estimate of the transition matrix in the presence of noise. If $\tilde{\boldsymbol{\xi}}_j = \boldsymbol{\xi}_j + \boldsymbol{\varepsilon}_j$ is the estimate of $\boldsymbol{\xi}_j$ with error $\boldsymbol{\varepsilon}_j$, ordinary least squares assumes a regression model of the form

$$\tilde{\boldsymbol{\xi}}_{j+1} - \boldsymbol{\varepsilon}_{j+1} = \boldsymbol{P}(t_0)\boldsymbol{\xi}_j \tag{2.24}$$

which is incorrect because $\boldsymbol{\xi}_j$ is not known exactly; only an estimate $\tilde{\boldsymbol{\xi}}_j$ is known. The correct model is an errors-in-variables model,

$$\tilde{\boldsymbol{\xi}}_{j+1} - \boldsymbol{\varepsilon}_{j+1} = \boldsymbol{P}(t_0)\left(\tilde{\boldsymbol{\xi}}_j - \boldsymbol{\varepsilon}_j\right) \tag{2.25}$$

Total least squares is a method for handling this type of problem when the error vectors are independent and normally distributed. Using total least squares, it's possible to specify the relative variance of the noise in the inputs and outputs, including possibly different variances for different state variables. To estimate the transition matrix as in eq. (2.25), the variance of the noise is the same for the inputs and the outputs. The total least squares parameter values can be computed using a standard implementation, such as the SciPy library's [49] wrapper for the ODRPACK library [50]. The results in Section 2.4 demonstrate that total least squares eliminates the systematic bias of ordinary least squares when estimating CMs.

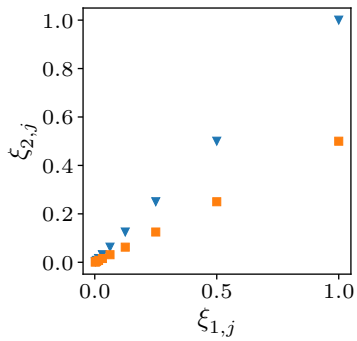


FIGURE 2.5: The effect of multiple time series when there are repeated characteristic multipliers. The perturbations from the two time series are illustrated with different markers. The union of the two time series is a set of perturbation vectors with rank 2, even though the set of perturbation vectors for each time series has rank 1. Contrast this with fig. 2.4b, which shows the perturbations from a single time series.

2.3.3 Handling Repeated Characteristic Multipliers

As described in Section 2.2.2, when multiple characteristic multipliers have similar values, the data matrices become ill-conditioned and numerical issues arise when estimating the CMs. A workaround for this case is to collect data starting from r different initial conditions $\boldsymbol{\xi}_0^{(1)} = \boldsymbol{V}(t_0)\boldsymbol{s}_0^{(1)}, \boldsymbol{\xi}_0^{(2)} = \boldsymbol{V}(t_0)\boldsymbol{s}_0^{(2)}, \ldots, \boldsymbol{\xi}_0^{(r)} = \boldsymbol{V}(t_0)\boldsymbol{s}_0^{(r)}$ at the same phase, in such a way that the rows of the $\boldsymbol{S}_0$ matrix corresponding to the repeated eigenvalues become linearly independent. For example, this is illustrated in fig. 2.5 for a two-dimensional system with repeated CMs. By starting from two linearly independent initial conditions, the rank of the data matrix containing both time series is two. When data is collected from r different initial conditions, the $\boldsymbol{S}_0$ matrix becomes

$$\boldsymbol{S}_0 = \left[\begin{array}{ccc|ccc|c|ccc} \boldsymbol{s}_0^{(1)} & \cdots & \boldsymbol{s}_{m/r-1}^{(1)} & \boldsymbol{s}_0^{(2)} & \cdots & \boldsymbol{s}_{m/r-1}^{(2)} & \cdots & \boldsymbol{s}_0^{(r)} & \cdots & \boldsymbol{s}_{m/r-1}^{(r)} \end{array}\right] \tag{2.26}$$

$$= \left[\begin{array}{ccc|ccc|c|ccc} \mu_1^0 s_{1,0}^{(1)} & \cdots & \mu_1^{m/r-1} s_{1,0}^{(1)} & \mu_1^0 s_{1,0}^{(2)} & \cdots & \mu_1^{m/r-1} s_{1,0}^{(2)} & \cdots & \mu_1^0 s_{1,0}^{(r)} & \cdots & \mu_1^{m/r-1} s_{1,0}^{(r)} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \mu_d^0 s_{d,0}^{(1)} & \cdots & \mu_d^{m/r-1} s_{d,0}^{(1)} & \mu_d^0 s_{d,0}^{(2)} & \cdots & \mu_d^{m/r-1} s_{d,0}^{(2)} & \cdots & \mu_d^0 s_{d,0}^{(r)} & \cdots & \mu_d^{m/r-1} s_{d,0}^{(r)} \end{array}\right] \tag{2.27}$$

where the time series are indicated with parenthesized superscripts, and the indices within the time series are indicated with subscripts. For illustration, consider the case where $\mu_i = \mu_j = \mu_k$ for some i, j, k. The corresponding rows of the original $\boldsymbol{S}_0$ matrix using a single initial condition are given by eq. (2.16) to be

$$\begin{bmatrix} \mu_i^0 s_{i,0} & \mu_i^1 s_{i,0} & \cdots & \mu_i^{m-1} s_{i,0} \end{bmatrix} = s_{i,0} \begin{bmatrix} \mu_i^0 & \mu_i^1 & \cdots & \mu_i^{m-1} \end{bmatrix} = s_{i,0} \begin{bmatrix} \mu_i^0 & \mu_i^1 & \cdots & \mu_i^{m-1} \end{bmatrix} \tag{2.28}$$

$$\begin{bmatrix} \mu_j^0 s_{j,0} & \mu_j^1 s_{j,0} & \cdots & \mu_j^{m-1} s_{j,0} \end{bmatrix} = s_{j,0} \begin{bmatrix} \mu_j^0 & \mu_j^1 & \cdots & \mu_j^{m-1} \end{bmatrix} = s_{j,0} \begin{bmatrix} \mu_i^0 & \mu_i^1 & \cdots & \mu_i^{m-1} \end{bmatrix} \tag{2.29}$$

$$\begin{bmatrix} \mu_k^0 s_{k,0} & \mu_k^1 s_{k,0} & \cdots & \mu_k^{m-1} s_{k,0} \end{bmatrix} = s_{k,0} \begin{bmatrix} \mu_k^0 & \mu_k^1 & \cdots & \mu_k^{m-1} \end{bmatrix} = s_{k,0} \begin{bmatrix} \mu_i^0 & \mu_i^1 & \cdots & \mu_i^{m-1} \end{bmatrix} \tag{2.30}$$

which are clearly linearly dependent. However, if data is collected for three different initial conditions in such a way that $\begin{bmatrix} s_{i,0}^{(1)} & s_{i,0}^{(2)} & s_{i,0}^{(3)} \end{bmatrix}$, $\begin{bmatrix} s_{j,0}^{(1)} & s_{j,0}^{(2)} & s_{j,0}^{(3)} \end{bmatrix}$, and $\begin{bmatrix} s_{k,0}^{(1)} & s_{k,0}^{(2)} & s_{k,0}^{(3)} \end{bmatrix}$ are linearly independent (which should almost always be the case if the initial conditions are generated by randomly perturbing the system), then the i, j, k rows of the $\boldsymbol{S}_0$ matrix become linearly independent; by eq. (2.27), they are:

$$\begin{bmatrix} \mu_i^0 s_{i,0}^{(1)} & \cdots & \mu_i^{m/r-1} s_{i,0}^{(1)} & \mu_i^0 s_{i,0}^{(2)} & \cdots & \mu_i^{m/r-1} s_{i,0}^{(2)} & \mu_i^0 s_{i,0}^{(3)} & \cdots & \mu_i^{m/r-1} s_{i,0}^{(3)} \end{bmatrix}$$
$$= \begin{bmatrix} \mu_i^0 s_{i,0}^{(1)} & \cdots & \mu_i^{m/r-1} s_{i,0}^{(1)} & \mu_i^0 s_{i,0}^{(2)} & \cdots & \mu_i^{m/r-1} s_{i,0}^{(2)} & \mu_i^0 s_{i,0}^{(3)} & \cdots & \mu_i^{m/r-1} s_{i,0}^{(3)} \end{bmatrix} \tag{2.31}$$

$$\begin{bmatrix} \mu_j^0 s_{j,0}^{(1)} & \cdots & \mu_j^{m/r-1} s_{j,0}^{(1)} & \mu_j^0 s_{j,0}^{(2)} & \cdots & \mu_j^{m/r-1} s_{j,0}^{(2)} & \mu_j^0 s_{j,0}^{(3)} & \cdots & \mu_j^{m/r-1} s_{j,0}^{(3)} \end{bmatrix}$$
$$= \begin{bmatrix} \mu_i^0 s_{j,0}^{(1)} & \cdots & \mu_i^{m/r-1} s_{j,0}^{(1)} & \mu_i^0 s_{j,0}^{(2)} & \cdots & \mu_i^{m/r-1} s_{j,0}^{(2)} & \mu_i^0 s_{j,0}^{(3)} & \cdots & \mu_i^{m/r-1} s_{j,0}^{(3)} \end{bmatrix} \tag{2.32}$$

$$\begin{bmatrix} \mu_k^0 s_{k,0}^{(1)} & \cdots & \mu_k^{m/r-1} s_{k,0}^{(1)} & \mu_k^0 s_{k,0}^{(2)} & \cdots & \mu_k^{m/r-1} s_{k,0}^{(2)} & \mu_k^0 s_{k,0}^{(3)} & \cdots & \mu_k^{m/r-1} s_{k,0}^{(3)} \end{bmatrix}$$
$$= \begin{bmatrix} \mu_i^0 s_{k,0}^{(1)} & \cdots & \mu_i^{m/r-1} s_{k,0}^{(1)} & \mu_i^0 s_{k,0}^{(2)} & \cdots & \mu_i^{m/r-1} s_{k,0}^{(2)} & \mu_i^0 s_{k,0}^{(3)} & \cdots & \mu_i^{m/r-1} s_{k,0}^{(3)} \end{bmatrix} \tag{2.33}$$

2.3.4 Combining These Improvements

Combining all these improvements in a single method avoids the limitations of previous methods. The combination of the affine model with total least squares is an errors-in-variables model,

$$\tilde{\boldsymbol{x}}_{j+1} - \boldsymbol{\varepsilon}_{j+1} = \boldsymbol{P}(t_0)\left(\tilde{\boldsymbol{x}}_j - \boldsymbol{\varepsilon}_j\right) + \boldsymbol{b}(t_0) \tag{2.34}$$

This model can be used with state measurements directly, or to reduce the effect of noise, with a moving sum/integral of the state measurements, as described in Section 2.3.1. When there is a possibility of repeated characteristic multipliers, multiple time series should be collected, as described in Section 2.3.3.

2.4 Validation Studies

In order to validate the proposed improvements and compare to existing methods, tests were conducted using numerically simulated dynamical systems with analytically-known characteristic multipliers. Figure 2.1 and table 2.1 summarize the tested methods. The SDO method is the traditional difference method described in Section 2.2.1, SAO uses the affine transition model as described in Section 2.2.1, IZO is the method described in Section 2.2.1, and IAT is the method proposed in this chapter. The other methods are various combinations of improvements to existing methods. In the following text, sets of methods may be referred to with an asterisk to represent 'any option'; for example, 'S*O' refers to the SDO and SAO methods. Numerical integrals for the I** methods were computed as left Riemann sums.

The number of simulated periods was varied between some test cases to illustrate the effect of the length of the time series. Recall that in order to perform the regression to estimate the transition matrix, a set of pairs of data points is needed, where the two points in each pair are separated by one period. For example, if three consecutive pairs of data points $\{(\boldsymbol{x}_0, \boldsymbol{x}_1), (\boldsymbol{x}_1, \boldsymbol{x}_2), (\boldsymbol{x}_2, \boldsymbol{x}_3)\}$ are used from a single

Table 2.1: Properties of the characteristic multiplier estimation methods. "S." refers to the number of samples per period used in the calculation, "Reg." refers to the regression technique, "OLS" refers to ordinary least squares, and "TLS" refers to total least squares.

Name	S.	Steady-state response	Reg.	Transition model	Error model
SDO	1	est. from last sample	OLS	$\boldsymbol{\xi}_{j+1} = \boldsymbol{P}(t_0)\boldsymbol{\xi}_j$	$\tilde{\boldsymbol{\xi}}_{j+1} - \boldsymbol{\varepsilon}_{j+1} = \boldsymbol{P}(t_0)\boldsymbol{\xi}_j$
SDT	1	est. from last sample	TLS	$\boldsymbol{\xi}_{j+1} = \boldsymbol{P}(t_0)\boldsymbol{\xi}_j$	$\tilde{\boldsymbol{\xi}}_{j+1} - \boldsymbol{\varepsilon}_{j+1} = \boldsymbol{P}(t_0)\left(\tilde{\boldsymbol{\xi}}_j - \boldsymbol{\varepsilon}_j\right)$
SAO	1	implicit (affine model)	OLS	$\boldsymbol{x}_{j+1} = \boldsymbol{P}(t_0)\boldsymbol{x}_j + \boldsymbol{b}(t_0)$	$\tilde{\boldsymbol{x}}_{j+1} - \boldsymbol{\varepsilon}_{j+1} = \boldsymbol{P}(t_0)\tilde{\boldsymbol{x}}_j + \boldsymbol{b}(t_0)$
SAT	1	implicit (affine model)	TLS	$\boldsymbol{x}_{j+1} = \boldsymbol{P}(t_0)\boldsymbol{x}_j + \boldsymbol{b}(t_0)$	$\tilde{\boldsymbol{x}}_{j+1} - \boldsymbol{\varepsilon}_{j+1} = \boldsymbol{P}(t_0)\left(\tilde{\boldsymbol{x}}_j - \boldsymbol{\varepsilon}_j\right) + \boldsymbol{b}(t_0)$
IZO	n	assume zero mean	OLS	$\bar{\boldsymbol{x}}_{j+1} = \bar{\boldsymbol{P}}(t_0)\bar{\boldsymbol{x}}_j$	$\tilde{\bar{\boldsymbol{x}}}_{j+1} - \boldsymbol{\varepsilon}_{j+1} = \bar{\boldsymbol{P}}(t_0)\bar{\boldsymbol{x}}_j$
IZT	n	assume zero mean	TLS	$\bar{\boldsymbol{x}}_{j+1} = \bar{\boldsymbol{P}}(t_0)\bar{\boldsymbol{x}}_j$	$\tilde{\bar{\boldsymbol{x}}}_{j+1} - \boldsymbol{\varepsilon}_{j+1} = \bar{\boldsymbol{P}}(t_0)\left(\tilde{\bar{\boldsymbol{x}}}_j - \boldsymbol{\varepsilon}_j\right)$
IDO	n	est. from last period	OLS	$\bar{\boldsymbol{\xi}}_{j+1} = \bar{\boldsymbol{P}}(t_0)\bar{\boldsymbol{\xi}}_j$	$\tilde{\bar{\boldsymbol{\xi}}}_{j+1} - \boldsymbol{\varepsilon}_{j+1} = \bar{\boldsymbol{P}}(t_0)\bar{\boldsymbol{\xi}}_j$
IDT	n	est. from last period	TLS	$\bar{\boldsymbol{\xi}}_{j+1} = \bar{\boldsymbol{P}}(t_0)\bar{\boldsymbol{\xi}}_j$	$\tilde{\bar{\boldsymbol{\xi}}}_{j+1} - \boldsymbol{\varepsilon}_{j+1} = \bar{\boldsymbol{P}}(t_0)\left(\tilde{\bar{\boldsymbol{\xi}}}_j - \boldsymbol{\varepsilon}_j\right)$
IAO	n	implicit (affine model)	OLS	$\bar{\boldsymbol{x}}_{j+1} = \bar{\boldsymbol{P}}(t_0)\bar{\boldsymbol{x}}_j + \bar{\boldsymbol{b}}(t_0)$	$\tilde{\bar{\boldsymbol{x}}}_{j+1} - \boldsymbol{\varepsilon}_{j+1} = \bar{\boldsymbol{P}}(t_0)\bar{\boldsymbol{x}}_j + \bar{\boldsymbol{b}}(t_0)$
IAT	n	implicit (affine model)	TLS	$\bar{\boldsymbol{x}}_{j+1} = \bar{\boldsymbol{P}}(t_0)\bar{\boldsymbol{x}}_j + \bar{\boldsymbol{b}}(t_0)$	$\tilde{\bar{\boldsymbol{x}}}_{j+1} - \boldsymbol{\varepsilon}_{j+1} = \bar{\boldsymbol{P}}(t_0)\left(\tilde{\bar{\boldsymbol{x}}}_j - \boldsymbol{\varepsilon}_j\right) + \bar{\boldsymbol{b}}(t_0)$

time series, then the time series would need to be four periods long so that it contains all four data points $\boldsymbol{x}_0$, $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, and $\boldsymbol{x}_3$. When comparing methods across different numbers of time series, the number of *pairs* of data points was held constant, which was slightly different from the number of periods of data.

For each test case, the system was simulated 10 000 times with randomly sampled initial conditions and noise in order to estimate the distributions of the characteristic multiplier estimates, which are shown as histograms. Ideally, the chosen method will consistently have a narrow distribution, centered about the true value of the CM.

2.4.1 Forced Linear Harmonic Oscillator

The first dynamical system considered was a forced linear harmonic oscillator,

$$\ddot{x}(t) + 2\zeta\omega_\mathrm{n}\dot{x}(t) + \omega_\mathrm{n}^2 x(t) = \Gamma \sin \Omega t + \Gamma_0 \tag{2.35}$$

which has characteristic multipliers $\mu_{1,2} = \mathrm{e}^{-\zeta\omega_\mathrm{n}T}\left(\cos\omega_\mathrm{d}T \pm \mathrm{i}\sin\omega_\mathrm{d}T\right)$, where $\omega_\mathrm{d} = \omega_\mathrm{n}\sqrt{1-\zeta^2}$, $T = 2\pi/\Omega$. The parameter values common to all cases were $\zeta = 0.05$, $\omega_\mathrm{n} = 1$, and $\Gamma = 1$, so this is a lightly-damped system.

For the first set of cases, the CMs are distinct. These cases show the effect of the number of samples per period used in the integral methods, the number of periods

of available data, and a nonzero mean steady-state response. For the second set of cases, the CMs are identical. These cases show the numerical issues that appear when using a single time series and how using a second time series fixes the issue.

Distinct Characteristic Multipliers

For these cases, the forcing frequency was chosen to be $\Omega = 1.5\omega_\mathrm{d}$, such that the characteristic multipliers are distinct complex conjugates. The distribution of the initial conditions was $\boldsymbol{x}(0) \sim \mathcal{N}\big(\boldsymbol{x}_\mathrm{p}(0), \mathrm{diag}(0.16^2, 0.24^2)\big)$, and observation noise in each component of the measurement of $\boldsymbol{x}(t)$ for each data point was independently sampled from $\mathcal{N}(0, 0.025^2)$. In other words, the standard deviation of the noise was about one eighth the standard deviation of the initial perturbation. Figure 2.6 shows the steady-state response and example perturbed time series when the Γ_0 is chosen such that mean steady-state response is zero (fig. 2.6a) or nonzero (fig. 2.6b).

The first case, shown in fig. 2.7, is the ideal case. The number of samples per period for the integral methods is large, there are many periods of data, and the mean of the steady-state response is zero. All the methods work fairly well in this case. However, the bias for the methods using ordinary least squares (**O) is clearly evident, especially for the SDO method. The methods using the affine transition model (SA*, IA*), which implicitly estimate the steady-state response from all available data, have slightly less spread than the difference methods (SD*, ID*), which explicitly estimate the steady-state response from the last period of data. The moving integral methods (I**) have substantially less spread than the single-sample-per-period methods, since they integrate over a large number of samples per period.

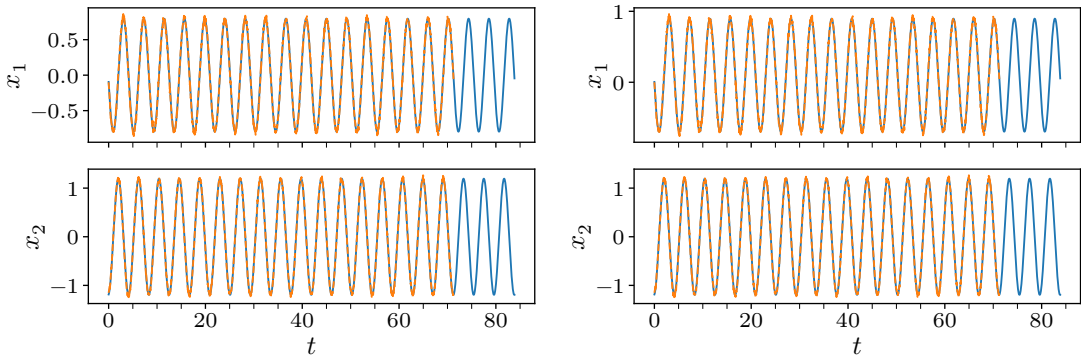
Figure 2.8 shows the effect of a nonzero mean steady-state response. The distributions for most of the methods are essentially unchanged, but the methods which assume that the mean of the steady-state response is zero (*Z*) perform very poorly. This shows the importance of using one of the other methods when the mean steady-

state response may be even slightly nonzero.

Figure 2.9 shows the effect of additionally reducing the number of periods of data. In this case, the last period is a poor approximation of the steady-state periodic response since the perturbation has not sufficiently died out, so the difference methods (*D*) no longer perform well. The only methods that work well in this case are those using the affine transition model (*A*).

Finally, fig. 2.10 shows the effect of additionally reducing the number of samples per period used by the integral methods (I**). In this case, the integration does not provide as substantial a benefit over the single-sample-per-period methods.

In summary, these results show how the proposed improvements to the methods produce more accurate results. Methods using the affine transition model (*A*) work well even when the mean of the steady-state response is nonzero and the last period is not a good approximation of the steady-state response. They also appear to produce slightly less spread than the equivalent difference methods (*D*). Replacing ordinary least squares (**O) with total least squares (**T) removes substantial bias in some cases. The IAT method, which combines the affine transition model, total least squares, and moving integral, produces as good or better results than the other methods in all of the cases.



(a) $\Gamma_0 = 0$ (b) $\Gamma_0 = 0.1$

FIGURE 2.6: True steady-state response (blue line) and an example noisy, perturbed time series (dashed orange line) with 16 sampled pairs of consecutive periods.

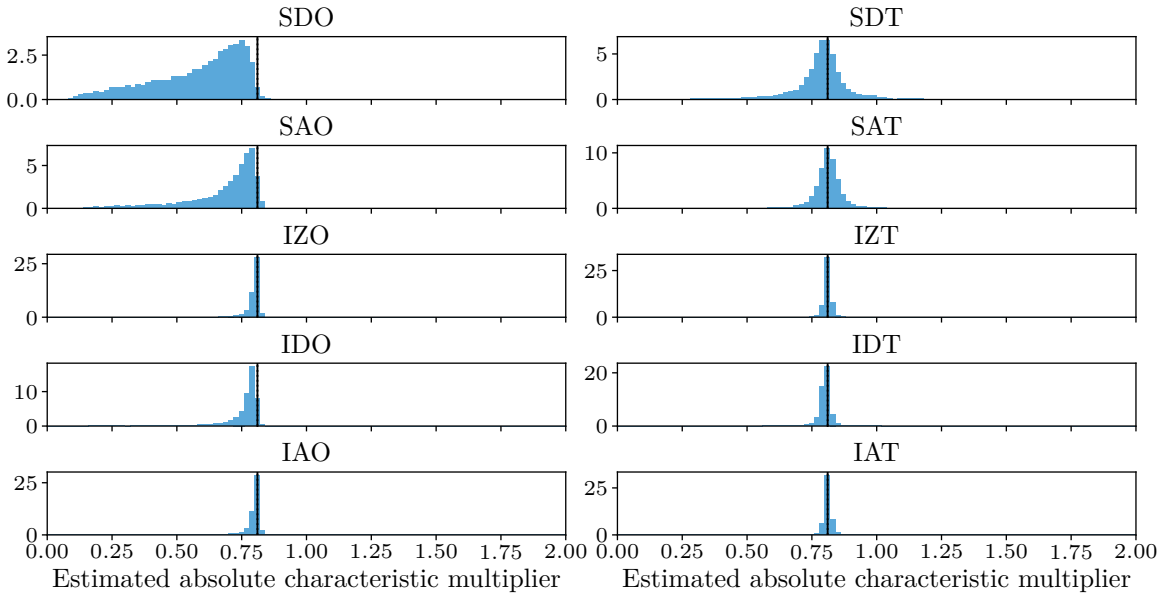


FIGURE 2.7: The distribution of the largest estimated absolute CM for each method, using 100 samples per period, 16 sampled pairs of consecutive periods, and $\Gamma_0 = 0$. The vertical line in each plot is the correct value.

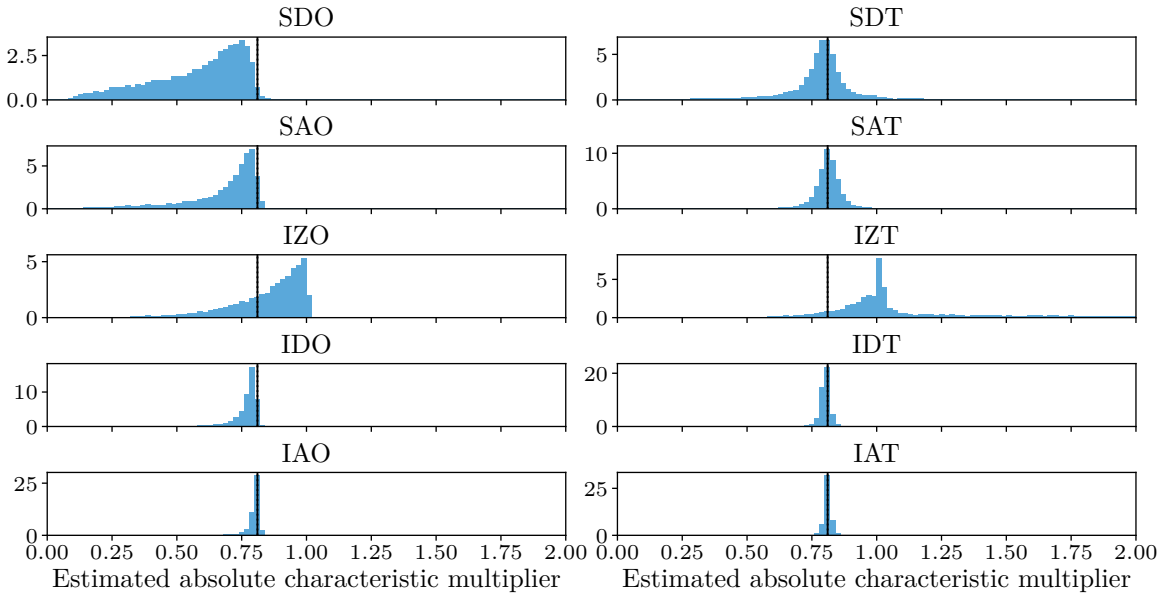


FIGURE 2.8: The distribution of the largest estimated absolute CM for each method, using 100 samples per period, 16 sampled pairs of consecutive periods, and $\Gamma_0 = 0.1$. The vertical line in each plot is the correct value.

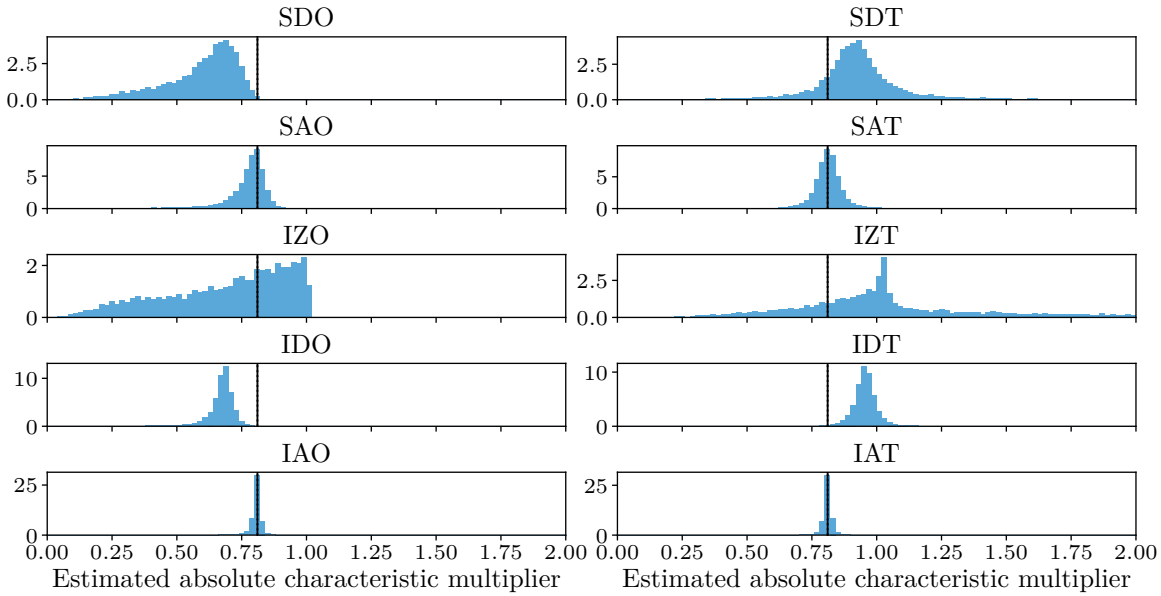


FIGURE 2.9: The distribution of the largest estimated absolute CM for each method, using 100 samples per period, 6 sampled pairs of consecutive periods, and $\Gamma_0 = 0.1$. The vertical line in each plot is the correct value.

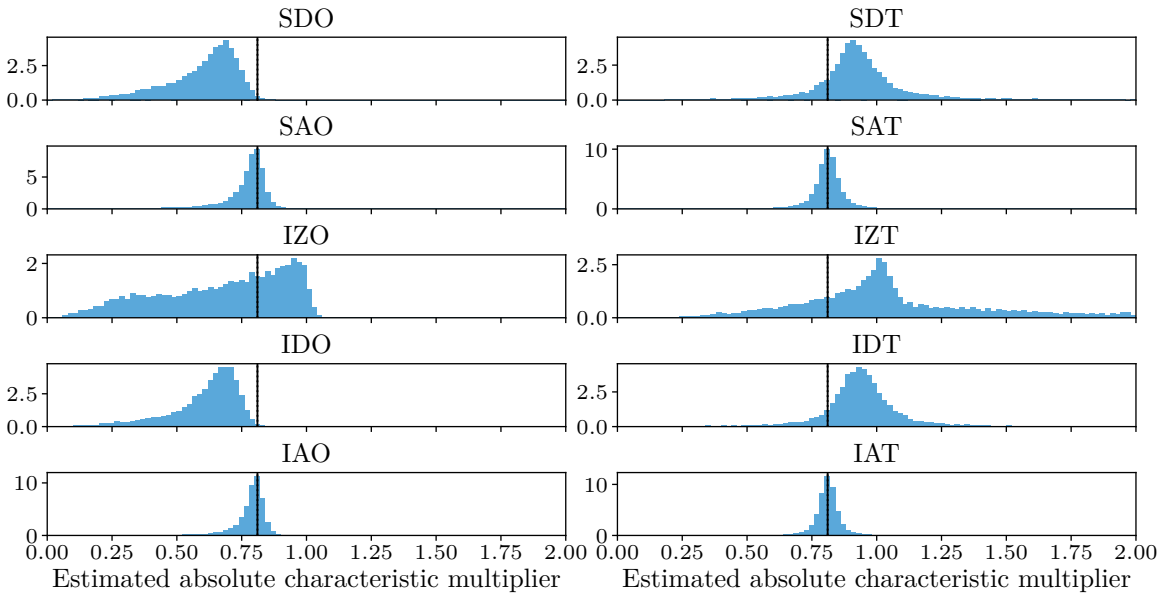
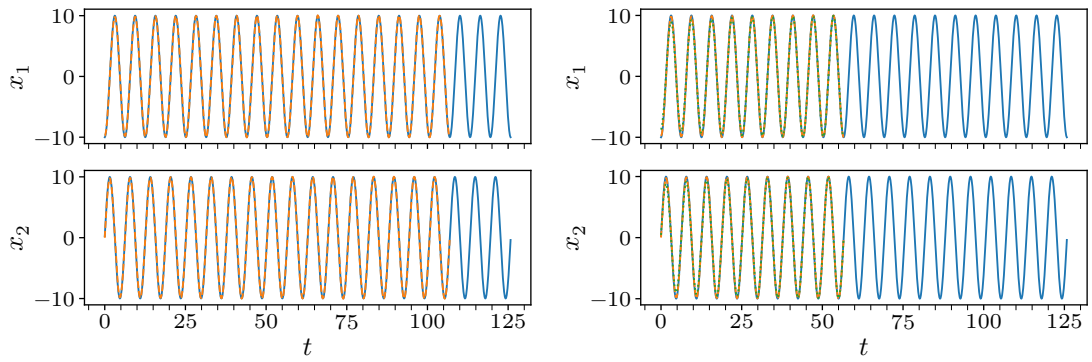


FIGURE 2.10: The distribution of the largest estimated absolute CM for each method, using 10 samples per period, 6 sampled pairs of consecutive periods, and $\Gamma_0 = 0.1$. The vertical line in each plot is the correct value.

Repeated Characteristic Multipliers

The previous section considered a forcing frequency that resulted in distinct characteristic multipliers. For the following case, the forcing frequency was chosen to be $\Omega = \omega_\mathrm{d}$ such that the two characteristic multipliers are identical. Recall that for this system, the CMs are repeated whenever $\Omega = 2\omega_\mathrm{d}/k$, where $k = 1, 2, ...$, so the $\Omega = \omega_\mathrm{d}$ case is just one of infinitely many cases with repeated CMs. The distribution of the initial conditions was $\boldsymbol{x}(0) \sim \mathcal{N}\left(\boldsymbol{x}_\mathrm{p}(0), \mathrm{diag}(2^2, 2^2)\right)$, and observation noise in each component of the measurement of $\boldsymbol{x}(t)$ for each data point was independently sampled from $\mathcal{N}(0, 0.025^2)$. Figure 2.11 shows the steady-state response and example perturbed time series.

Figures 2.12 and 2.13 show the distributions of the estimates of the absolute values of the two characteristic multipliers when using a single time series; fig. 2.12 shows the larger estimate, and fig. 2.13 shows the smaller estimate. As expected, none of the methods accurately estimates both characteristic multipliers when using



(a) One time series.

(b) Two time series.

FIGURE 2.11: True steady-state response (blue line) and example noisy, perturbed time series (dashed orange / dotted green lines) with 16 total sampled pairs of periods.

a single time series. Figures 2.14 and 2.15 show the results when using two time series. All of the methods perform very well in this case, except for the difference methods (*D*), which have a lower CM estimate slightly below the correct value.

In summary, when the characteristic multipliers are not distinct, it is necessary to collect multiple time series to be able to accurately estimate all of the CMs.

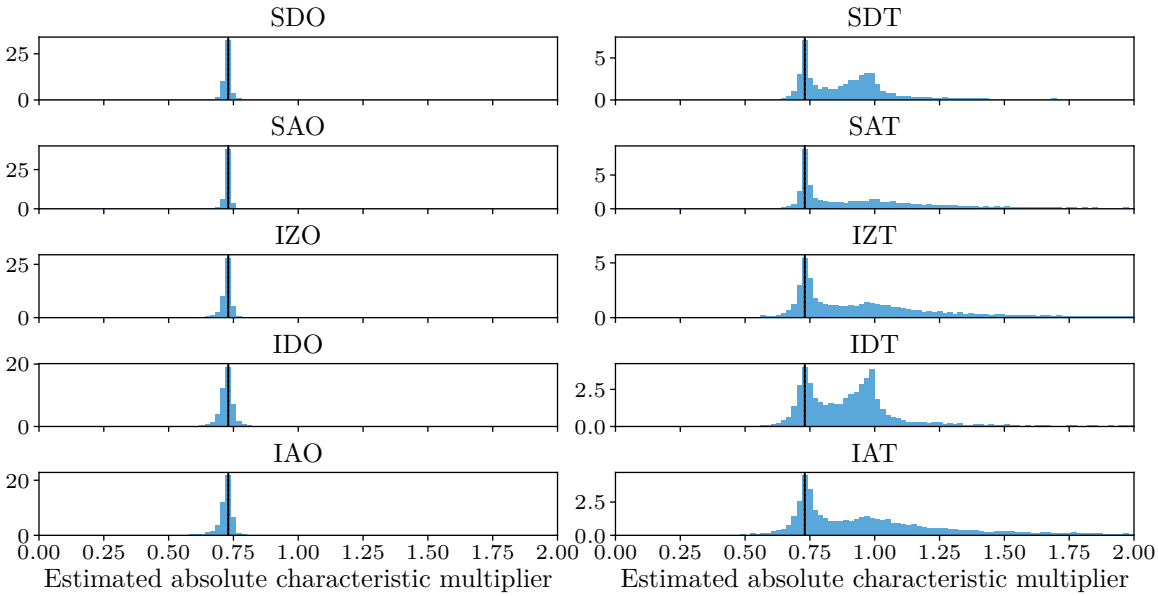


FIGURE 2.12: The distribution of the largest estimated absolute CM for each method, for a single time series with a single initial condition, with 100 samples per period, 16 sampled pairs of consecutive periods, and $\Gamma_0 = 0$. The vertical line in each plot is the correct value.

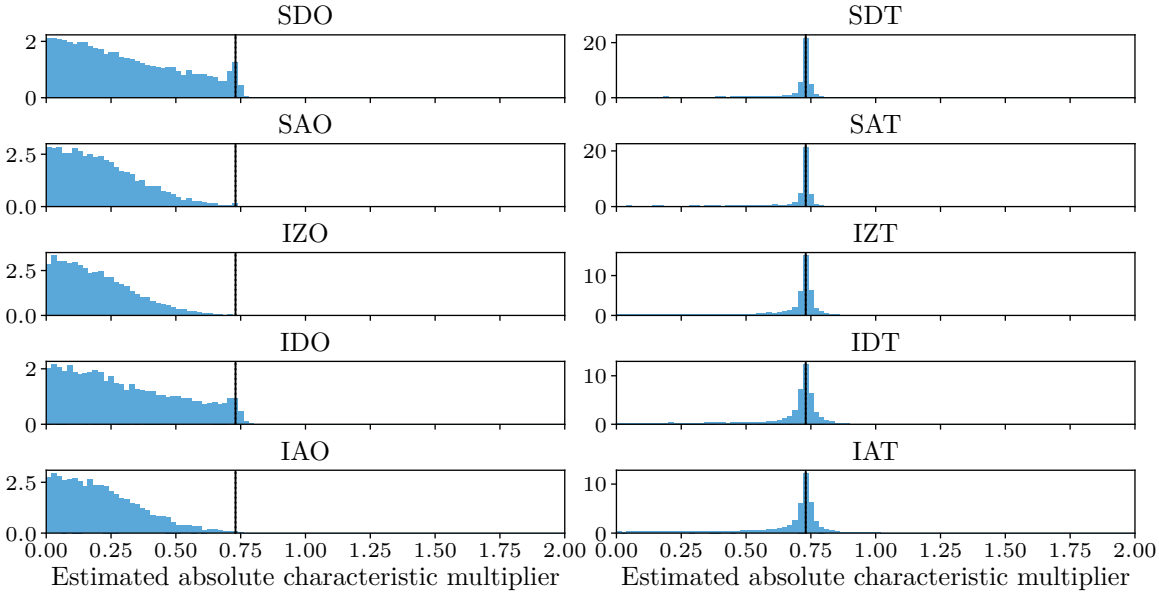


FIGURE 2.13: The distribution of the smallest estimated absolute CM for each method, for a single time series with a single initial condition, with 100 samples per period, 16 sampled pairs of consecutive periods, and $\Gamma_0 = 0$. The vertical line in each plot is the correct value.

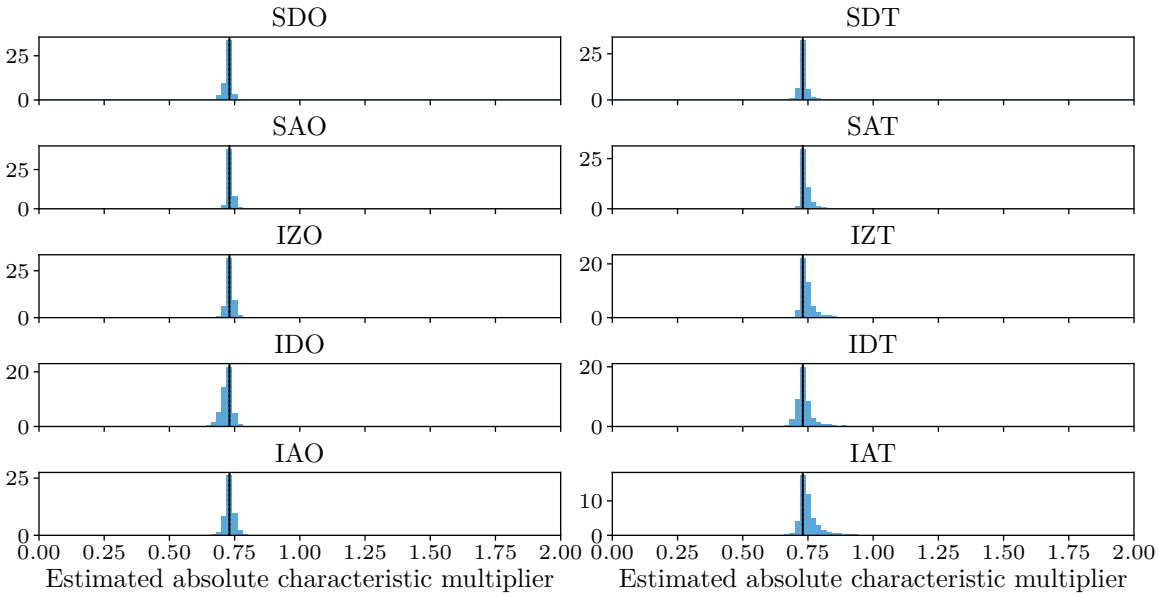


FIGURE 2.14: The distribution of the largest estimated absolute CM for each method, for two time series with different initial conditions, with 100 samples per period, 16 sampled pairs of consecutive periods, and $\Gamma_0 = 0$. The vertical line in each plot is the correct value.

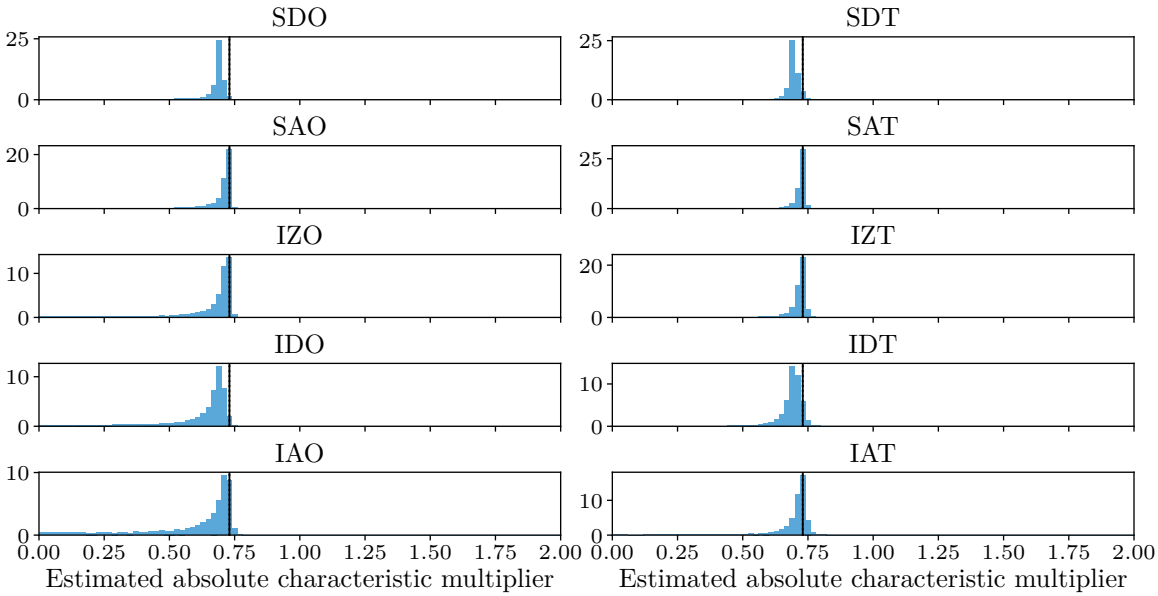


FIGURE 2.15: The distribution of the smallest estimated absolute CM for each method, for two time series with different initial conditions, with 100 samples per period, 16 sampled pairs of consecutive periods, and $\Gamma_0 = 0$. The vertical line in each plot is the correct value.

2.4.2 Higher-Dimensional System

A system with four state variables was also tested to demonstrate the applicability of the methods to higher-dimensional systems and systems with non-constant coefficients. The dynamics of the system were described by

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}(t)\boldsymbol{x}(t) + \boldsymbol{g}(t) \tag{2.36}$$

where $\boldsymbol{A}(t)$ was a matrix-valued periodic function with period 1 and $\boldsymbol{g}(t)$ was a vector-valued periodic function with the same period. The details are provided in Appendix B. The system was constructed to have an analytically-known periodic steady-state solution and analytically-known characteristic multipliers with values $0.5, 0.75\mathrm{e}^{\mathrm{i}}, 0.75\mathrm{e}^{-\mathrm{i}}, 0.9$. The distribution of the initial conditions was $\boldsymbol{x}(0) \sim \mathcal{N}\Big(\boldsymbol{x}_{\mathrm{p}}(0), \mathrm{diag}(0.2^2, 0.2^2, 0.1^2, 0.1^2)\Big)$, and observation noise in each component of the measurement of $\boldsymbol{x}(t)$ for each data point was independently sampled from $\mathcal{N}(0, 0.025^2)$. Figure 2.16 shows the steady-state periodic solution and an example noisy, perturbed time series.

Figures 2.17 to 2.20 show the results of estimating the characteristic multipliers using all of the methods. Figure 2.17 shows the distribution of the largest of the absolute values of the CM estimates, fig. 2.18 shows the distribution of the second largest, etc. Some of the single-sample-per-period methods (S**) perform reasonably well for the larger characteristic multipliers, but the noise prevents them from accurately estimating the smallest characteristic multiplier. The methods which assume that the mean of the steady-state response is zero (IZ*) perform poorly for all the characteristic multipliers because this assumption is violated. The difference methods using the moving integral (ID*) perform well except for the largest characteristic multiplier, which corresponds to a slower decay and the last period not approximating the steady-state response as well. The only methods which produce good estimates of all of the characteristic multipliers are the methods using the affine

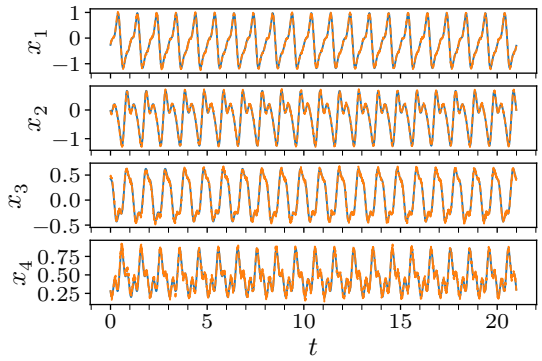


FIGURE 2.16: True steady-state response (blue line) and an example noisy, perturbed time series (dashed orange line) for 20 sampled pairs of periods.

transition model and moving integral (IA*). The integral is sufficient to reduce the noise to a level where the difference between ordinary least squares (IAO) and total least squares (IAT) is not very noticeable in this case.

In summary, the method combining the affine transition model, total least squares, and moving integral (IAT) provides the most accurate characteristic multiplier estimates. The results demonstrate that it works well even for higher-dimensional systems with non-constant coefficients.

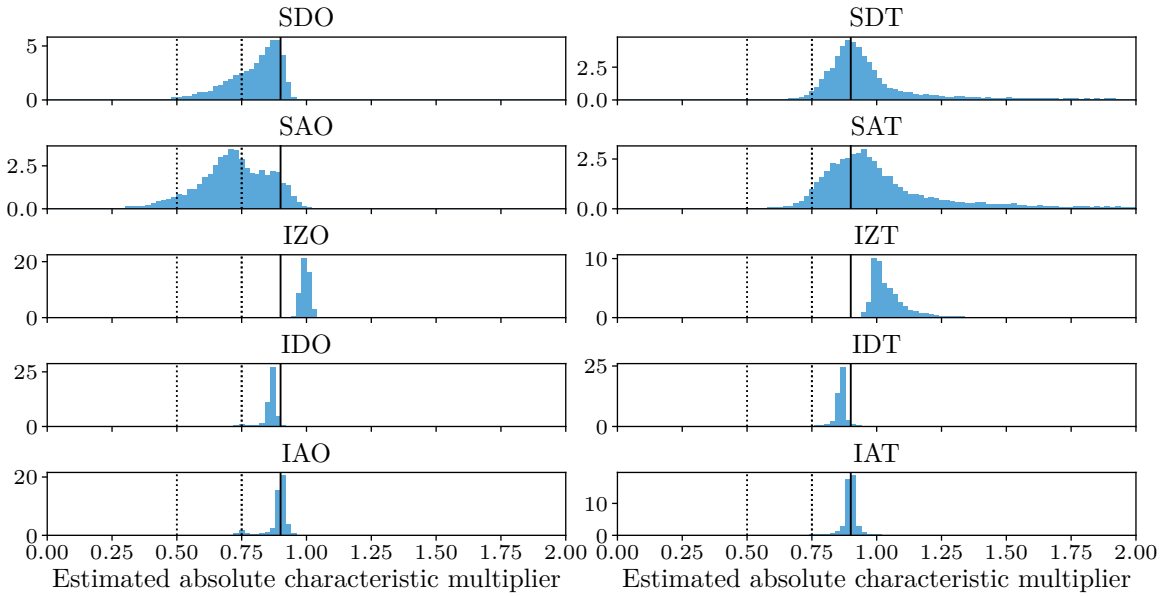


FIGURE 2.17: The distribution of the largest estimated absolute CM for each method, using a single time series with 100 samples per period and 20 sampled pairs of consecutive periods. The true CMs are indicated with vertical dotted or solid lines, with the largest indicated by the solid line.

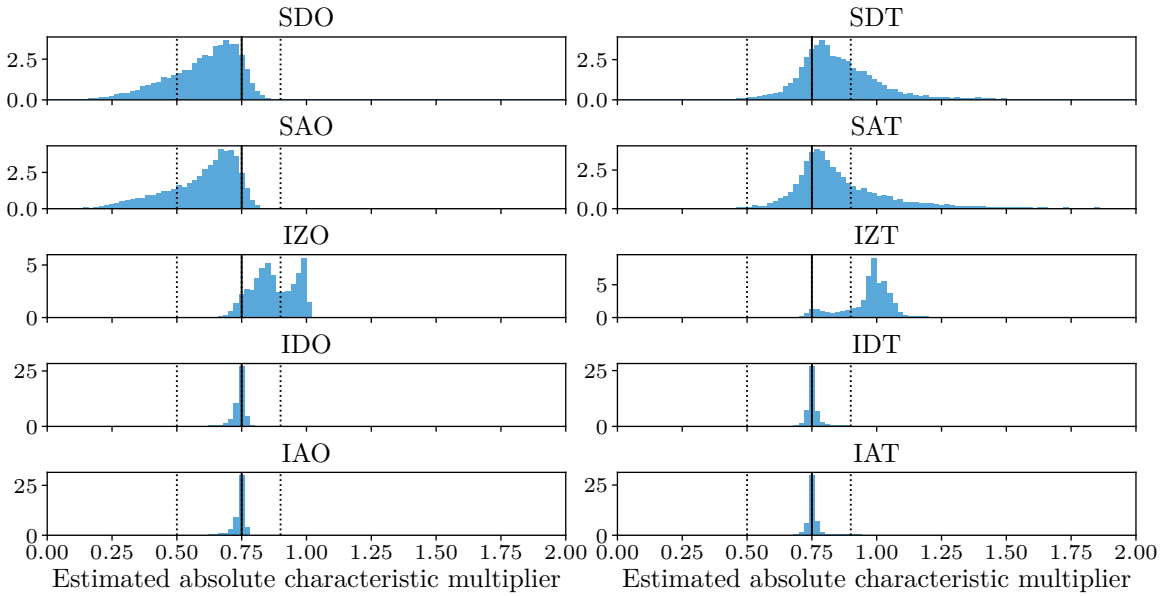


FIGURE 2.18: The distribution of the second largest estimated absolute CM for each method, using a single time series with 100 samples per period and 20 sampled pairs of consecutive periods. The true CMs are indicated with vertical dotted or solid lines, with the second largest indicated by the solid line.

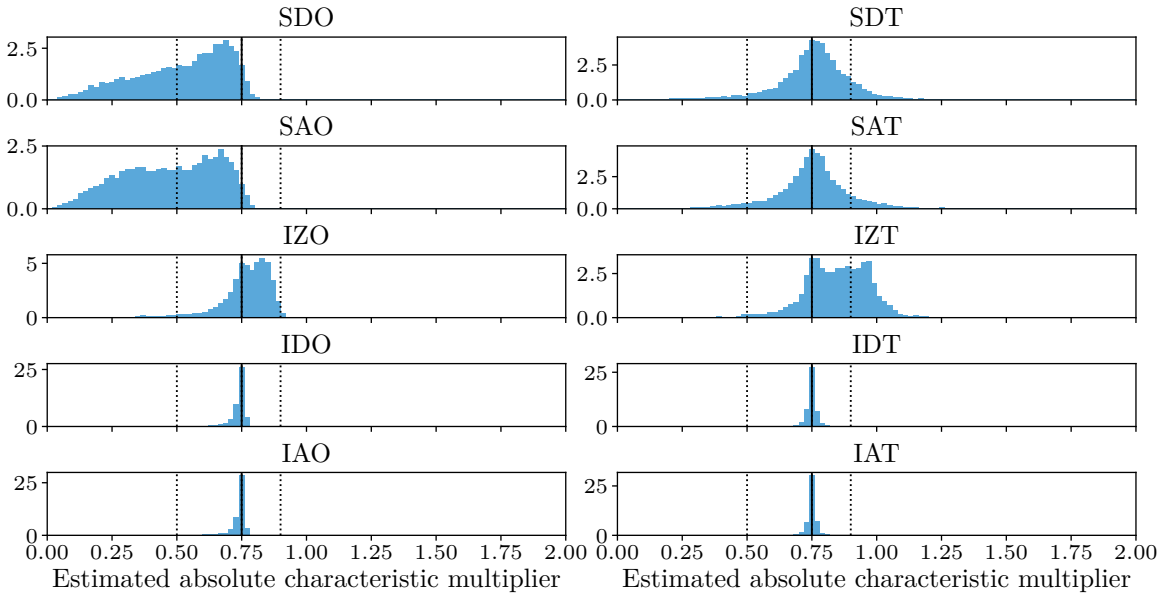


FIGURE 2.19: The distribution of the third largest estimated absolute CM for each method, using a single time series with 100 samples per period and 20 sampled pairs of consecutive periods. The true CMs are indicated with vertical dotted or solid lines, with the third largest indicated by the solid line.

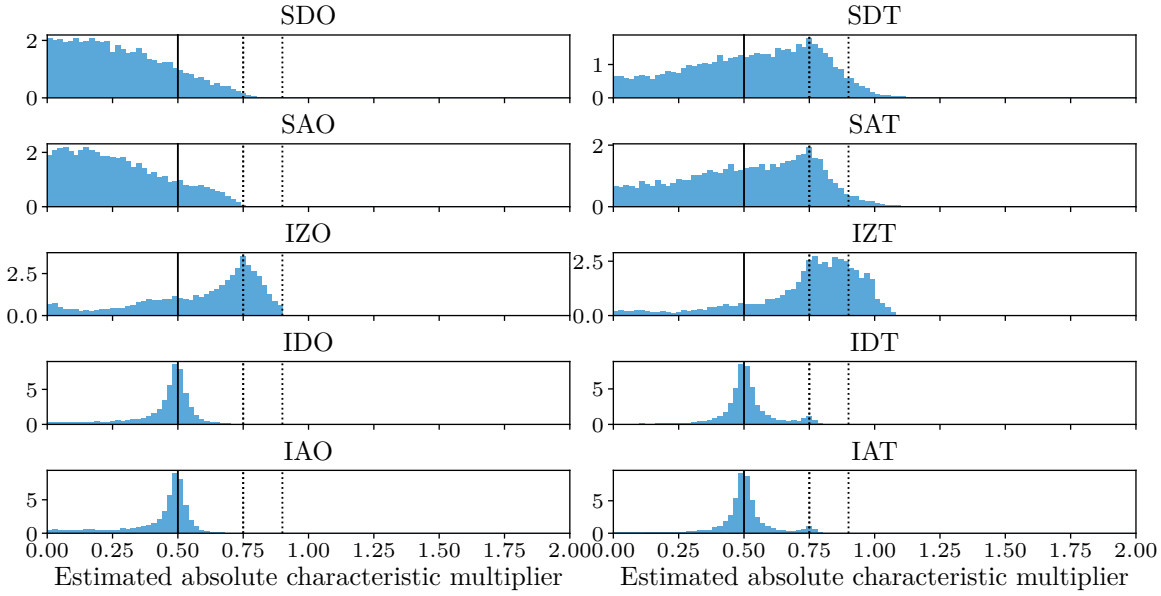


FIGURE 2.20: The distribution of the smallest estimated absolute CM for each method, using a single time series with 100 samples per period and 20 sampled pairs of consecutive periods. The true CMs are indicated with vertical dotted or solid lines, with the smallest indicated by the solid line.

2.5 Conclusion

CMs describe the stability of a time-periodic steady-state response of a dynamical system. CMs can be experimentally estimated using a time series of the system's state following an initial perturbation. However, existing CM estimation methods have practical limitations, including a need to estimate or constrain the steady-state response, bias due to ordinary least squares, and an inability to handle repeated characteristic multipliers. This chapter shows that limitations can be solved, respectively, by using an affine transition model on the state or state integral, using total least squares instead of ordinary least squares, and collecting multiple independent time series. Simulation results from a variety of scenarios show that a method combining these improvements and the moving integral approach proposed by Little et al. [19] produces more accurate CM estimates than existing methods. This new CM estimation method avoids the practical limitations of existing methods with minimal additional complexity, while producing more accurate results.

3

Estimating Parameters of Delay Differential Equations from Time Series

3.1 Introduction

Delay differential equations (DDEs) can model many real-world systems and can be used to make stability predictions. Unfortunately, in many cases, some of the model parameters are unknown and cannot be measured directly. As a result, it is useful to be able to estimate the unknown model parameters given time series measurements of the system's state. Specifically, we are interested in estimating parameters for multivariate, linear DDEs with periodic coefficients from multiple time series. The estimated parameters can then be used to make predictions, such as the stability of the system, under various conditions.

3.1.1 Related Work

A number of researchers have considered the problem of estimating the parameters of a DDE model or pieces of the DDE model itself from time series.

Most works consider only autonomous DDEs, i.e.

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left(\boldsymbol{x}(t), \boldsymbol{x}(t-\tau_0), \ldots, \boldsymbol{x}(t-\tau_n), \boldsymbol{p}\right) \tag{3.1}$$

and rely on noisy estimates of $\boldsymbol{x}(t)$, $\boldsymbol{x}(t-\tau_0), \ldots, \boldsymbol{x}(t-\tau_n)$, and $\dot{\boldsymbol{x}}(t)$. After plotting $\left(\boldsymbol{x}(t), \boldsymbol{x}(t-\tau_0), \ldots, \boldsymbol{x}(t-\tau_n), \dot{\boldsymbol{x}}(t)\right)$ on a single set of axes, regression or other statistical methods can be used to estimate $\boldsymbol{f}$ and $\boldsymbol{p}$ or, if $\boldsymbol{f}$ is known, $\boldsymbol{p}$ [28–31]. For models of specific forms, convenient expressions involving $\boldsymbol{x}(t)$, $\boldsymbol{x}(t-\tau_0), \ldots, \boldsymbol{x}(t-\tau_n)$, and $\dot{\boldsymbol{x}}(t)$ may be plotted instead [32–36]. This technique can also be applied to DDEs with some types of dynamics noise [30, 51]. Another variation on this technique is plotting only $\left(\boldsymbol{x}(t), \boldsymbol{x}(t-\tau_0), \ldots, \boldsymbol{x}(t-\tau_n)\right)$ for times when $\dot{\boldsymbol{x}}(t) = 0$ [28, 37, 39]. When not all the state variables are observable, time lag embedding or filtering may be used to reconstruct a suitable representation of the state [34, 38]. The primary limitations of these approaches are that (1) they require directly measuring $\dot{\boldsymbol{x}}(t)$ or estimating it using numerical techniques, (2) they require measuring or reconstructing a full representation of the state, and (3) they have been developed only for autonomous DDEs or for DDEs where the only nonautonomous component is a stochastic coefficient.

In contrast, trajectory matching techniques do not have these limitations. Trajectory matching works by finding the initial conditions and parameters which produce a simulated trajectory which most closely matches the measurements. This approach can work even if the derivative cannot be estimated accurately due to noise or a large time step between measurements, only a subset of state variables are measured, and the DDE is nonautonomous, i.e.

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left(t, \boldsymbol{x}(t), \boldsymbol{x}(t-\tau_0), \ldots, \boldsymbol{x}(t-\tau_n), \boldsymbol{p}\right) \tag{3.2}$$

Trajectory matching has also been applied to ordinary differential equation (ODE) models; the primary difference when applying it to DDE models is that instead

of estimating the initial condition at a single instant in time, it is necessary to estimate $\boldsymbol{x}(t)$ for an interval of time with length equal to the largest time delay in the DDE. Early work applying trajectory matching to DDEs did not consider the problem of fitting an interval of initial states [40]. Later work by Horbelt, et al., demonstrated a multiple-shooting trajectory matching approach, where initial segments were approximated by cubic splines [41]. Deshmukh suggested a similar multiple-shooting approach, using Chebyshev spectral collocation to approximate the solution of the DDE, but did not consider observation noise [42]. Dai, et al., demonstrated trajectory matching for DDE problems but assumed that the initial interval was fully known, without noise [43]. Finally, recent work by Sysoev, et al., demonstrated a single-shooting trajectory matching approach for DDEs, with observations from a single variable, using a cubic spline representation of the initial interval [44]. These works have demonstrated the promise of trajectory matching for DDEs, but further improvements can expand the applicability and improve the accuracy of this approach.

Mann and Young introduced another optimization-based approach—matching analytically-computed characteristic multipliers (CMs) to CMs estimated from the time series—for a one-degree-of-freedom milling system [11]. The technique was successful for that problem, but further improvements are possible.

Finally, several authors have described other methods for estimating DDE parameters from time series which are not directly applicable to the problem of interest. These include methods which require actively perturbing the system [52, 53], an online adaptive synchronization method relying on noiseless measurements [54], and a method for coupled DDEs of a specific form [55].

3.1.2 Contributions

This chapter presents two novel approaches to estimate parameters for multivarate DDE models with periodic coefficients from multiple time series. Additionally, it evaluates the accuracy of the two methods in simulation.

The first approach involves matching analytical CMs computed using the spectral element method for the DDE model to CMs estimated from the time series. Unlike the existing literature, the approach described here fits multiple CM estimates per time series, instead of only the largest one, and incorporates the improved techniques described in Chapter 2 for more accurate CM estimation.

The second approach is a trajectory matching approach which provides an improvement over existing approaches in the literature. In particular, the approach introduced here handles multivarate observations and prior knowledge about parameters in a principled way, uses the spectral element method to provide a convenient representation of the initial interval and reduce the computational cost of computing the objective function, and fits multiple time series simultaneously instead of only a single time series. Fitting multiple time series improves the accuracy of the parameter estimates and is necessary to estimate parameters given time series which may not individually contain all of the necessary information.

Finally, the accuracy of the parameter estimates estimated with these approaches is evaluated by simulating an example multivariate DDE with time-periodic coefficients.

3.2 Problem

Consider a system whose dynamics follow the DDE

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}(t, \boldsymbol{p})\boldsymbol{x}(t) + \boldsymbol{B}(t, \boldsymbol{p})\boldsymbol{x}(t - T), \quad \forall t, \boldsymbol{p} \tag{3.3}$$

with time-periodic coefficients

$$\boldsymbol{A}(t,\boldsymbol{p}) = \boldsymbol{A}(t+T,\boldsymbol{p}), \quad \forall t,\boldsymbol{p} \tag{3.4}$$

$$\boldsymbol{B}(t,\boldsymbol{p}) = \boldsymbol{B}(t+T,\boldsymbol{p}), \quad \forall t,\boldsymbol{p} \tag{3.5}$$

where $\boldsymbol{p}$ is a vector of parameters. While the methods presented here can also be applied to problems where the time delay is not equal to the period, for the sake of simplicity, this chapter only considers the case when they are equal.

Assume that time series of observations of the system have been collected for various combinations of parameter values. For each time series i, a sequence of measurement vectors $\boldsymbol{z}_0^{[i]}, \boldsymbol{z}_1^{[i]}, ...$ for times $t_0^{[i]}, t_1^{[i]}, ...$ is available. For example, $t_0^{[i]}$ is the time of the first measurement in the i^{th} time series, $t_1^{[i]}$ is the time of the next measurement, etc. Each measurement $\boldsymbol{z}_j^{[i]}$ corresponds to an unknown, true state of the system $\boldsymbol{x}_j^{[i]}$ at time $t_j^{[i]}$.

Assume a known measurement model which is the same for all time series. In particular, assume that each measurement $\boldsymbol{z}_j^{[i]}$ is a known function $\boldsymbol{h}$ of the state $\boldsymbol{x}_j^{[i]}$ with additive multivariate normal noise,

$$\boldsymbol{z}_j^{[i]} = \boldsymbol{h}\left(\boldsymbol{x}_j^{[i]}\right) + \boldsymbol{\epsilon}_j^{[i]}, \quad \boldsymbol{\epsilon}_j^{[i]} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}) \tag{3.6}$$

and that the noise vectors $\boldsymbol{\epsilon}_j^{[i]}$ for all the samples are mutually independent.

The objective is to find the values of the unknown elements of the parameter vector $\boldsymbol{p}$ which best match the time series data.

3.3 Overview of the Spectral Element Method for DDEs

Both parameter fitting methods described in this chapter rely on the spectral element method for DDEs introduced by Khasawneh and Mann [10]. The spectral element method is a way to approximate solutions and compute the CMs of DDEs in a

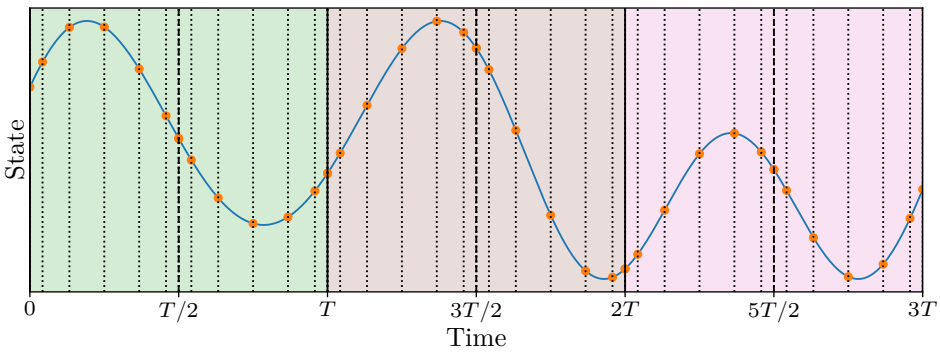


FIGURE 3.1: Illustration of the spectral element method nodes for two equally-sized elements per period and sixth order trial functions. The curve shows the true state of the system. The vertical lines indicate the locations of the nodes, with the solid lines indicating the division between periods, and the dashed lines indicating the division between the two elements within each period. The state at each node is indicated with a circular marker; the spectral element solution is the Lagrange interpolation between these nodes.

computationally efficient way. It approximates each period of the solution with interpolation between samples at discrete times $\eta_0, \ldots, \eta_\ell$, which are known as nodes. Each period is broken up into one or more elements, and the nodes in each element are chosen to be Legendre–Gauss–Lobatto (LGL) points. Interpolation between the nodes in each element is performed by Lagrange polynomials. An example of the nodes for the spectral element method with two elements per period and sixth order Lagrange polynomial trial functions is illustrated in fig. 3.1.

The spectral element method produces an approximate linear map $\boldsymbol{M}(\boldsymbol{p})$ for computing the solution at the nodes in future periods. Let $\boldsymbol{y}(t)$ be the spectral element approximation of the state $\boldsymbol{x}(t)$, then

$$\begin{bmatrix} \boldsymbol{y}\left((k+1)T+\eta_0\right) \\ \boldsymbol{y}\left((k+1)T+\eta_1\right) \\ \vdots \\ \boldsymbol{y}\left((k+1)T+\eta_\ell\right) \end{bmatrix} = \boldsymbol{M}(\boldsymbol{p}) \begin{bmatrix} \boldsymbol{y}(kT+\eta_0) \\ \boldsymbol{y}(kT+\eta_1) \\ \vdots \\ \boldsymbol{y}(kT+\eta_\ell) \end{bmatrix}, \quad \begin{array}{c} 0 \le \eta_0 \le \cdots \le \eta_\ell \le T, \\ k = 0, 1, \ldots \end{array} \tag{3.7}$$

Figure 3.1 illustrates an example mapping over multiple periods. For example, this is the mapping from the nodes in the green period $(0, \ldots, T)$ to the nodes in the brown period $(T, \ldots, 2T)$ and from the nodes in the brown period $(T, \ldots, 2T)$ to the

nodes in the pink period $(2T, \ldots, 3T)$. The matrix $\boldsymbol{M}(\boldsymbol{p})$ is an approximation of the system's true monodromy matrix. Its eigenvalues are an approximation of the system's CMs.

The spectral element approximation converges as the number of elements or number of nodes per element increases. Convergence is linear with the number of elements and exponential with the number of nodes per element [10].

3.4 Estimating Parameters by Characteristic Multiplier Matching

One approach for estimating DDE model parameters from multiple time series is to use the following two-stage process:

1. Estimate the CMs for each time series individually.

2. Fit the unknown model parameters for all time series simultaneously by minimizing the difference between the model's spectral element CMs and the CMs estimated in step 1.

The CMs for a DDE can be estimated from time series data using a method similar to the one presented in Chapter 2. The primary difference is that, in this case, state space reconstruction is necessary.

3.4.1 State Space Reconstruction

State space reconstruction can be used to convert measurements into values which are more suitable for CM estimation. Unlike finite-dimensional ODEs, DDEs have an infinite-dimensional state space. For example, for a system described by eq. (3.3), the trajectory of the system after a time t_0 depends not just on the current value $\boldsymbol{x}(t_0)$, but on the entire continuous interval $\boldsymbol{x}(t)\ \forall t \in [t_0 - T, t_0]$. Measurements of $\boldsymbol{x}$ at many discrete times in the interval can provide an approximate representation

of this infinite-dimensional state, but such a high-dimensional representation is difficult to work with numerically. State space reconstruction can be used to obtain a representation with reduced dimensionality which still contains the most important state information. It works well even if measurements include only a subset of the elements of $\boldsymbol{x}$. For example, displacement sensors may be used to measure position but not velocity.

The method used here for reconstructing time series from measurements of a subset of the state variables is based on the Karhunen–Loève transform (KLT), as described by Little, et al. [19]. Consider a time series i of noisy measurements $\boldsymbol{z}_0^{[i]}, \boldsymbol{z}_1^{[i]}, ...$ of a subset of the state variables, equally spaced in time. First, a moving average with window $n^{[i]}$ is applied to smooth the data without altering the CMs:

$$\bar{\boldsymbol{z}}_j^{[i]} = \frac{1}{n^{[i]}} \sum_{k=j}^{j+n-1} \boldsymbol{z}_k^{[i]} \tag{3.8}$$

Then, $m^{[i]}$ time-shifted copies of the smoothed data are used to embed the data into a high-dimensional pseudo phase space. The principal components $\boldsymbol{W}^{[i]}$ in the pseudo phase space are computed, and the components $\tilde{\boldsymbol{W}}^{[i]}$ with largest variance are selected. Note that it is important for $n^{[i]}$ to be large enough so that the moving average improves the signal-to-noise ratio to a level where the principal components corresponding to the transient response dominate. If the signal-to-noise ratio is too low, then it is difficult to determine a suitable number of columns for $\tilde{\boldsymbol{W}}^{[i]}$, and the selected principal components may be of low quality.

The selected components $\tilde{\boldsymbol{W}}^{[i]}$ are used to transform the time-shifted data into the reconstructed state $\boldsymbol{q}^{[i]}$:

$$\boldsymbol{q}_j^{[i]} = \left(\tilde{\boldsymbol{W}}^{[i]}\right)^\top \begin{bmatrix} \bar{\boldsymbol{z}}_j^{[i]} \\ \vdots \\ \bar{\boldsymbol{z}}_{j+m^{[i]}-1}^{[i]} \end{bmatrix} \tag{3.9}$$

The reconstructed state time series $\boldsymbol{q}_0^{[i]}, \boldsymbol{q}_1^{[i]}, ...$ is then used as the input to the CM estimation method described in Chapter 2.

3.4.2 Estimating Characteristic Multipliers from Reconstructed Time Series

The CMs can be estimated by first fitting a discrete-time affine model to the reconstructed time series using total least squares, and then computing the eigenvalues of the transformation matrix, as described in Chapter 2.

To properly weight the residuals in total least squares, it's necessary to account for the covariance of the noise in reconstructed state. First, observe that both the moving average and component extraction are linear transformations. In particular, eq. (3.8) can be expanded into eq. (3.10)

$$\begin{bmatrix} \bar{\boldsymbol{z}}_j^{[i]} \\ \vdots \\ \bar{\boldsymbol{z}}_{j+m^{[i]}-1}^{[i]} \end{bmatrix} = \boldsymbol{G}^{[i]} \begin{bmatrix} \boldsymbol{z}_j^{[i]} \\ \vdots \\ \boldsymbol{z}_{j+m^{[i]}+n^{[i]}-2}^{[i]} \end{bmatrix} \tag{3.10}$$

where

$$\boldsymbol{G}^{[i]} = \frac{1}{n^{[i]}} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{I} & \boldsymbol{I} & \cdots & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{I} & \cdots & \boldsymbol{I} & \boldsymbol{I} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{I} & \cdots & \boldsymbol{I} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \cdots & \boldsymbol{I} & \boldsymbol{I} & \boldsymbol{I} \end{bmatrix} \tag{3.11}$$

So, the overall transformation from relevant portion of the original time series to the corresponding reconstructed state vector is given by

$$\boldsymbol{q}_j^{[i]} = \left(\tilde{\boldsymbol{W}}^{[i]}\right)^\top \boldsymbol{G}^{[i]} \begin{bmatrix} \boldsymbol{z}_j^{[i]} \\ \vdots \\ \boldsymbol{z}_{j+m^{[i]}+n^{[i]}-2}^{[i]} \end{bmatrix} \tag{3.12}$$

Substituting eq. (3.6), this becomes

$$\boldsymbol{q}_j^{[i]} = \left(\tilde{\boldsymbol{W}}^{[i]}\right)^\top \boldsymbol{G}^{[i]} \begin{bmatrix} \boldsymbol{h}\left(\boldsymbol{x}_j^{[i]}\right) + \boldsymbol{\epsilon}_j^{[i]} \\ \vdots \\ \boldsymbol{h}\left(\boldsymbol{x}_{j+m^{[i]}+n^{[i]}-2}^{[i]}\right) + \boldsymbol{\epsilon}_{j+m^{[i]}+n^{[i]}-2}^{[i]} \end{bmatrix} \tag{3.13}$$

$$= \left(\tilde{\boldsymbol{W}}^{[i]}\right)^{\top} \boldsymbol{G}^{[i]} \begin{bmatrix} \boldsymbol{h}\left(\boldsymbol{x}_j^{[i]}\right) \\ \vdots \\ \boldsymbol{h}\left(\boldsymbol{x}_{j+m^{[i]}+n^{[i]}-2}^{[i]}\right) \end{bmatrix} + \boldsymbol{\xi}_j^{[i]} \tag{3.14}$$

where

$$\boldsymbol{\xi}_j^{[i]} = \left(\tilde{\boldsymbol{W}}^{[i]}\right)^{\top} \boldsymbol{G}^{[i]} \begin{bmatrix} \boldsymbol{\epsilon}_j^{[i]} \\ \vdots \\ \boldsymbol{\epsilon}_{j+m^{[i]}+n^{[i]}-2}^{[i]} \end{bmatrix} \tag{3.15}$$

So, each reconstructed state vector $\boldsymbol{q}_j^{[i]}$ has additive multivariate normal noise $\boldsymbol{\xi}_j^{[i]} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{\xi}^{[i]}})$, where

$$\boldsymbol{\Sigma}_{\boldsymbol{\xi}^{[i]}} = \left(\tilde{\boldsymbol{W}}^{[i]}\right)^{\top} \boldsymbol{G}^{[i]} \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}} & \boldsymbol{0} & \cdots \\ \boldsymbol{0} & \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}} & \ddots \\ \vdots & \ddots & \ddots \end{bmatrix} \left(\left(\tilde{\boldsymbol{W}}^{[i]}\right)^{\top} \boldsymbol{G}^{[i]}\right)^{\top} \tag{3.16}$$

Note that the noise for values of $\boldsymbol{q}^{[i]}$ close in time is not independent: each $\boldsymbol{q}_j^{[i]}$ is computed from measurements $\boldsymbol{z}_j^{[i]}, \ldots, \boldsymbol{z}_{j+m^{[i]}+n^{[i]}-2}^{[i]}$, so $\boldsymbol{q}_j^{[i]}$ shares at least one measurement with all of $\boldsymbol{q}_{j+1}^{[i]}, \ldots, \boldsymbol{q}_{j+m^{[i]}+n^{[i]}-2}^{[i]}$. As a result, when reconstructing a time series for estimating CMs, it is important to choose $m^{[i]}$ and $n^{[i]}$ small enough so that the noise is independent between the values of $\boldsymbol{q}^{[i]}$ which are actually used in total least squares for the CM estimation. Otherwise, it would be necessary to account for the correlated noise when estimating the CMs. For example, a good choice is for $m^{[i]} + n^{[i]} - 1$ to equal the number of measurements per period.

3.4.3 Estimating Parameters from Characteristic Multipliers

The model parameters can be estimated by finding the parameter values for which the CMs computed using the spectral element method most closely match the CMs estimated from the time series. This can be framed as an optimization problem

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_i d\left(B^{[i]}, g\left(\boldsymbol{\theta}, \boldsymbol{\psi}^{[i]}\right)\right) \tag{3.17}$$

where $B^{[i]}$ is the set of CM estimates for time series i; $\boldsymbol{\theta}$ is the vector of unknown model parameters, i.e. the unknown elements of $\boldsymbol{p}$; $\boldsymbol{\psi}^{[i]}$ is the vector of any known parameters for time series i; and g computes the set of approximate CMs for the parameter values using the spectral element method. This optimization problem can be solved using standard global optimization methods, such as the basin-hopping implementation with Powell's method provided by the SciPy library [49].

The function d describes the error between the CMs estimated from the time series and the CMs computed using the spectral element method for the given parameters. Since the number of CMs in each of the two sets may differ and the correspondence of the CMs between the two sets is unknown, the error is computed as the sum of the distances between the CMs for the assignment which minimizes the sum,

$$d(B,C) = \begin{cases} \min\limits_{\pi \in \Pi_{|B|,|C|}} \sum_{k=1}^{|C|} \left| b_{\pi(k)} - c_k \right| & \text{if } |C| \leq |B| \\ d(C,B) & \text{otherwise} \end{cases} \tag{3.18}$$

where $B = \left\{ b_1, \ldots, b_{|B|} \right\}$, $C = \left\{ c_1, \ldots, c_{|C|} \right\}$, and $\Pi_{|B|,|C|}$ is the set of all permutations of length $|C|$ with elements taken from $\{1, \ldots, |B|\}$. This is similar to the optimal subpattern assignment (OSPA) metric commonly used for evaluating multi-object filters [56], but without a term for the difference in cardinality between the two sets, since there are infinitely many true CMs. Equation (3.18) can be evaluated in polynomial time using the Kuhn–Munkres algorithm, a.k.a. the Hungarian algorithm [57].

3.5 Estimating Parameters by Trajectory Matching

An alternative approach for estimating the DDE model parameters is to find the parameter values for which simulated trajectories most closely match the measured time series.

3.5.1 Parameterizing Trajectories

For ODE models, any trajectory according to the model can be fully specified by the parameter values and the initial condition at a single instant in time. Analogously, for the DDE model in eq. (3.3), any trajectory can be fully specified by the parameters $\boldsymbol{p}$ and the continuous initial period of the trajectory, i.e. $\boldsymbol{x}(t) \ \forall \ t \in [0, T]$. So, the problem of matching a trajectory to time series can be described as finding the parameters and initial period which minimize the difference from the measured time series.

However, it is not possible to optimize an infinite number of unknown parameters, such as the continuous initial period; an approximate representation is necessary. The spectral element method provides a convenient approximate representation for the initial period which can then be directly used to approximate the rest of the trajectory. This representation for time series i is the state values at the nodes in the initial period:

$$\begin{bmatrix} \boldsymbol{y}^{[i]}(\eta_0) \\ \boldsymbol{y}^{[i]}(\eta_1) \\ \vdots \\ \boldsymbol{y}^{[i]}(\eta_\ell) \end{bmatrix} \tag{3.19}$$

Given this initial vector, the state at the nodes in any later period can be approximated by recursively applying eq. (3.7). Alternatively, the nodes for multiple periods can be computed in parallel via powers of $\boldsymbol{M}(\boldsymbol{p})$. The approximate state $\boldsymbol{y}^{[i]}(t_0) \approx \boldsymbol{x}^{[i]}(t_0)$ for an arbitrary time t_0 can then be computed by Lagrange interpolation between the nodes in the period containing t_0. So, this approach can be used to compute the approximate trajectory at times $t_0^{[i]}, t_1^{[i]}, ...$ for matching time series i.

Note that, in many cases, it's not necessary to represent a full initial period of *all* the state variables when fitting the model parameters. For example, consider the

following DDE:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} b_{11} & 0 \\ b_{21} & 0 \end{bmatrix} \begin{bmatrix} x_1(t-T) \\ x_2(t-T) \end{bmatrix} \tag{3.20}$$

In this case, $\dot{x}_1(t)$ and $\dot{x}_2(t)$ depend only on $x_1(t)$, $x_2(t)$, and $x_1(t-T)$; they do not depend on $x_2(t-T)$. As a result, to compute $x_1(t)$, $x_2(t)$ for $t > t_0$, it is necessary to know only $x_1(t)$ for $t \in [t_0 - T, t_0]$ and $x_2(t_0)$; it is not necessary to know $x_2(t)$ for $t \in [t_0 - T, t_0)$. So, when fitting the model parameters with trajectory matching, it would be sufficient to use only $x_1(\eta_0), \ldots, x_1(\eta_\ell), x_2(\eta_\ell)$ to represent the initial period. The only potential issue is matching the measurements in the initial period, since x_2 could be an input to the measurement function $\boldsymbol{h}$; see eq. (3.6). However, as long as no elements of the measurement vector $\boldsymbol{z}$ depend on both x_1 and x_2, replacing $x_2(\eta_0), \ldots, x_2(\eta_{\ell-1})$ with constants, such as zeros, does not effect which model parameter values minimize the objective function. Another strategy would be to represent the period *before* the start of the time series in order to avoid having to match measurements for that period, but, in that case, it would be more difficult to make a good initial guess for the optimization.

3.5.2 Fitting the Parameters

The objective is to find the vector of unknown model parameters and initial conditions which maximizes the posterior density $p\left(\boldsymbol{\theta} \mid \boldsymbol{z}_1^{[1]}, \ldots\right)$ conditional on all the time series measurements. The $\boldsymbol{\theta}$ vector contains the unknown elements of $\boldsymbol{p}$ and the representation of the initial period for each time series i, as described in Section 3.5.1. The prior belief for $\boldsymbol{\theta}$ is assumed to be a multivariate normal density, $\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}})$. As described in Appendix C, this optimization problem can be

expressed as minimizing a sum of squares:

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left(\sum_{i,j} \left(\boldsymbol{r}_j^{[i]} \right)^\top \boldsymbol{r}_j^{[i]} + \left(\boldsymbol{r}_{\boldsymbol{\theta}} \right)^\top \boldsymbol{r}_{\boldsymbol{\theta}} \right) \tag{3.21}$$

where

$$\boldsymbol{r}_j^{[i]} = \boldsymbol{L}_{\boldsymbol{\epsilon}}^\top \left(\boldsymbol{z}_j^{[i]} - \boldsymbol{h}\left(\boldsymbol{y}_j^{[i]} \right) \right) \tag{3.22}$$

$$\boldsymbol{r}_{\boldsymbol{\theta}} = \boldsymbol{L}_{\boldsymbol{\theta}}^\top \left(\boldsymbol{\theta} - \boldsymbol{\mu}_{\boldsymbol{\theta}} \right) \tag{3.23}$$

and $\boldsymbol{L}_{\boldsymbol{\epsilon}}$, $\boldsymbol{L}_{\boldsymbol{\theta}}$ are factors in the Cholesky decompositions $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}^{-1} = \boldsymbol{L}_{\boldsymbol{\epsilon}} \boldsymbol{L}_{\boldsymbol{\epsilon}}^\top$, $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1} = \boldsymbol{L}_{\boldsymbol{\theta}} \boldsymbol{L}_{\boldsymbol{\theta}}^\top$.

This problem can be solved using standard nonlinear least-squares optimization methods such as the Levenberg–Marquardt method. However, observe that only a subset of the parameters in $\boldsymbol{\theta}$ affect each term in the sum. For example, the approximate initial period for one time series does not affect the approximate solution for any other time series. This observation can be exploited to significantly reduce the computational cost of optimization. For this work, the `scipy.optimize.least_squares` SciPy library function [49] with the Trust Region Reflective algorithm and sparse Jacobians was used to solve the optimization problem.

One remaining consideration is how to guess an initial value of $\boldsymbol{\theta}$ for the optimization algorithm. In many cases, a guess for the representations of the initial periods can be obtained with ordinary least squares, as described in Appendix D. When that is not possible, zeros could be used instead, or the values representing the initial period could be optimized separately to fit the initial period of measurements. Guessing the initial model parameters requires some prior knowledge. For example, the model parameters can be sampled from the prior distribution or simply sampled uniformly over a reasonable range. Note that least squares optimization algorithms typically find only local optima, so it can be beneficial to run the optimization algorithm multiple times, for different randomly sampled guesses, so that it is more likely that at least one guess will be in the same basin as the global optimum.

3.6 Results and Discussion

The proposed methods were tested by evaluating their performance for the following simulated multivariate DDE model and initial conditions [10, 58]:

$$\ddot{u} + 2\zeta\dot{u} + (1 - \delta\cos 2\Omega t)u - \delta\sin(2\Omega t)v = bu(t - T) \tag{3.24}$$

$$\ddot{v} + 2\zeta\dot{v} + (1 + \delta\cos 2\Omega t)v - \delta\sin(2\Omega t)u = bv(t - T) \tag{3.25}$$

$$u(s) = \sin(2\Omega s), \quad \text{for } s \in [-T, 0] \tag{3.26}$$

$$v(s) = \cos(2\Omega s), \quad \text{for } s \in [-T, 0] \tag{3.27}$$

Equations (3.24) and (3.25) are described by eq. (3.3), with

$$\boldsymbol{x}(t) = \begin{bmatrix} u(t) \\ v(t) \\ \dot{u}(t) \\ \dot{v}(t) \end{bmatrix} \tag{3.28}$$

$$\boldsymbol{A}(t, \boldsymbol{p}) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -(1 - \delta\cos 2\Omega t) & \delta\sin 2\Omega t & -2\zeta & 0 \\ \delta\sin 2\Omega t & -(1 + \delta\cos 2\Omega t) & 0 & -2\zeta \end{bmatrix} \tag{3.29}$$

$$\boldsymbol{B}(t, \boldsymbol{p}) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ b & 0 & 0 & 0 \\ 0 & b & 0 & 0 \end{bmatrix} \tag{3.30}$$

The variables u and v represent non-dimensionalized deflections of the midspan of an asymmetric shaft with delayed feedback, where ζ is the damping, δ describes the asymmetry, b is a feedback gain, Ω is the rotational frequency, and $T = \pi/\Omega$ is the delay in the feedback [10].

The system was simulated for parameter values $\zeta = 0.02$, $\delta = 0.3$, $b = 0.04$, and four different frequencies $\Omega \in \{0.4, 0.8, 1.2, 1.3\}$. The stable region in (Ω, δ) parameter space corresponding to these values of ζ and b is illustrated by the shaded region in fig. 3.2. Each of the four time series consisted of 25 periods of length

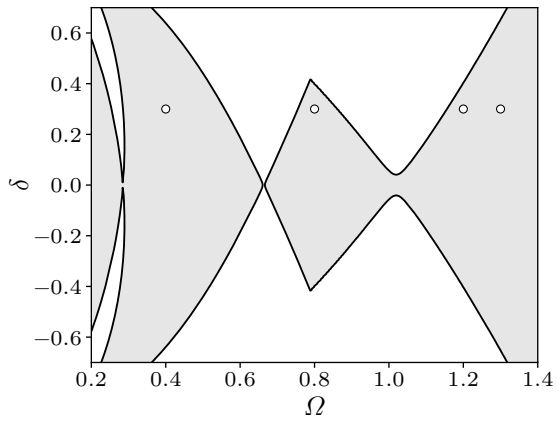


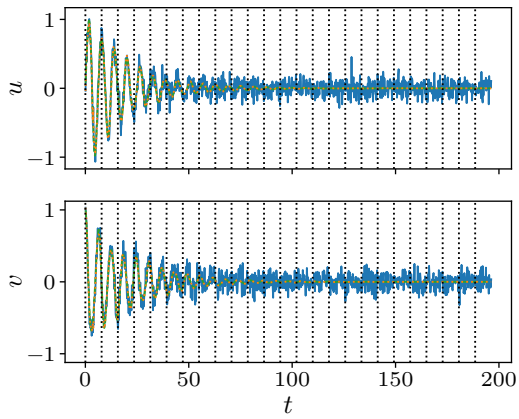
FIGURE 3.2: Stability chart for the true parameter values, $\zeta = 0.02$ and $b = 0.04$. The shaded region indicates where the system is stable. The values of Ω and δ for the simulated time series are shown with circular markers.

T, with 40 uniformly-spaced measurements per period. Measurements consisted of noisy measurements of the deflections:

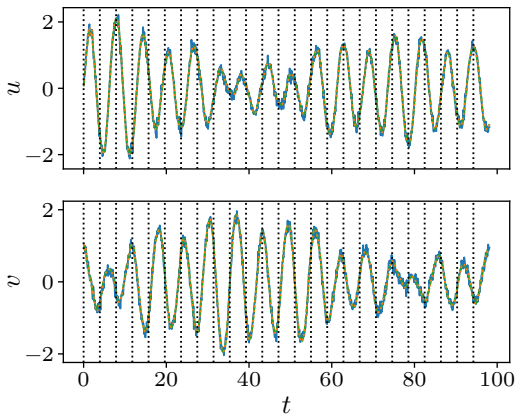
$$\boldsymbol{z}_j^{[i]} = \begin{bmatrix} u_j^{[i]} \\ v_j^{[i]} \end{bmatrix} + \boldsymbol{\epsilon}_j^{[i]}, \quad \boldsymbol{\epsilon}_j^{[i]} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix} \right) \tag{3.31}$$

with $\sigma = 10^{-1}$. Note, in particular, that the measurements did not depend on the velocities $\dot{u}$, $\dot{v}$. Examples of the noisy time series are shown in fig. 3.3, and the locations of the corresponding parameter values on the stability chart are shown in fig. 3.2.

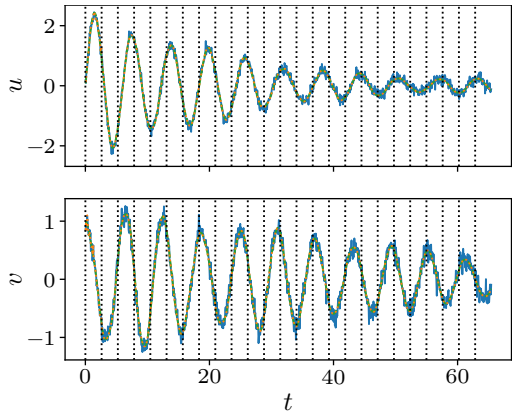
It was assumed that Ω was always known, so the parameters to fit were $(\log \zeta)$, δ, and b. With unconstrained optimization, fitting $(\log \zeta)$ instead of fitting ζ directly ensures that $\zeta = \mathrm{e}^{\log \zeta}$ is positive; additionally, working in log space is beneficial when the order of magnitude of ζ is uncertain. The initial parameter guesses for the optimizers were uniformly distributed in the ranges $(\log \zeta) \in [-3, -1]$, $\delta \in [-0.7, 0.7]$, and $b \in [-0.2, 0.2]$. For both methods, when guessing initial parameter values, 1000 guesses were considered, and the best guess was used to initialize the optimizer. For the CM matching method, the unknown parameter vector $\boldsymbol{\theta}$ consisted of $(\log \zeta)$,



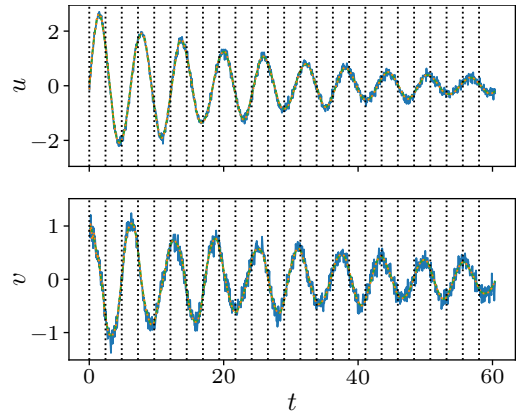
(a) Time series 0.



(b) Time series 1.



(c) Time series 2.



(d) Time series 3.

FIGURE 3.3: Time series of the true state, example measurements, and corresponding fitted time series. The noisy observations are indicated with a solid blue line; the true state values are indicated with a green dashed line; and the fitted state values are indicated with an orange dotted line. The start of each period of length T is indicated with a vertical dotted line.

δ, and b. For the trajectory matching method, the unknown parameter vector $\boldsymbol{\theta}$ consisted of $(\log \zeta)$, δ, b, and the relevant state values in the initial period for each time series i: $u^{[i]}(\eta_0), \ldots, u^{[i]}(\eta_\ell)$, $v^{[i]}(\eta_0), \ldots, v^{[i]}(\eta_\ell)$, $\dot{u}^{[i]}(\eta_\ell)$, $\dot{v}^{[i]}(\eta_\ell)$. The guesses for the initial periods were computed using the method described in Appendix D. The spectral element method was order 10, with 2 elements per period T.

For the CM matching method, state space reconstruction used $n = 13$, i.e. 1/3 of

the number of measurements per period, and $m = 26$, i.e. 2/3 of the number of measurements per period. The threshold for selecting the dominant components, i.e. the number of columns of $\tilde{\boldsymbol{W}}$, was 99% of the variance (to avoid spurious components) or 4 components (to keep the number of parameters for total least squares CM estimation less than the number of periods), whichever was lower. The basin-hopping parameter optimizer for matching the CMs used 30 iterations.

For the trajectory matching method, the optimizer was run 15 times with different initial guesses, and the optimized parameter set with the lowest cost was selected. The prior density had $\boldsymbol{\mu_\theta} = \mathbf{0}$, $\boldsymbol{\Sigma_\theta^{-1}} = \mathbf{0}$, i.e. no prior knowledge about the parameters.

Figure 3.4 shows the distributions of the fitted parameters computed using the two methods, for 50 different seeds for the random observation noise and initial parameter guesses. The CM matching method produced estimates which were reasonably close to the true values. The only exception was δ, for which this method correctly estimated the absolute value but not the sign: roughly half of the samples resulted in a negative estimate of δ. It turns out that, due to symmetries in the system, the CMs are identical regardless of the sign of δ; thus, it is not possible to determine the sign of δ from the CMs. In comparison to the CM matching method, the trajectory matching method produces more consistently accurate estimates of the parameters with less variation.

Figure 3.5 shows stability charts for the parameters estimated using the two methods. The color at a point indicates the fraction of the 50 sampled cases for which the point was stable for the estimated parameter values. It is clear that while the parameter values estimated via the CM matching method produce stability estimates which are reasonably close to reality, the location of estimated stability boundary is quite variable in some places, such as near $\Omega = 0.66$, $\delta = 0$. The parameter values estimated using trajectory matching produced stability estimates which were

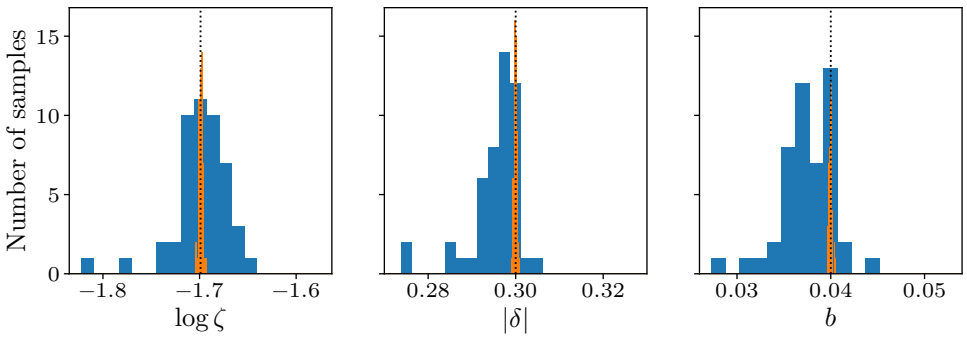
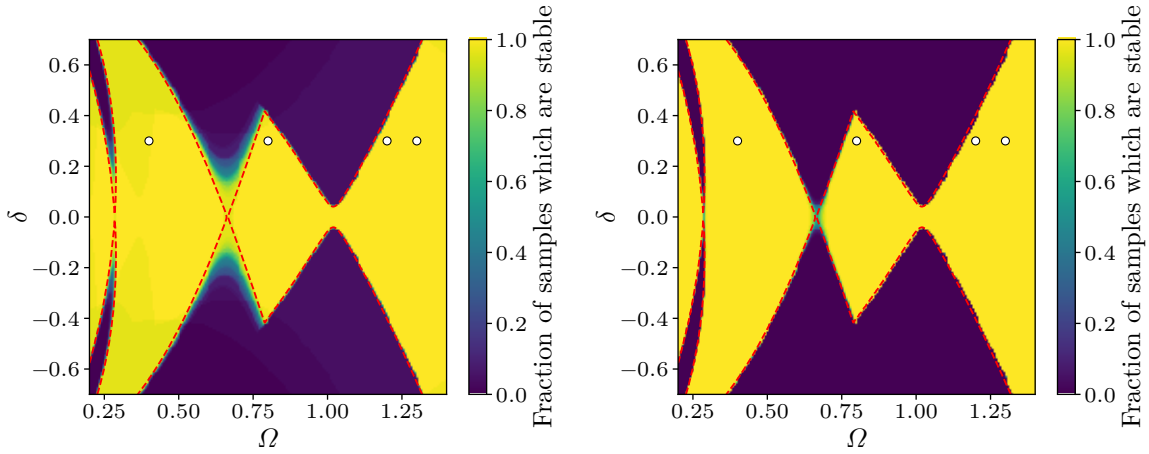


FIGURE 3.4: Histograms of the fitted parameter values for the two methods. The CM matching histogram has a blue background, and the trajectory matching histogram has an orange background. The correct parameter value is indicated with a dotted vertical line. The distributions were obtained by using 50 different seeds for the measurement noise and initial parameter guesses. Note that the absolute value of δ is plotted, instead of δ itself, since about half of the δ estimates using CM matching were negative. (In contrast, the estimates for δ using trajectory matching were all positive.) Note also that two, i.e. 4%, of the samples for the CM matching method were outliers, outside the axis limits, likely due to failure of the optimizer to find the global optimum.

variable in the same areas, but with much less overall variation. In general, the stability boundaries computed from the parameter values estimated using trajectory matching were very close to the true stability boundary.

For the trajectory matching method, it is also worthwhile to look at the difference between the fitted time series and the true state values. Figure 3.3 shows the fitted time series overlaid on the time series of the true state and measurements for one sample case, and fig. 3.6 shows the difference of the fitted time series and the measurements from the true states. The fitted time series are quite close to the true state values; the primary exception is in the first period, where the fitted time series matches the noisy measurements more closely than the true state in some places. This is unsurprising, especially when the initial period can be selected to fit the measurements more closely without significantly affecting the predictions for later periods. However, if a more accurate estimate of the initial period is desired, it may be beneficial to choose a prior belief on the representation of the initial period which has greater covariance for nodes nearby in time. For example, a good choice if the



(a) CM matching.

(b) Trajectory matching.

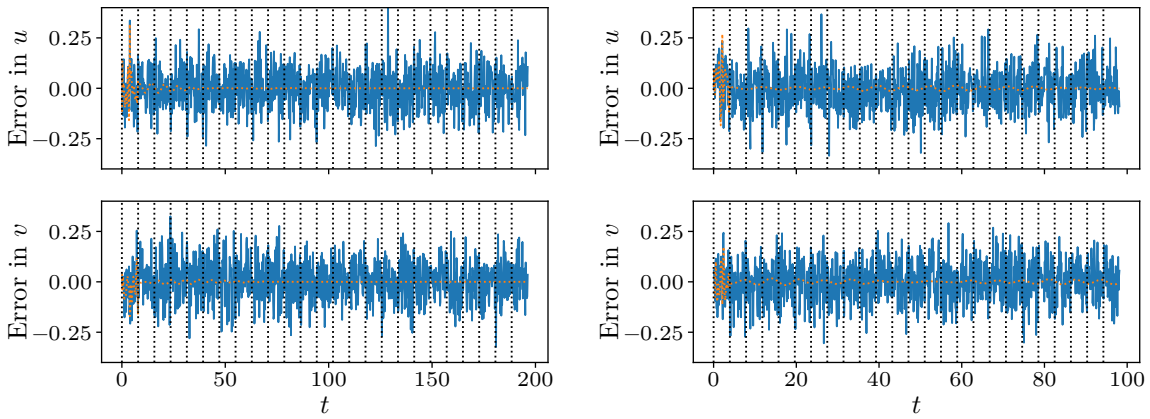
FIGURE 3.5: Stability charts for the estimated parameter values. The color indicates the fraction of the 50 sampled cases for which the point was stable for the fitted parameter values. The true stability boundary is indicated with a red dashed line. The parameter values for the time series are shown with circular markers.

initial period is assumed to be smooth would be the squared exponential covariance commonly used in Gaussian process regression [59].

In summary, both parameter estimation methods work fairly well for this example DDE, but the trajectory matching method produces more consistently accurate parameter estimates and stability boundaries and, unlike CM matching, can correctly identify the sign of δ.

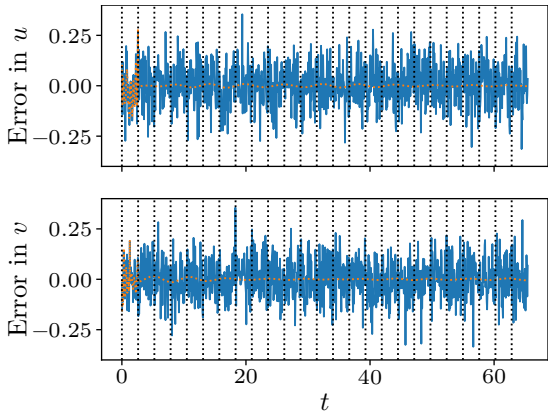
3.7 Conclusion

This chapter presents two novel approaches to estimate parameters for multivariate DDE models with periodic coefficients from multiple time series. It then evaluates the quality of their estimates for an example system. The first approach, CM matching, works fairly well, but its estimates have more variability than would be ideal, and it cannot distinguish between different parameter values which produce the same CMs. For example, for the test problem, CM matching could not identify the sign of the

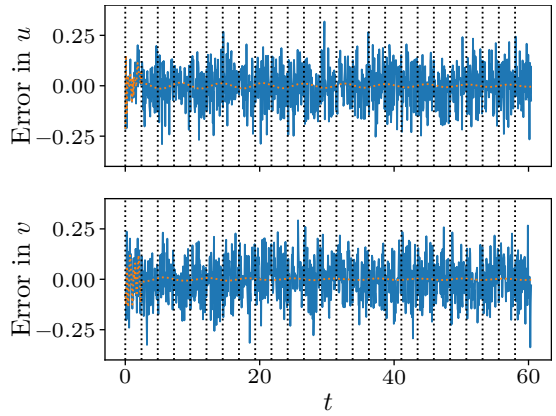


(a) Time series 0.

(b) Time series 1.



(c) Time series 2.



(d) Time series 3.

FIGURE 3.6: Error, i.e. the difference from the true state values, in the measurements (blue line) and corresponding time series fitted using trajectory matching (dotted orange line) for one sample case. The start of each period of length T is indicated with a vertical dotted line.

δ parameter, since the CMs were symmetric about $\delta = 0$. The second approach, trajectory matching, produces consistently accurate parameter estimates which also produce stability charts which closely match the true stability boundary. In addition to the improved accuracy and correctly determining the sign of δ, the trajectory matching method can more easily handle measurements at arbitrary times than the CM matching method. One problematic case for the trajectory matching method described here is fitting chaotic behavior, due to sensitivity to initial conditions; in

this case, it may be better to use trajectory matching with multiple-shooting instead of single-shooting to limit the length of segments being fit. For future investigation, the parameter estimates from the CM matching method could possibly be improved by matching only the largest CMs estimated from the time series, which tend to be more accurate, or by modifying the objective function to account for the uncertainty of the CM estimates in a principled way. Regardless, both methods presented here worked well for the example problem.

4

Empirical Stability Prediction for Milling

4.1 Introduction

Milling is one of the most commonly used manufacturing processes. The milling machine rapidly spins a cutting tool and gradually feeds it into the workpiece. As the teeth on the tool contact the workpiece, they remove small chips of material, with the goal of producing a smooth surface at the desired location. However, vibrations of the cutting tool can cause variation in the chip thickness, a poor surface finish, and reduced accuracy of the machined surface. Due to the relative motion between the tool and workpiece, each tooth passage affects the cutting forces and chip thickness during the next tooth passage. For certain combinations of cutting parameters, feedback between sequential tooth passages can cause unstable motion of the tool, known as chatter. The large forces in chatter can damage the tool and produce a poorly machined, inaccurate surface. Thus, it would be very useful to be able predict when chatter will occur.

Researchers have developed dynamics models for milling and methods using these models to predict stability, e.g. [21, 60]. However, these models require estimates

of parameter values which vary between every combination of cutting tool, tool holder, spindle, machine, workpiece material, etc. [61, 62]. Measuring the parameters directly requires time-consuming tests with a skilled engineer and specialized equipment [61–64]. It would be much more useful to be able to, within a few minutes, make a few test cuts with the milling machine itself to collect data, and then run an algorithm to generate an estimated stability chart from the data. This chapter presents a method to accomplish this. The proposed method is based on the trajectory matching approach described in Chapter 3, but it is specifically adapted to the problem of milling.

4.1.1 Related Work

Early work on the vibration of machine tools provided models to explain chatter, including the development of stability charts which compactly represent stability as a function of the system parameters, such as the spindle speed and cutting depth, and identifying regions of stability where material could be removed more quickly by cutting at higher spindle speeds [65–67]. Related works expanding on those efforts have investigated refinements in the models, experimental validation, and improvements in the analysis techniques [60, 65–84]. The physics-based models which describe the dynamics of machining are typically written as delay differential equations (DDEs) [11, 20–27]. Researchers have investigated methods to analytically or numerically approximate the solutions of these DDEs, primarily with a focus on stability [8, 9, 21, 60, 85–90]. Researchers have also worked on methods for optimization of machining parameters, e.g. [91, 92]. Although researchers have developed sophisticated models and analysis techniques for vibration of machine tools, the practical applicability of this knowledge has been limited. In particular, a significant limitation is the need for model parameters which accurately describe the structural dynamics and cutting forces; obtaining these parameters is time-consuming, difficult, and often requires

the expertise of a skilled engineer. This work was inspired by [11], which sought to estimate parameters for a one-degree-of-freedom milling model from vibration time series collected during a milling process. This chapter expands on that work by estimating parameters for a two-degree-of-freedom milling model based on [21, 60]; it presents a new trajectory-matching approach specifically designed for time series collected from an instrumented milling machine.

4.1.2 Contributions

For the first time, this chapter presents a method to estimate the parameters for a two-degree-of-freedom DDE milling model directly from time series collected on an instrumented milling machine. It combines and extends models from [21, 60] into an updated model which incorporates steady-state vibration of the tool outside of cutting, due to effects such as mass imbalance or misalignment of the cutting tool in the tool holder. Appendix E adapts and extends the spectral element method [10] to efficiently predict the trajectory of the tool using this model. Finally, this chapter validates the proposed methods using experimental data and evaluates the quality of the stability predictions.

4.2 Dynamics Model

Figure 4.1 provides a schematic of the down-milling process. As the cutting tool rotates and is fed into the workpiece, it removes material from the workpiece. The tool also bends and vibrates due to the cutting forces. Considering only the first mode of vibration of the tool in the X and Y directions of fig. 4.1, the 2-D vibration of the tool tip in the X and Y directions can be modeled by [60]:

$$\boldsymbol{M}\ddot{\boldsymbol{q}}(t) + \boldsymbol{C}\dot{\boldsymbol{q}}(t) + \boldsymbol{K}\boldsymbol{q}(t) = \boldsymbol{f}_{\mathrm{c}}(t) \tag{4.1}$$

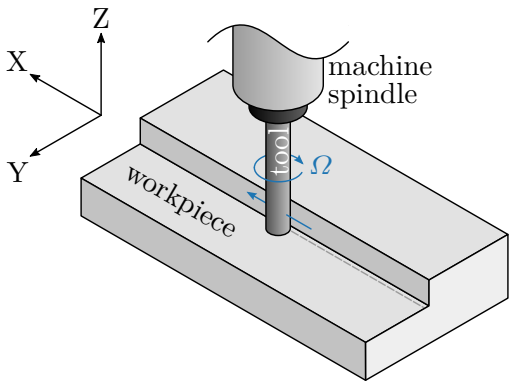


FIGURE 4.1: Schematic of down-milling with an end mill. The directions of the rotation and feed of the tool are indicated with blue arrows.

where the two components of $\boldsymbol{q}(t) = \begin{bmatrix} q_{\mathrm{X}}(t) & q_{\mathrm{Y}}(t) \end{bmatrix}^{\top}$ are the displacements of the tool tip in the X and Y directions, respectively, relative to the axis of rotation, at time t; $\boldsymbol{M}$, $\boldsymbol{C}$, and $\boldsymbol{K}$ are mass, damping, and stiffness matrices of the tool; and $\boldsymbol{f}_{\mathrm{c}}(t)$ is the vector of cutting forces in the X and Y directions exerted by the workpiece on the tool near the tool tip at time t.

The cutting forces can be modeled by [21, 60]:

$$\boldsymbol{f}_{\mathrm{c}}(t) = \boldsymbol{K}_{\mathrm{c}}(t) \left(\boldsymbol{q}(t) - \boldsymbol{q}(t-\tau) \right) + \boldsymbol{f}_0(t) \tag{4.2}$$

where $\boldsymbol{K}_{\mathrm{c}}(t)$ is a time-periodic coefficient for cutting forces which depend on the variation in the chip thickness; $\boldsymbol{f}_0(t)$ is a time-periodic vector of additional cutting forces; and τ is the tooth passing period. The value of τ is given by $\tau = 2\pi\Omega^{-1}n_{\mathrm{teeth}}^{-1}$, where Ω is the spindle speed in rad/s, and n_{teeth} is the number of teeth on the tool. Section 4.2.1 discusses the cutting forces in more detail.

Equation (4.1) assumes that the steady-state position of the tool in the absence of cutting forces is perfectly aligned to and centered about the axis of rotation. However, this assumption may not be valid. For example, the rotating components, such as the tool and tool holder, may have a mass imbalance which can cause vibration of the

tool; the tool could also be imperfectly positioned in the tool holder. These effects can be approximated by an additional, time-periodic, term $\boldsymbol{f}_{\mathrm{u}}(t)$:

$$\boldsymbol{M}\ddot{\boldsymbol{q}}(t) + \boldsymbol{C}\boldsymbol{q}(t) + \boldsymbol{K}\boldsymbol{q}(t) = \boldsymbol{f}_{\mathrm{c}}(t) + \boldsymbol{f}_{\mathrm{u}}(t) \tag{4.3}$$

Substituting eq. (4.2) and rearranging the terms, eq. (4.3) can be written as a first-order DDE:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}(t, \boldsymbol{p})\boldsymbol{x}(t) + \boldsymbol{B}(t, \boldsymbol{p})\boldsymbol{x}(t - \tau) + \boldsymbol{c}(t, \boldsymbol{p}) \tag{4.4}$$

where

$$\boldsymbol{x}(t) = \begin{bmatrix} \boldsymbol{q}(t) \\ \dot{\boldsymbol{q}}(t) \end{bmatrix} \tag{4.5}$$

$$\boldsymbol{A}(t, \boldsymbol{p}) = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{M}^{-1}\boldsymbol{K} + \boldsymbol{M}^{-1}\boldsymbol{K}_{\mathrm{c}}(t) & -\boldsymbol{M}^{-1}\boldsymbol{C} \end{bmatrix} \tag{4.6}$$

$$\boldsymbol{B}(t, \boldsymbol{p}) = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ -\boldsymbol{M}^{-1}\boldsymbol{K}_{\mathrm{c}}(t) & \boldsymbol{0} \end{bmatrix} \tag{4.7}$$

$$\boldsymbol{c}(t, \boldsymbol{p}) = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{M}^{-1}\boldsymbol{f}_0(t) + \boldsymbol{M}^{-1}\boldsymbol{f}_{\mathrm{u}}(t) \end{bmatrix} \tag{4.8}$$

The $\boldsymbol{f}_{\mathrm{u}}(t)$ term is modeled by:

$$\boldsymbol{f}_{\mathrm{u}}(t) = \boldsymbol{f}_{\mathrm{u},0} + \mathcal{R}\left[\boldsymbol{f}_{\mathrm{u,p}} \mathrm{e}^{\mathrm{i}\Omega t}\right] \tag{4.9}$$

where $\boldsymbol{f}_{\mathrm{u},0}$ is a constant, real vector, where the subscript '0' refers to this being the constant term; $\boldsymbol{f}_{\mathrm{u,p}}$ is a constant, complex vector, where the subscript 'p' refers to this being the coefficient of the periodic term; and $\mathcal{R} : \mathbb{C} \to \mathbb{R}$ extracts the real part of its argument. Note that $\boldsymbol{f}_{\mathrm{u}}(t)$ affects the trajectory of the system, so it is important for fitting time series, but it does not affect the characteristic multipliers (CMs) or stability. This can be seen most easily using the spectral element method; see Section 4.2.3 and Appendix E.

4.2.1 Cutting Forces

The derivation presented here of the cutting forces for a helical cutting tool is similar to [21], but this derivation provides simplified expressions for $\boldsymbol{K}_{\mathrm{c}}(t)$ and $\boldsymbol{f}_0(t)$, and it incorporates edge coefficients as in [60]. The cutting forces are distributed along a small contact region, as shown in fig. 4.2a. The total cutting forces on the tool in the X and Y directions can be computed by summing over the teeth and integrating over the contact region for each tooth:

$$\begin{aligned} &\boldsymbol{f}_{\mathrm{c}}(t) \\ &= \sum_{k=1}^{n_{\mathrm{teeth}}} g_k(t) \int_{z_{k,\mathrm{lo}}(t)}^{z_{k,\mathrm{hi}}(t)} \left(\begin{bmatrix} -\cos\theta_k(z,t) \\ \sin\theta_k(z,t) \end{bmatrix} f_{\mathrm{t}}(z,t) + \begin{bmatrix} -\sin\theta_k(z,t) \\ -\cos\theta_k(z,t) \end{bmatrix} f_{\mathrm{n}}(z,t) \right) \mathrm{d}z \end{aligned} \tag{4.10}$$

where $f_{\mathrm{n}}(z,t)$ and $f_{\mathrm{t}}(z,t)$ are the normal and tangential cutting forces per unit height at height z and time t, as illustrated in fig. 4.2b; $z_{k,\mathrm{lo}}(t)$ is the lowest position where tooth k is in contact with the workpiece; $z_{k,\mathrm{hi}}(t)$ is the highest position where tooth k is in contact with the workpiece; $g_k(t)$ is 1 if tooth k is cutting, and 0 otherwise; and the instantaneous angle of the point at height z on tooth k at time t is given by

$$\theta_k(z,t) = \theta_k(z,t_0) + \Omega \cdot (t - t_0) \tag{4.11}$$

For points in the contact region, the radial chip thickness can be approximated by

$$w_k(z,t) = h \sin\theta_k(z,t) + \begin{bmatrix} \sin\theta_k(z,t) & \cos\theta_k(z,t) \end{bmatrix} (\boldsymbol{q}(t) - \boldsymbol{q}(t-\tau)) \tag{4.12}$$

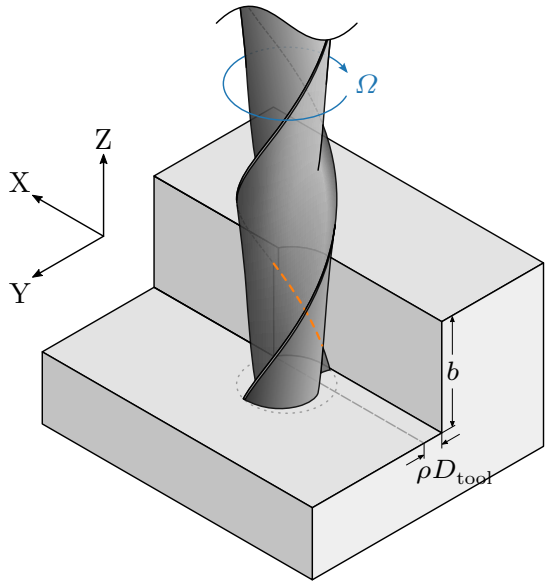
where h is the feed per tooth. The cutting forces per unit height can be modeled by

$$f_{\mathrm{t}}(z,t) = K_{\mathrm{t}} w_k(z,t) + K_{\mathrm{te}} \tag{4.13}$$

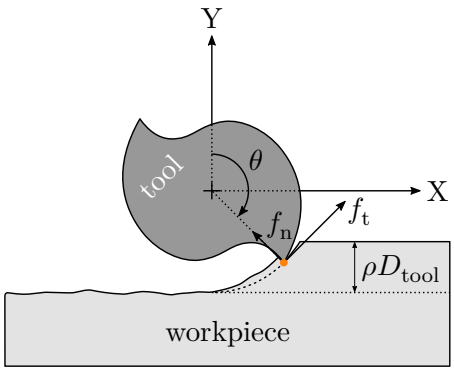
$$f_{\mathrm{n}}(z,t) = K_{\mathrm{n}} w_k(z,t) + K_{\mathrm{ne}} \tag{4.14}$$

where K_{t} and K_{n} are the cutting pressures, and K_{te} and K_{ne} are the edge coefficients. Substituting and rearranging, eq. (4.10) becomes eq. (4.2), with

$$\boldsymbol{K}_{\mathrm{c}}(t) = \sum_{k=1}^{n_{\mathrm{teeth}}} g_k(t) \int_{z_{k,\mathrm{lo}}(t)}^{z_{k,\mathrm{hi}}(t)} \begin{bmatrix} -K_{\mathrm{t}} sc - K_{\mathrm{n}} s^2 & -K_{\mathrm{t}} c^2 - K_{\mathrm{n}} sc \\ K_{\mathrm{t}} s^2 - K_{\mathrm{n}} sc & K_{\mathrm{t}} sc - K_{\mathrm{n}} c^2 \end{bmatrix} \mathrm{d}z \tag{4.15}$$



(a) Isometric view of the tool and workpiece. The contact region between the tooth and the workpiece is indicated with a dashed orange line.



(b) Cross section of the tool and workpiece, viewed from above. The arrows labeled f_t and f_n show the tangential and normal cutting forces per unit height at this height and instant in time.

FIGURE 4.2: Illustrations of the contact region and the cutting forces per unit height for a helical cutting tool with two teeth.

$$\boldsymbol{f}_0(t) = \sum_{k=1}^{n_\mathrm{teeth}} g_k(t) \int_{z_{k,\mathrm{lo}}(t)}^{z_{k,\mathrm{hi}}(t)} \left(h \begin{bmatrix} -K_\mathrm{t} sc - K_\mathrm{n} s^2 \\ K_\mathrm{t} s^2 - K_\mathrm{n} sc \end{bmatrix} + \begin{bmatrix} -K_\mathrm{te} c - K_\mathrm{ne} s \\ K_\mathrm{te} s - K_\mathrm{ne} c \end{bmatrix} \right) \mathrm{d}z \quad (4.16)$$

where

$$s = \sin\theta_k(z,t) \qquad\qquad c = \cos\theta_k(z,t) \qquad\qquad (4.17)$$

To evaluate eqs. (4.15) and (4.16), it is necessary to compute $g_k(t)$ and the bounds of integration. For down-milling, the leading tip of a tooth enters the cut at angle $\theta_\mathrm{en} = \pi - \arccos(1-2\rho)$ and exits the cut at angle $\theta_\mathrm{ex} = \pi$, where the radial immersion is ρD_tool, and D_tool is the diameter of the tool. Note, however, that for a tool with a helix angle $\beta \neq 0$, the upper contact point of the tooth does not start cutting until a short time after the tip starts cutting, and it does not exit the cut

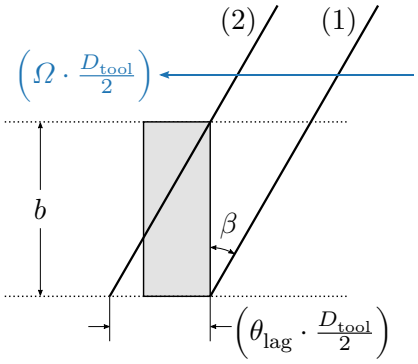


FIGURE 4.3: Illustration of the helix and lag angles. The view is looking at the side of the tool from the point of view of the workpiece, with the helix unrolled onto a plane. The thick diagonal lines are the helical edge of the tooth (1) when the tip of the tooth starts cutting and (2) when the point at height b on the tooth starts cutting. The tool rotates through an angle of θ_{lag} between instants (1) and (2). The shaded rectangle indicates the size of the chip being cut on this tooth passage; the size varies with the axial cut depth b, which determines the height, and the radial immersion, which affects the width. At any instant during the cut, the contact between the tooth and the workpiece is the intersection between a diagonal line representing the tooth edge and the shaded rectangle.

until a short time after the tip of the tooth exits the cut. Let

$$\kappa = \frac{\tan \beta}{D_{\text{tool}}/2} \tag{4.18}$$

so that

$$\theta_k(z,t) = \theta_k(0,t) - \kappa z \tag{4.19}$$

Then, for axial cut depth b, the tool rotates through an angle of $\theta_{\text{lag}} = \kappa b$ between the tip of a tooth entering the cut and the upper contact point on the tooth entering the cut, and between the tip of a tooth exiting the cut and the upper contact point on the tooth exiting the cut. This is illustrated in fig. 4.3. So, the contact indicator function for tooth k is

$$g_k(t) = \begin{cases} 1 & \text{if } \theta_k(0,t) \in [\theta_{\text{en}}, \theta_{\text{ex}} + \theta_{\text{lag}}] \\ 0 & \text{otherwise} \end{cases} \tag{4.20}$$

The heights of the instantaneous lower and upper points of contact between tooth k

and the workpiece are given by

$$z_{k,\mathrm{lo}}(t) = \begin{cases} 0 & \text{if } \theta_k(0,t) \in [\theta_{\mathrm{en}}, \theta_{\mathrm{ex}}] \\ -\kappa^{-1}\left(\theta_{\mathrm{ex}} - \theta_k(0,t)\right) & \text{if } \theta_k(0,t) \in \left[\theta_{\mathrm{ex}}, \theta_{\mathrm{ex}} + \theta_{\mathrm{lag}}\right] \end{cases} \tag{4.21}$$

$$z_{k,\mathrm{hi}}(t) = \begin{cases} -\kappa^{-1}\left(\theta_{\mathrm{en}} - \theta_k(0,t)\right) & \text{if } \theta_k(0,t) \in \left[\theta_{\mathrm{en}}, \theta_{\mathrm{en}} + \theta_{\mathrm{lag}}\right] \\ b & \text{if } \theta_k(0,t) \in \left[\theta_{\mathrm{en}} + \theta_{\mathrm{lag}}, \theta_{\mathrm{ex}} + \theta_{\mathrm{lag}}\right] \end{cases} \tag{4.22}$$

and the corresponding angles for the lower and upper points of contact are

$$\theta_{k,\mathrm{lo}}(t) = \theta_k\left(z_{k,\mathrm{lo}}(t), t\right) \tag{4.23}$$

$$= \begin{cases} \theta_k(0,t) & \text{if } \theta_k(0,t) \in [\theta_{\mathrm{en}}, \theta_{\mathrm{ex}}] \\ \theta_{\mathrm{ex}} & \text{if } \theta_k(0,t) \in \left[\theta_{\mathrm{ex}}, \theta_{\mathrm{ex}} + \theta_{\mathrm{lag}}\right] \end{cases} \tag{4.24}$$

$$\theta_{k,\mathrm{hi}}(t) = \theta_k\left(z_{k,\mathrm{hi}}(t), t\right) \tag{4.25}$$

$$= \begin{cases} \theta_{\mathrm{en}} & \text{if } \theta_k(0,t) \in \left[\theta_{\mathrm{en}}, \theta_{\mathrm{en}} + \theta_{\mathrm{lag}}\right] \\ \theta_k(0,t) - \theta_{\mathrm{lag}} & \text{if } \theta_k(0,t) \in \left[\theta_{\mathrm{en}} + \theta_{\mathrm{lag}}, \theta_{\mathrm{ex}} + \theta_{\mathrm{lag}}\right] \end{cases} \tag{4.26}$$

Then, eqs. (4.15) and (4.16) can be simplified to

$$\boldsymbol{K}_{\mathrm{c}}(t) = \sum_{k=1}^{n_{\mathrm{teeth}}} g_k(t) \begin{bmatrix} -K_{\mathrm{t}}\overline{sc} - K_{\mathrm{n}}\overline{s^2} & -K_{\mathrm{t}}\overline{c^2} - K_{\mathrm{n}}\overline{sc} \\ K_{\mathrm{t}}\overline{s^2} - K_{\mathrm{n}}\overline{sc} & K_{\mathrm{t}}\overline{sc} - K_{\mathrm{n}}\overline{c^2} \end{bmatrix} \tag{4.27}$$

$$\boldsymbol{f}_0(t) = \sum_{k=1}^{n_{\mathrm{teeth}}} g_k(t) \left(h \begin{bmatrix} -K_{\mathrm{t}}\overline{sc} - K_{\mathrm{n}}\overline{s^2} \\ K_{\mathrm{t}}\overline{s^2} - K_{\mathrm{n}}\overline{sc} \end{bmatrix} + \begin{bmatrix} -K_{\mathrm{te}}\overline{c} - K_{\mathrm{ne}}\overline{s} \\ K_{\mathrm{te}}\overline{s} - K_{\mathrm{ne}}\overline{c} \end{bmatrix} \right) \tag{4.28}$$

where, for t such that $g_k(t) = 1$,

$$\overline{s} = \int_{z_0(t)}^{z_1(t)} \sin\theta_k(z,t)\,\mathrm{d}z = \begin{cases} b\sin\theta_k(0,t) & \text{if } \kappa = 0 \\ -\frac{1}{\kappa}\left(-\cos\theta\right)\Big|_{\theta=\theta_{k,\mathrm{lo}}(t)}^{\theta=\theta_{k,\mathrm{hi}}(t)} & \text{if } \kappa \neq 0 \end{cases} \tag{4.29}$$

$$\overline{c} = \int_{z_0(t)}^{z_1(t)} \cos\theta_k(z,t)\,\mathrm{d}z = \begin{cases} b\cos\theta_k(0,t) & \text{if } \kappa = 0 \\ -\frac{1}{\kappa}\sin\theta\Big|_{\theta=\theta_{k,\mathrm{lo}}(t)}^{\theta=\theta_{k,\mathrm{hi}}(t)} & \text{if } \kappa \neq 0 \end{cases} \tag{4.30}$$

$$\overline{s^2} = \int_{z_0(t)}^{z_1(t)} \sin^2\theta_k(z,t)\,\mathrm{d}z = \begin{cases} b\sin^2\theta_k(0,t) & \text{if } \kappa = 0 \\ -\frac{1}{\kappa}\left(\frac{1}{2}\left(\theta - \sin\theta\cos\theta\right)\right)\Big|_{\theta=\theta_{k,\mathrm{lo}}(t)}^{\theta=\theta_{k,\mathrm{hi}}(t)} & \text{if } \kappa \neq 0 \end{cases} \tag{4.31}$$

$$\overline{c^2} = \int_{z_0(t)}^{z_1(t)} \cos^2\theta_k(z,t)\,\mathrm{d}z = \begin{cases} b\cos^2\theta_k(0,t) & \text{if } \kappa = 0 \\ -\frac{1}{\kappa}\left(\frac{1}{2}\left(\theta + \sin\theta\cos\theta\right)\right)\Big|_{\theta=\theta_{k,\mathrm{lo}}(t)}^{\theta=\theta_{k,\mathrm{hi}}(t)} & \text{if } \kappa \neq 0 \end{cases} \tag{4.32}$$

$$\overline{sc} = \int_{z_0(t)}^{z_1(t)} \sin\theta_k(z,t)\cos\theta_k(z,t)\,\mathrm{d}z = \begin{cases} b\sin\theta_k(0,t)\cos\theta_k(0,t) & \text{if } \kappa = 0 \\ -\frac{1}{\kappa}\left(-\frac{1}{2}\cos^2\theta\right)\Big|_{\theta=\theta_{k,\mathrm{lo}}(t)}^{\theta=\theta_{k,\mathrm{hi}}(t)} & \text{if } \kappa \neq 0 \end{cases} \tag{4.33}$$

4.2.2 Analytical Solution for Vibration Between Cuts

When no teeth are in contact with the workpiece, eq. (4.4) simplifies to an ordinary differential equation (ODE), which can be solved analytically:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}_\mathrm{o}(\boldsymbol{p})\boldsymbol{x}(t) + \boldsymbol{c}_\mathrm{o}(t,\boldsymbol{p}) \qquad \forall t \text{ s.t. } \left(g_k(t) = 0\ \forall k \in \left\{1,\dots,n_\mathrm{teeth}\right\}\right) \tag{4.34}$$

where

$$\boldsymbol{A}_\mathrm{o}(\boldsymbol{p}) = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{M}^{-1}\boldsymbol{K} & -\boldsymbol{M}^{-1}\boldsymbol{C} \end{bmatrix} \qquad \boldsymbol{c}_\mathrm{o}(t,\boldsymbol{p}) = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{M}^{-1}\boldsymbol{f}_\mathrm{u}(t) \end{bmatrix} \tag{4.35}$$

It can be shown that the solution to eq. (4.34), using eq. (4.9), is

$$\boldsymbol{x}(t) = \boldsymbol{x}_\mathrm{ss}(t) + \boldsymbol{x}_\mathrm{h}(t) \qquad \forall t \text{ s.t. } \left(g_k(t) = 0\ \forall k \in \left\{1,\dots,n_\mathrm{teeth}\right\}\right) \tag{4.36}$$

where

$$\boldsymbol{x}_\mathrm{h}(t) = \mathrm{e}^{\boldsymbol{A}_\mathrm{o}(\boldsymbol{p})(t-t_0)}\left(\boldsymbol{x}(t_0) - \boldsymbol{x}_\mathrm{ss}(t_0)\right) \tag{4.37}$$

$$\boldsymbol{x}_\mathrm{ss}(t) = \boldsymbol{x}_{\mathrm{ss},0} + \mathcal{R}\left[\boldsymbol{x}_{\mathrm{ss,p}}\mathrm{e}^{\mathrm{i}\Omega t}\right] \tag{4.38}$$

$$\boldsymbol{x}_{\mathrm{ss},0} = -\left(\boldsymbol{A}_\mathrm{o}(\boldsymbol{p})\right)^{-1}\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{M}^{-1}\boldsymbol{f}_{\mathrm{u},0} \end{bmatrix} \tag{4.39}$$

$$\boldsymbol{x}_{\mathrm{ss,p}} = \left(\mathrm{i}\Omega\boldsymbol{I} - \boldsymbol{A}_\mathrm{o}(\boldsymbol{p})\right)^{-1}\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{M}^{-1}\boldsymbol{f}_{\mathrm{u,p}} \end{bmatrix} \tag{4.40}$$

Note that this implies that

$$\boldsymbol{M}^{-1}\boldsymbol{f}_{\mathrm{u},0} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}\left(-\left(\boldsymbol{A}_\mathrm{o}(\boldsymbol{p})\right)\boldsymbol{x}_{\mathrm{ss},0}\right) \tag{4.41}$$

$$\boldsymbol{M}^{-1}\boldsymbol{f}_{\mathrm{u,p}} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}\left(\mathrm{i}\Omega\boldsymbol{I} - \boldsymbol{A}_\mathrm{o}(\boldsymbol{p})\right)\boldsymbol{x}_{\mathrm{ss,p}} \tag{4.42}$$

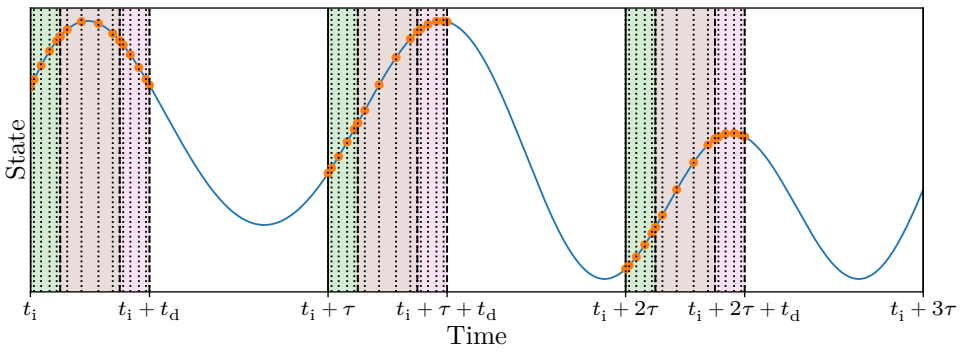


FIGURE 4.4: Illustration of the spectral element segments, elements, and nodes for approximating the solution of eq. (4.4). The state of the system is indicated with a curved blue line. The divisions between segments are indicated with solid vertical lines. The end of each element is indicated with a vertical dashed line. The location of each node is indicated with a vertical line (solid, dashed, or dotted), and the corresponding value of the state is indicated with an orange circular marker. The intervals of time during cuts have a shaded backgrounds (with a different color for each element within a segment), while the intervals of time between cuts have a white background.

which provides a convenient way to obtain $\boldsymbol{M}^{-1}\boldsymbol{f}_\mathrm{u}(t)$ from the steady-state solution between cuts $\boldsymbol{x}_\mathrm{ss}(t)$.

4.2.3 Approximate Solution using the Spectral Element Method

The full solution of the DDE model (eq. (4.4)) can be approximated in a computationally efficient way by an extended version of the spectral element method introduced by Khasawneh and Mann [10]. Let $\boldsymbol{y}(t)$ be this approximation of the state $\boldsymbol{x}(t)$. Time is split into segments of length τ, where each segment begins at the start of a cut and ends at the start of the next cut. The time interval during each cut is split into multiple elements. Within each element, the solution is approximated by polynomial interpolation between the values of $\boldsymbol{y}(t)$ at discrete times, known as nodes. Between cuts, the solution is described by eq. (4.36), where the initial condition is the node at the end of the preceding cut. This is illustrated in fig. 4.4, where the start of the initial cut is t_i, and the duration of each cut is t_d.

The spectral element method computes an approximate mapping from the nodes

in one segment to the nodes in the next, which can be written as:

$$\begin{bmatrix} \boldsymbol{y}\left(t_{\mathrm{i}}+(k+1)\tau+\eta_0\right) \\ \boldsymbol{y}\left(t_{\mathrm{i}}+(k+1)\tau+\eta_1\right) \\ \vdots \\ \boldsymbol{y}\left(t_{\mathrm{i}}+(k+1)\tau+\eta_\ell\right) \end{bmatrix} = \boldsymbol{Q}_{k+1}(\boldsymbol{p}) \begin{bmatrix} \boldsymbol{y}(t_{\mathrm{i}}+k\tau+\eta_0) \\ \boldsymbol{y}(t_{\mathrm{i}}+k\tau+\eta_1) \\ \vdots \\ \boldsymbol{y}(t_{\mathrm{i}}+k\tau+\eta_\ell) \end{bmatrix} + \boldsymbol{r}_{k+1}(\boldsymbol{p}), \tag{4.43}$$

$$0 \le \eta_0 \le \cdots \le \eta_\ell \le t_{\mathrm{d}}, \quad k = 0, 1, ...$$

where $\boldsymbol{Q}_{k+1}(\boldsymbol{p})$ and $\boldsymbol{r}_{k+1}(\boldsymbol{p})$ represent the mapping from segment k to segment $k+1$, and $\eta_0, \ldots, \eta_\ell$ are the times of the each segment's nodes relative to the start of the segment. Due to the periodicity of $\boldsymbol{A}(t, \boldsymbol{p})$, $\boldsymbol{B}(t, \boldsymbol{p})$, and $\boldsymbol{c}(t, \boldsymbol{p})$ in eq. (4.4) with respect to the spindle rotation period, the mapping is also periodic:

$$\boldsymbol{Q}_{k+n_{\mathrm{teeth}}}(\boldsymbol{p}) = \boldsymbol{Q}_k(\boldsymbol{p}) \qquad \boldsymbol{r}_{k+n_{\mathrm{teeth}}}(\boldsymbol{p}) = \boldsymbol{r}_k(\boldsymbol{p}) \tag{4.44}$$

So, given the solution at the initial nodes, $\boldsymbol{y}(t_{\mathrm{i}}+\eta_0), \ldots, \boldsymbol{y}(t_{\mathrm{i}}+\eta_\ell)$, a continuous approximation of the solution for the first n segments can be obtained by the following steps:

1. Compute $\boldsymbol{Q}_k(\boldsymbol{p})$ and $\boldsymbol{r}_k(\boldsymbol{p})$ for $k = 1, \ldots, n_{\mathrm{teeth}}$.

2. Apply eq. (4.43) repeatedly to compute the solution at the nodes for each of the first n segments.

3. For each segment:

   - Within the cut, use Lagrange polynomial interpolation between the nodes within each element.
   - Following the cut, use the last node from the cut as the initial condition for eq. (4.36).

The matrices $\boldsymbol{Q}_1(\boldsymbol{p}), \ldots, \boldsymbol{Q}_{n_{\mathrm{teeth}}}(\boldsymbol{p})$ can also be used to compute the CMs, and thus stability, of the system.

For more details, see Appendix E. Equation (4.4) corresponds to eq. (E.1), and eq. (4.36) corresponds to eq. (E.2). It was found that, for this problem, a good choice for the number of elements and the polynomial order was $n_{\mathrm{e}} = 3$ and $n_{\mathrm{o}} = 5$. The boundaries of the elements were chosen to coincide with the times when the cutting forces were non-differentiable, i.e. when the tip of the tooth started and stopped cutting and when the point on the tooth at height b started and stopped cutting. This choice of element boundaries was beneficial because the spectral element approximation was infinitely differentiable within each element but only continuous across elements.

4.3 Experimental Setup

The proposed model and methods were evaluated using data collected from test cuts on a 5-axis linear motor milling machine, cutting into an aluminum (7050-T7451) block, with various spindle speeds and cutting depths [60]. The tool was a carbide end mill with $n_{\mathrm{teeth}} = 2$, diameter $D_{\mathrm{tool}} = 12.75\,\mathrm{mm}$, helix angle $\beta = 30°$, and 106 mm overhang. The radial immersion ratio was $\rho = 0.05$, and the feed rate was $h = 0.127\,\mathrm{mm}$ per tooth. A clean-up pass to smooth the workpiece was performed between each test cut.

The milling machine was instrumented with sensors to collect time series data for each test cut. The time series data for each test cut consisted of a tachometer signal and noisy measurements of the position of the tool tip. Figure 4.5 shows a photograph of the sensor setup. The position of the tool 19 mm from the tip was measured using two capacitive sensors held close to the tool in a rigid fixture; the position of the tool tip was estimated by scaling these measurements. The tachometer signal indicated at each instant in time whether a mark drawn on the tool was detected or not; this indirectly provided information about the spindle speed and orientation of the tool. The sampling frequency for the sensors was 25 kHz. For more details on the

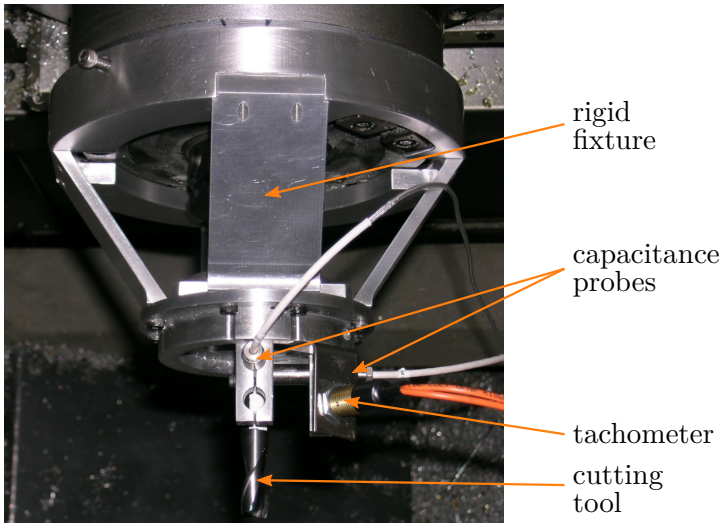


FIGURE 4.5: Annotated photograph of the cutting tool and sensor setup. [Photograph by Brian P. Mann, 2003.]

experimental setup and data collection, see [60].

4.4 Fitting to Time Series Data

The objective of this chapter is to estimate the unknown model parameters by fitting the model to the time series data, so that stability charts can be generated using the estimated parameters. In theory, it would be possible to estimate all of the unknowns simultaneously by trajectory matching, using a global optimization algorithm to fit the spectral element approximation of the milling model to the time series. However, this is difficult due to the large number of unknown parameters. Fortunately, the unknown parameters can instead be estimated in a sequence of stages, where each stage estimates a different subset of the parameters. For example, it is possible to estimate parameters describing $\boldsymbol{M}^{-1}\boldsymbol{K}$ and $\boldsymbol{M}^{-1}\boldsymbol{C}$ using the portions of the data between cuts, independently of the parameters describing the cutting forces. This approach of breaking up the overall problem into smaller subproblems limits the number of parameters which need to be estimated simultaneously, which significantly simplifies the optimization problems which need to be solved. This section describes

how the model can be parameterized, provides a brief overview of the trajectory matching technique used in the last two stages, and then describes the stages of the parameter estimation strategy in sequence.

4.4.1 Parameterizing the Model

To make model fitting work well, it is beneficial to define the parameter vector $\boldsymbol{p}$ such that the number of unknown parameters to fit is minimized and the parameter values always meet physical constraints. This section describes the specific parameter choices, in the order which they were estimated. The details of how they were estimated are provided in the following sections.

First, the axial cut depth b, radial immersion ratio ρ, and feed h per tooth were assumed to be known but possibly different between time series; they correspond to settings on the milling machine:

$$p_1 = b \tag{4.45}$$

$$p_2 = \rho \tag{4.46}$$

$$p_3 = h \tag{4.47}$$

The number of teeth, diameter, and helix angle of the tool were assumed to be known and the same for all time series:

$$p_4 = n_{\text{teeth}} \tag{4.48}$$

$$p_5 = D_{\text{tool}} \tag{4.49}$$

$$p_6 = \beta \tag{4.50}$$

The spindle speed Ω should be approximately controlled by the milling machine, but, for higher accuracy, Ω was estimated from separate tachometer measurements. The time $t_{\text{up}}(0)$ of the first upward edge of the tachometer signal—which was used as an indicator of the phase of the tool—could also be estimated from the tachometer

signal, as described in Section 4.4.3.

$$p_7 = \Omega \tag{4.51}$$

$$p_8 = t_{\mathrm{up}}(0) \tag{4.52}$$

As described in Section 4.4.4, an initial estimate of the steady-state solution outside of cutting, $\boldsymbol{x}_{\mathrm{ss}}(t)$ (eq. (4.38)), was obtained by fitting the steady-state vibration data before cutting started. Let

$$\boldsymbol{q}_{\mathrm{ss}}(t) = \boldsymbol{q}_{\mathrm{ss,0}} + \mathcal{R}\left[\boldsymbol{q}_{\mathrm{ss,p}} \mathrm{e}^{\mathrm{i}\Omega t}\right] \tag{4.53}$$

such that

$$\boldsymbol{x}_{\mathrm{ss}}(t) = \begin{bmatrix} \boldsymbol{q}_{\mathrm{ss}}(t) \\ \dot{\boldsymbol{q}}_{\mathrm{ss}}(t) \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}_{\mathrm{ss,0}} \\ \boldsymbol{0} \end{bmatrix} + \mathcal{R}\left[\begin{bmatrix} \boldsymbol{q}_{\mathrm{ss,p}} \\ \mathrm{i}\Omega \boldsymbol{q}_{\mathrm{ss,p}} \end{bmatrix} \mathrm{e}^{\mathrm{i}\Omega t}\right] \tag{4.54}$$

So, the coefficients of the steady-state solution in eq. (4.38) are

$$\boldsymbol{x}_{\mathrm{ss,0}} = \begin{bmatrix} \boldsymbol{q}_{\mathrm{ss,0}} \\ \boldsymbol{0} \end{bmatrix} \qquad \boldsymbol{x}_{\mathrm{ss,p}} = \begin{bmatrix} \boldsymbol{q}_{\mathrm{ss,p}} \\ \mathrm{i}\Omega \boldsymbol{q}_{\mathrm{ss,p}} \end{bmatrix} \tag{4.55}$$

Six real parameters are sufficient to describe the steady-state solution:

$$p_9 = \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{q}_{\mathrm{ss,0}} \tag{4.56}$$

$$p_{10} = \begin{bmatrix} 0 & 1 \end{bmatrix} \boldsymbol{q}_{\mathrm{ss,0}} \tag{4.57}$$

$$p_{11} = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathcal{R}\left[\boldsymbol{q}_{\mathrm{ss,p}}\right] \qquad p_{12} = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathcal{I}\left[\boldsymbol{q}_{\mathrm{ss,p}}\right] \tag{4.58}$$

$$p_{13} = \begin{bmatrix} 0 & 1 \end{bmatrix} \mathcal{R}\left[\boldsymbol{q}_{\mathrm{ss,p}}\right] \qquad p_{14} = \begin{bmatrix} 0 & 1 \end{bmatrix} \mathcal{I}\left[\boldsymbol{q}_{\mathrm{ss,p}}\right] \tag{4.59}$$

Then,

$$\boldsymbol{q}_{\mathrm{ss,0}} = \begin{bmatrix} p_9 \\ p_{10} \end{bmatrix} \qquad \boldsymbol{q}_{\mathrm{ss,p}} = \begin{bmatrix} p_{11} + \mathrm{i}p_{12} \\ p_{13} + \mathrm{i}p_{14} \end{bmatrix} \tag{4.60}$$

Recall that, by eqs. (4.9), (4.41), and (4.42), this also describes $\boldsymbol{M}^{-1}\boldsymbol{f}_{\mathrm{u}}(t)$. Note that these parameters differ between time series. In particular, the amplitude tends to increase with spindle speed, and the phase relative to $t = 0\,\mathrm{s}$ differs.

As described in Section 4.4.5, by fitting eq. (4.34) to the vibration between cuts, it is possible to estimate the $\boldsymbol{M}^{-1}\boldsymbol{K}$ and $\boldsymbol{M}^{-1}\boldsymbol{C}$ matrices, refine the estimate of the steady-state solution $\boldsymbol{x}_{\mathrm{ss}}(t)$ (which also describes $\boldsymbol{M}^{-1}\boldsymbol{f}_{\mathrm{u}}(t)$), and estimate the phase of the cuts. These were parameterized as follows. First, it is reasonable to assume that the $\boldsymbol{M}$, $\boldsymbol{K}$, and $\boldsymbol{C}$ matrices are approximately diagonal, so $\boldsymbol{M}^{-1}\boldsymbol{K}$ and $\boldsymbol{M}^{-1}\boldsymbol{C}$ can be written as

$$\boldsymbol{M}^{-1}\boldsymbol{K} = \begin{bmatrix} \left(\boldsymbol{M}^{-1}\boldsymbol{K}\right)_{\mathrm{XX}} & 0 \\ 0 & \left(\boldsymbol{M}^{-1}\boldsymbol{K}\right)_{\mathrm{YY}} \end{bmatrix} \tag{4.61}$$

$$\boldsymbol{M}^{-1}\boldsymbol{C} = \begin{bmatrix} \left(\boldsymbol{M}^{-1}\boldsymbol{C}\right)_{\mathrm{XX}} & 0 \\ 0 & \left(\boldsymbol{M}^{-1}\boldsymbol{C}\right)_{\mathrm{YY}} \end{bmatrix} \tag{4.62}$$

However, to be physically realistic, all the diagonal elements in these matrices should be positive, so it's beneficial to use the logarithm of a diagonal element as the corresponding parameter to ensure that the resulting diagonal element is positive. Additionally, the two diagonal elements in each matrix should be very similar to each other, so it can be beneficial for one parameter to describe the first element and the other parameter to describe the ratio between the two elements. In other words, the parameters used to describe $\boldsymbol{M}^{-1}\boldsymbol{K}$ and $\boldsymbol{M}^{-1}\boldsymbol{C}$ were the following:

$$p_{15} = \ln\left(\left(\boldsymbol{M}^{-1}\boldsymbol{K}\right)_{\mathrm{XX}}\right) \tag{4.63}$$

$$p_{16} = \ln\frac{\left(\boldsymbol{M}^{-1}\boldsymbol{K}\right)_{\mathrm{YY}}}{\left(\boldsymbol{M}^{-1}\boldsymbol{K}\right)_{\mathrm{XX}}} = \ln\left(\left(\boldsymbol{M}^{-1}\boldsymbol{K}\right)_{\mathrm{YY}}\right) - \ln\left(\left(\boldsymbol{M}^{-1}\boldsymbol{K}\right)_{\mathrm{XX}}\right) \tag{4.64}$$

$$p_{17} = \ln\left(\left(\boldsymbol{M}^{-1}\boldsymbol{C}\right)_{\mathrm{XX}}\right) \tag{4.65}$$

$$p_{18} = \ln\frac{\left(\boldsymbol{M}^{-1}\boldsymbol{C}\right)_{\mathrm{YY}}}{\left(\boldsymbol{M}^{-1}\boldsymbol{C}\right)_{\mathrm{XX}}} = \ln\left(\left(\boldsymbol{M}^{-1}\boldsymbol{C}\right)_{\mathrm{YY}}\right) - \ln\left(\left(\boldsymbol{M}^{-1}\boldsymbol{C}\right)_{\mathrm{XX}}\right) \tag{4.66}$$

Then,

$$\boldsymbol{M}^{-1}\boldsymbol{K} = \begin{bmatrix} \mathrm{e}^{p_{15}} & 0 \\ 0 & \mathrm{e}^{p_{15}+p_{16}} \end{bmatrix} \qquad \boldsymbol{M}^{-1}\boldsymbol{C} = \begin{bmatrix} \mathrm{e}^{p_{17}} & 0 \\ 0 & \mathrm{e}^{p_{17}+p_{18}} \end{bmatrix} \tag{4.67}$$

The phase was represented by the orientation of the tip of a tooth at time $t_{\mathrm{up}}(0)$:

$$p_{19} = \theta_1 \left(0, t_{\mathrm{up}}(0)\right) \tag{4.68}$$

since the orientation of the tool at time $t_{\mathrm{up}}(0)$ should be consistent regardless of the other parameters, as long as the tachometer position and the mark on the tool detected by the tachometer are consistent.

During cutting, the values of $\boldsymbol{M}^{-1}\boldsymbol{K}_{\mathrm{c}}(t)$ and $\boldsymbol{M}^{-1}\boldsymbol{f}_0(t)$ are also necessary. Dividing through by the first element in the mass matrix reduces the number of parameters by one, as shown below. First, observe that $\boldsymbol{M}^{-1}$ can be written as

$$\boldsymbol{M}^{-1} = \begin{bmatrix} \frac{1}{M_{\mathrm{XX}}} & 0 \\ 0 & \frac{1}{M_{\mathrm{YY}}} \end{bmatrix} = \frac{1}{M_{\mathrm{XX}}} \begin{bmatrix} 1 & 0 \\ 0 & \frac{M_{\mathrm{XX}}}{M_{\mathrm{YY}}} \end{bmatrix} \tag{4.69}$$

So, substituting eqs. (4.27) and (4.28),

$$\boldsymbol{M}^{-1}\boldsymbol{K}_{\mathrm{c}}(t) \tag{4.70}$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & \frac{M_{\mathrm{XX}}}{M_{\mathrm{YY}}} \end{bmatrix} \sum_{k=1}^{n_{\mathrm{teeth}}} g_k(t) \begin{bmatrix} -\frac{K_{\mathrm{t}}}{M_{\mathrm{XX}}}\overline{sc} - \frac{K_{\mathrm{n}}}{M_{\mathrm{XX}}}\overline{s^2} & -\frac{K_{\mathrm{t}}}{M_{\mathrm{XX}}}\overline{c^2} - \frac{K_{\mathrm{n}}}{M_{\mathrm{XX}}}\overline{sc} \\ \frac{K_{\mathrm{t}}}{M_{\mathrm{XX}}}\overline{s^2} - \frac{K_{\mathrm{n}}}{M_{\mathrm{XX}}}\overline{sc} & \frac{K_{\mathrm{t}}}{M_{\mathrm{XX}}}\overline{sc} - \frac{K_{\mathrm{n}}}{M_{\mathrm{XX}}}\overline{c^2} \end{bmatrix}$$

$$\boldsymbol{M}^{-1}\boldsymbol{f}_0(t) \tag{4.71}$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & \frac{M_{\mathrm{XX}}}{M_{\mathrm{YY}}} \end{bmatrix} \sum_{k=1}^{n_{\mathrm{teeth}}} g_k(t) \left(h \begin{bmatrix} -\frac{K_{\mathrm{t}}}{M_{\mathrm{XX}}}\overline{sc} - \frac{K_{\mathrm{n}}}{M_{\mathrm{XX}}}\overline{s^2} \\ \frac{K_{\mathrm{t}}}{M_{\mathrm{XX}}}\overline{s^2} - \frac{K_{\mathrm{n}}}{M_{\mathrm{XX}}}\overline{sc} \end{bmatrix} + \begin{bmatrix} -\frac{K_{\mathrm{te}}}{M_{\mathrm{XX}}}\overline{c} - \frac{K_{\mathrm{ne}}}{M_{\mathrm{XX}}}\overline{s} \\ \frac{K_{\mathrm{te}}}{M_{\mathrm{XX}}}\overline{s} - \frac{K_{\mathrm{ne}}}{M_{\mathrm{XX}}}\overline{c} \end{bmatrix} \right)$$

In other words, dividing the pressure and edge coefficients by $\boldsymbol{M}_{\mathrm{XX}}$ allows for $\boldsymbol{M}_{\mathrm{XX}}$ to be eliminated as a parameter. As with the stiffness and damping, many of the parameters are known to be positive or to have a fairly consistent ratio, so it is beneficial to use logarithms and ratios as the parameters:

$$p_{20} = \ln \frac{K_{\mathrm{t}}}{\boldsymbol{M}_{\mathrm{XX}}} \tag{4.72}$$

$$p_{21} = \ln \frac{K_{\mathrm{n}}}{K_{\mathrm{t}}} = \ln \frac{K_{\mathrm{n}}/\boldsymbol{M}_{\mathrm{XX}}}{K_{\mathrm{t}}/\boldsymbol{M}_{\mathrm{XX}}} = \ln \frac{K_{\mathrm{n}}}{\boldsymbol{M}_{\mathrm{XX}}} - \ln \frac{K_{\mathrm{t}}}{\boldsymbol{M}_{\mathrm{XX}}} \tag{4.73}$$

$$p_{22} = \ln \frac{K_{\text{te}}}{\boldsymbol{M}_{\text{XX}}} \tag{4.74}$$

$$p_{23} = \ln \frac{K_{\text{ne}}}{K_{\text{te}}} = \ln \frac{K_{\text{ne}}/\boldsymbol{M}_{\text{XX}}}{K_{\text{te}}/\boldsymbol{M}_{\text{XX}}} = \ln \frac{K_{\text{ne}}}{\boldsymbol{M}_{\text{XX}}} - \ln \frac{K_{\text{te}}}{\boldsymbol{M}_{\text{XX}}} \tag{4.75}$$

$$p_{24} = \ln \frac{\boldsymbol{M}_{\text{YY}}}{\boldsymbol{M}_{\text{XX}}} = -\ln \frac{\boldsymbol{M}_{\text{XX}}}{\boldsymbol{M}_{\text{YY}}} \tag{4.76}$$

Then,

$$\boldsymbol{M}^{-1}\boldsymbol{K}_{\text{c}}(t) \tag{4.77}$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & \mathrm{e}^{-p_{24}} \end{bmatrix} \sum_{k=1}^{n_{\text{teeth}}} g_k(t) \begin{bmatrix} -\mathrm{e}^{p_{20}\overline{sc}} - \mathrm{e}^{p_{20}+p_{21}\overline{s^2}} & -\mathrm{e}^{p_{20}\overline{c^2}} - \mathrm{e}^{p_{20}+p_{21}\overline{sc}} \\ \mathrm{e}^{p_{20}\overline{s^2}} - \mathrm{e}^{p_{20}+p_{21}\overline{sc}} & \mathrm{e}^{p_{20}\overline{sc}} - \mathrm{e}^{p_{20}+p_{21}\overline{c^2}} \end{bmatrix}$$

$$\boldsymbol{M}^{-1}\boldsymbol{f}_0(t) \tag{4.78}$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & \mathrm{e}^{-p_{24}} \end{bmatrix} \sum_{k=1}^{n_{\text{teeth}}} g_k(t) \left(p_3 \begin{bmatrix} -\mathrm{e}^{p_{20}\overline{sc}} - \mathrm{e}^{p_{20}+p_{21}\overline{s^2}} \\ \mathrm{e}^{p_{20}\overline{s^2}} - \mathrm{e}^{p_{20}+p_{21}\overline{sc}} \end{bmatrix} + \begin{bmatrix} -\mathrm{e}^{p_{22}\overline{c}} - \mathrm{e}^{p_{22}+p_{23}\overline{s}} \\ \mathrm{e}^{p_{22}\overline{s}} - \mathrm{e}^{p_{22}+p_{23}\overline{c}} \end{bmatrix} \right)$$

In summary, $\boldsymbol{p} = \begin{bmatrix} p_1 & \cdots & p_{24} \end{bmatrix}^\top$ are all the parameters needed for the milling dynamics model, eq. (4.4). Note that when fitting to time series, the initial conditions are also unknowns which need to be estimated.

4.4.2 Overview of Trajectory Matching

One way to estimate the parameters of a dynamics model is to solve an optimization problem for the parameters, where the objective function describes how closely trajectories predicted by the model match the measured time series. The objective function incorporates any prior belief about the parameters and penalizes differences between the noisy measured time series and the noiseless measurements corresponding to the predicted trajectories. This works even if the measurements represent only a subset of the state variables, such as positions but not velocities. For cases where the measurement noise vectors are independent, additive, and multivariate normal and the prior belief on the parameters is a multivariate normal distribution

independent of the measurement noise, the optimization problem can be expressed as a nonlinear least squares problem which accounts for the distributions of the measurement noise and the prior belief. This approach is used to estimate parameters in Sections 4.4.5 and 4.4.6.

Fully describing a trajectory with the dynamics model requires the initial conditions of the trajectory in addition to the model parameters. Typically, these initial conditions are unknown and, as a result, need to be estimated simultaneously with the model parameters in order to match the predicted trajectory to the measurements. For an ODE model, these initial conditions consist of the system state at a single instant in time. For a DDE model with delay τ, however, the initial conditions consist of the trajectory of $\boldsymbol{x}(t)$ over a continuous interval of length τ, due to the dependency of the dynamics on past values of $\boldsymbol{x}(t)$. The spectral element method provides a convenient approximate representation of these initial conditions: the approximation of $\boldsymbol{x}(t)$ at the nodes in the first segment. By representing the initial conditions in this way, the future trajectory can easily be approximated using the spectral element method, as described in Section 4.2.3.

In summary, the parameters of a dynamics model can be estimated from time series by matching trajectories predicted by the model to the time series measurements. Under certain assumptions, this can be framed as a nonlinear least squares optimization problem which accounts for the covariance of the measurement noise and any prior belief on the parameters. The unknowns to optimize consist of both the unknown model parameters and the unknown initial conditions. The unknown model parameters may consist of parameters which are common to multiple time series, as well as parameters which are separate for each time series. Regardless, all of the unknowns can be concatenated together into a single vector of unknowns for the optimization problem. The optimization problem can be solved using standard techniques. For more details on the objective function, see Appendix C.

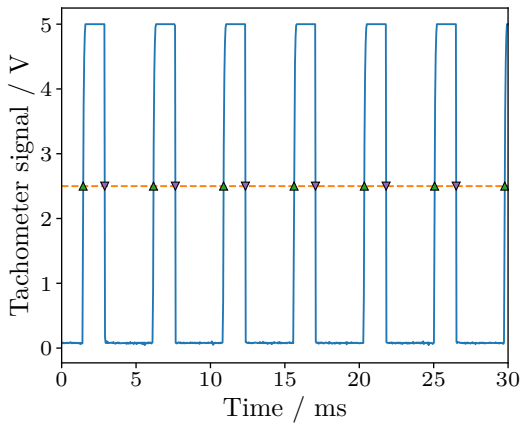


FIGURE 4.6: Example of finding the upward and downward edges of the tachometer signal; only the first 30 ms is shown, due to space limitations. The tachometer signal is the solid blue line; the threshold of 2.5 V shown with the dashed orange line; the times corresponding to the detected upward and downward edges are indicated with upward-pointing and downward-pointing triangular markers, respectively.

4.4.3 Estimating the Spindle Speed

As described in Section 4.3, the milling machine was instrumented with a tachometer to more accurately estimate the spindle speed. This section describes how the spindle speed was estimated from the tachometer signal. The tachometer signal was approximately a square wave between 0 V and 5 V; the edges of the signal were used to estimate the spindle speed. First, the upward and downward edges were estimated by finding the intersections between the tachometer signal and a threshold of 2.5 V, using linear interpolation between adjacent points of the tachometer signal. This is illustrated in fig. 4.6. After estimating the edges, a linear model

$$t_{\text{up}}(i) = t_{\text{up}}(0) + Ti \tag{4.79}$$

was fit to the indices and values of the estimated upward edges using ordinary least squares (OLS), where the first upward edge is $t_{\text{up}}(0)$, the next upward edge is $t_{\text{up}}(1)$, etc. The estimated spindle speed was given by $\Omega = 2\pi/T$. Finally, the estimated values of $t_{\text{up}}(0)$ and T were checked in various ways to verify that they were a good fit: (1) the estimated Ω was checked against the nominal spindle speed reported by

the milling machine, (2) the residuals between the actual estimated upward edges and eq. (4.79) were verified to be no more than a threshold fraction of T, and (3) the relative errors from T for the intervals between consecutive upward edges and consecutive downward edges were verified to be less than a threshold. If any of these checks failed, then the experiment was thrown out. The most common cause of failure was a small chip from the workpiece getting stuck to the side of the tool, which would occasionally interfere with the tachometer measurements.

Equation (4.79) was fit separately to the edges from before cutting started and the edges during cutting, since the spindle speed changed slightly during cutting due to the additional load on the machine. The estimated spindle speed before cutting was used to estimate the parameters of steady-state vibration before cutting started (Section 4.4.4), and the estimated spindle speed during cutting was used for fitting the dynamics model (Sections 4.4.5 and 4.4.6). Additionally, the estimate of $t_{\text{up}}(0)$ for the data during cutting was used to match the phases of the time series, since the orientation of the tool should have been the same for all upward edges.

4.4.4 Estimating Steady-State Vibration Before Cutting Started

For each experiment, the machine was started with the tool outside the workpiece and allowed to reach steady-state, then data collection was initialized, and then the tool was fed into the workpiece. The steady-state motion of the tool before cutting was estimated for two reasons: (1) the covariance of the difference between the measurements and the estimated motion was a good estimate of the covariance of the measurement noise, and (2) the estimated motion was a good initial guess for $\boldsymbol{q}_{\text{ss}}(t)$ (eq. (4.53)) for use in fitting the vibration between cuts (Section 4.4.5).

For each time series, the steady-state motion was estimated by fitting a Fourier

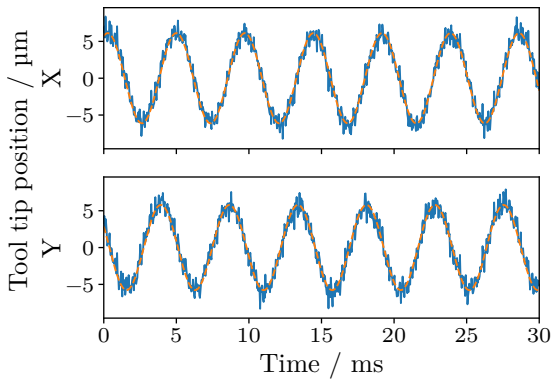


FIGURE 4.7: Example of fitting to the observations before cutting started; only the first 30 ms is shown, due to space limitations. The components in the X and Y directions are shown on separate axes. The position measurements are the solid blue lines, and the corresponding fitted steady-state vibration $\boldsymbol{q}_{\mathrm{ss}}(t)$ is indicated with dashed orange lines.

series

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{n} \left(a_k \cos k\Omega t + b_k \sin k\Omega t \right) \tag{4.80}$$

to each of the two position components of the measurements separately, using OLS. The value of Ω had been estimated earlier from the tachometer signal, as described in Section 4.4.3. For estimating $\boldsymbol{q}_{\mathrm{ss}}(t)$, the Fourier series were limited to $n = 1$; see fig. 4.7 for an example. For estimating the covariance of the measurement noise, $n = 10$ was used; the additional terms allowed fitting the data slightly more closely to obtain a slightly better estimate of the noise covariance in case the steady-state vibration did not perfectly match eq. (4.53).

4.4.5 Fitting Vibration Between Cuts and Estimating the Phase of Cuts

For the low radial immersion experiments, most of the time was spent with the tool between cuts, i.e. with no teeth in contact with the workpiece. So, the model could be fit to the data between cuts with fewer parameters; the parameters describing the cutting forces were not necessary. In particular, eq. (4.34) could be fit to the data instead of eq. (4.4). By fitting fewer model parameters, the optimization algorithm

converges faster and is more likely to find the global optimum. As an additional benefit, fitting the vibration between cuts can be used to estimate the phase when cuts occur. This section describes how the phase of the cuts and the relevant model parameters can be estimated.

Observe that if a time interval is between cuts, then all intervals which are integer multiples of τ later in time are also between cuts, since cuts are separated in time by τ. Given the information obtained in Sections 4.4.3 and 4.4.4, intervals between cuts can be described by two values: the interval length $\alpha\tau$, and the time $\varsigma\tau$ between the first upward tachometer edge $t_{\text{up}}(0)$ and the start of an interval between cuts. In other words, the intervals described by these two values are $[t_{\text{up}}(0)+\varsigma\tau+k\tau,\ t_{\text{up}}(0)+\varsigma\tau+k\tau+\alpha\tau]$ for $k = 0, 1, ...$

If suitable values of ς and α were known, then eq. (4.34) could be fit to the data in the intervals described by ς and α using trajectory matching. The parameters to estimate are the relevant unknown model parameters which are common to all time series, i.e. $p_{15}, \ldots, p_{18}$, the relevant unknown model parameters which differ between time series, i.e. $p_9, \ldots, p_{14}$, and the initial condition at the start of each interval. Note that good initial guesses for $p_9, \ldots, p_{14}$ for each time series are available from fitting the data before cutting starts, as described in Section 4.4.4. The trajectory matching can be framed as a least squares optimization problem, as described in Section 4.4.2, using the analytical solution (eq. (4.36)) for the predicted trajectories. The optimization problem can be solved using standard nonlinear least-squares methods, such as the `scipy.optimize.least_squares` SciPy library function [49] with the Trust Region Reflective algorithm. As in Section 3.5.2, only a subset of the parameters affects each term in the objective function, so sparse Jacobians can be used to significantly speed up the optimization algorithm. Specifically, each initial condition affects only the terms in the corresponding interval, and each time series has different values of $p_9, \ldots, p_{14}$. Figure 4.8 shows an example of fitting the model

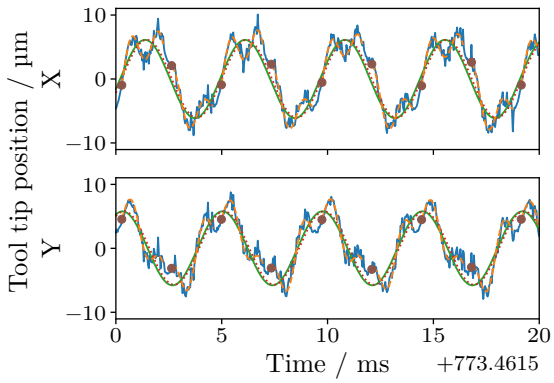


FIGURE 4.8: Example of fitting to the observations between cuts. Note that only part of one time series is shown; the model is actually fit to multiple time series simultaneously. The components in the X and Y directions are shown on separate axes. The position measurements are the solid blue lines, and the corresponding fitted positions $\boldsymbol{q}(t)$ between cuts are indicated with dashed orange lines. The estimated initial condition for each interval between cuts is indicated with a brown circular marker. The steady-state response $\boldsymbol{q}_{\text{ss}}(t)$ estimated from the data before cutting started (Section 4.4.4) and the updated estimate of $\boldsymbol{q}_{\text{ss}}(t)$ from the data between cuts (Section 4.4.5) are shown with solid green and dotted red lines, respectively.

to the data in the intervals between cuts. The fitted trajectories in the intervals are indicated with dashed orange lines, and the estimated initial condition for each interval is indicated with a brown circular marker.

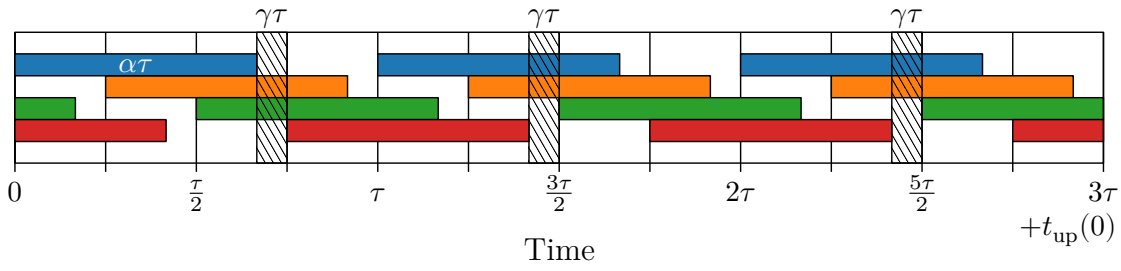
Unfortunately, the phase of the intervals between cuts was unknown, since none of the sensors could directly detect contact between the tool and the workpiece. So, it was necessary to find suitable values of ς and α using the time series data. One way to do this is to take advantage of the fact that eq. (4.34) should fit the data in intervals between cuts more closely than data in intervals which contain cuts, since the cutting forces cause the motion of the tool during cuts to deviate from the dynamics described by eq. (4.34). So, given a set of candidate values of (ς, α), the best candidate from the set can be chosen by comparing the quality of the fits for these candidates, as described by the least squares objective function. The question, then, is how to strategically choose candidates for (ς, α) in order to narrow in on the correct values.

First, it's useful to determine the maximum possible value of α for which there exists a value of ς such that all the intervals described by ς and α are between cuts. This maximum value for α is $1-\gamma$, where γ is the fraction of each tooth passage period for which a tooth may be cutting in any of the time series when they are aligned by their respective values of $t_{\text{up}}(0)$. When dealing with multiple time series of differing radial immersions and axial cut depths,

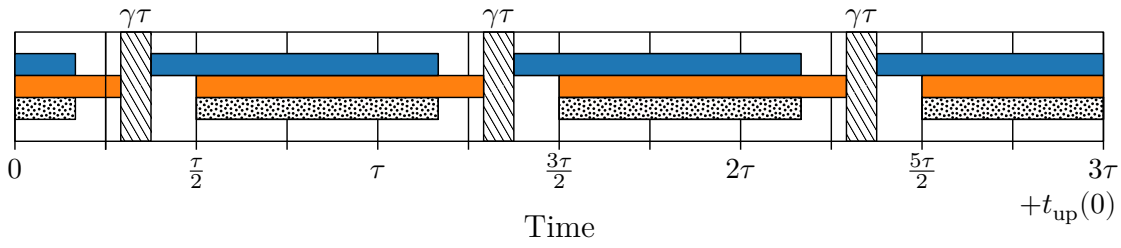
$$\gamma = \frac{\left(\max\limits_{k}\left(\theta_{\text{ex},k} + \theta_{\text{lag},k}\right)\right) - \left(\min\limits_{k}\theta_{\text{en},k}\right)}{2\pi/n_{\text{teeth}}} \tag{4.81}$$

where $\theta_{\text{en},k}$, $\theta_{\text{ex},k}$, and $\theta_{\text{lag},k}$ are the tooth tip cut entry, tooth tip cut exit, and lag angles for the time series with index k; these angles are described in more detail in Section 4.2.1. The process of estimating the phase can be implemented as a sequence of steps, where each step chooses between a small number of candidates of (ς, α). The first step starts with a smaller value of α in order to find a value of ς which corresponds to intervals between cuts, and then subsequent steps increase α and adjust the value of ς. After multiple steps, α approaches the maximum value of $1-\gamma$. At that point, the intervals described by ς and α are the largest possible intervals between cuts, and their endpoints describe when cuts occur.

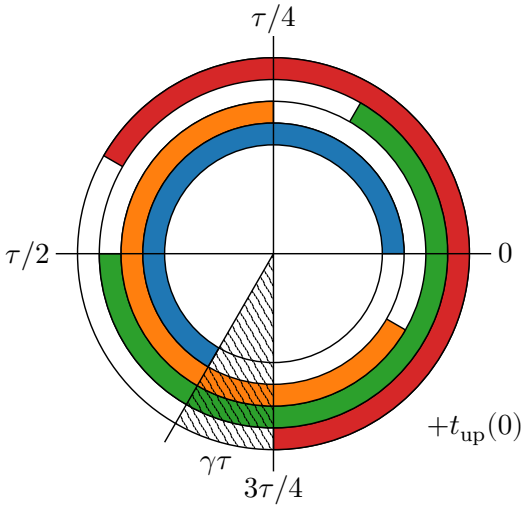
The first step considers four different candidates: $\varsigma \in \left\{\frac{0}{4}, \frac{1}{4}, \frac{2}{4}, \frac{3}{4}\right\}$, all with $\alpha = \frac{3}{4} - \gamma$. This value for α is chosen so that an interval of length $\gamma\tau$ cannot simultaneously overlap the intervals described by all four candidates. This ensures that there is at least one candidate which represents intervals which are entirely between cuts. Figure 4.9a shows how the choice of $\alpha = \frac{3}{4} - \gamma$ ensures this property. In this figure, the intervals for each of the four candidates are indicated with a solid color, and possible cutting intervals of duration $\gamma\tau$, separated by time τ, are indicated with hatched rectangles. If the hatched rectangles are slid together across the timeline, they never intersect all four candidates simultaneously. Figure 4.9c illustrates the



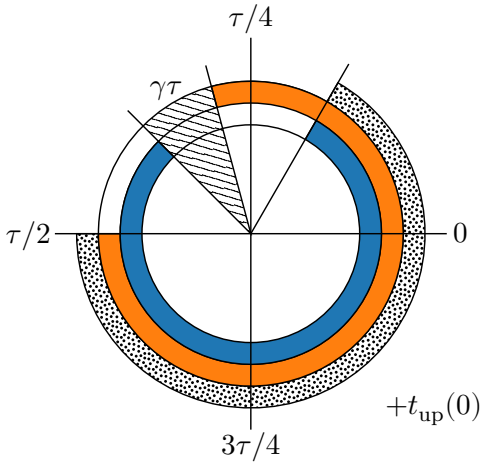
(a) Obtaining an initial estimate of an interval between cuts, using $\alpha = 3/4 - \gamma$.



(b) Expanding an estimate of an interval between cuts.



(c) Figure 4.9a wrapped around a circle, to illustrate the periodicity.



(d) Figure 4.9b wrapped around a circle, to illustrate the periodicity.

FIGURE 4.9: Strategy for estimating the phase of the cuts by the tool. The intervals to try fitting as vibration between cuts are indicated with solid backgrounds; the size of the interval when the tool may be cutting is indicated with a hatched region; and the previously chosen interval being expanded in figs. 4.9b and 4.9d is indicated with a dotted background.

same thing, but is wrapped around a circle to more easily show the periodicity. Once the model has been fit to the intervals for each candidate value of ς, the value of ς for which the model best fits the data is selected. Let ς_0 and α_0 be these initially-selected values.

In the following steps, the selected intervals are repeatedly expanded, until α is as close as desired to $1-\gamma$. For each step k, given values of ς_{k-1} and α_{k-1} which were selected on the previous step, ς_k and α_k can be chosen by considering two candidates which overlap the previously-selected candidate but are somewhat larger. This is illustrated in figs. 4.9b and 4.9d. The previously-selected candidate with intervals of length $\alpha_{k-1}\tau$ is indicated with a dotted background, and the two expanded candidates with intervals of length $\alpha_k\tau$ are indicated with solid backgrounds. If $\alpha_k = \alpha_{k-1} + \frac{1}{2}(1-\gamma-\alpha_{k-1})$, then if the hatched regions are slid together across the timeline, they cannot overlap both candidates simultaneously while remaining outside the intervals specified by $(\varsigma_{k-1}, \alpha_{k-1})$. So, if $(\varsigma_{k-1}, \alpha_{k-1})$ represents intervals which are entirely between cuts, then at least one of the candidates is guaranteed to represent intervals which are entirely between cuts. As the number of steps increases, α rapidly approaches the maximum of $1-\gamma$, which can be shown as follows. The value of α_i is given by

$$\alpha_i = \begin{cases} \frac{3}{4} - \gamma & \text{if } i = 0 \\ \alpha_{i-1} + \frac{1}{2}(1-\gamma-\alpha_{i-1}) & \text{if } i \in \{1, 2, ...\} \end{cases} \tag{4.82}$$

So, the difference between $1-\gamma$ and α_i is given by

$$(1-\gamma) - \alpha_i = \begin{cases} (1-\gamma) - (\frac{3}{4} - \gamma) & \text{if } i = 0 \\ (1-\gamma) - \Big(\alpha_{i-1} + \frac{1}{2}(1-\gamma-\alpha_{i-1})\Big) & \text{if } i \in \{1, 2, ...\} \end{cases} \tag{4.83}$$

$$= \begin{cases} \frac{1}{4} & \text{if } i = 0 \\ \frac{1}{2}(1-\gamma-\alpha_{i-1}) & \text{if } i \in \{1, 2, ...\} \end{cases} \tag{4.84}$$

$$= \tfrac{1}{4}\Big(\tfrac{1}{2}\Big)^i = 2^{-(i+2)} \tag{4.85}$$

In other words, the difference between $1-\gamma$ and α exponentially decays towards zero. For example, after five expansion steps, $(1-\gamma)-\alpha_5 < 1\%$. Using this procedure, the only limitations on the precision of the estimate of ς are the time step between measurements and the uncertainty due to measurement noise.

Once ς has been estimated using this procedure, it can be used in combination with $t_{\text{up}}(0)$ to specify the times when cuts occur, for fitting the milling model as described in the following section. While the estimate of ς obtained from this procedure is usually fairly close to the correct value, measurement noise can lead to larger error than would be ideal. The following section adds a parameter to allow for small errors in the estimates of $t_{\text{up}}(0)$ and ς. To allow for larger errors, it can be beneficial to also try values of ς somewhat farther away, selecting the value for which the model best fits the data. The estimates of the model parameters $p_9, \ldots, p_{18}$, obtained while fitting eq. (4.34) to the data for the chosen value of ς, are used in the following section to fit the remaining parameters.

4.4.6 Estimating the Cutting Coefficients and Mass Asymmetry

After completing the stages described in the previous sections, the only remaining parameters which have not been estimated are those describing the cutting coefficients and mass asymmetry, i.e. $p_{20}, \ldots, p_{24}$. These parameters can be estimated by fitting the full milling model, i.e. eq. (4.4), to the time series after cutting started, using the trajectory matching approach described in Section 4.4.2 with the spectral element method described in Section 4.2.3. This is illustrated in fig. 4.10.

Most of the parameter values estimated in the previous sections were used as-is. The only exception was an additional parameter for each time series to allow for small errors in the estimates of parameters p_8 and p_{19}, since small errors in the phase of the cuts can significantly affect the quality of the fit to the time series. To allow for these small errors, the orientation of the tip of tooth 1 at the estimate of $t_{\text{up}}(0)$

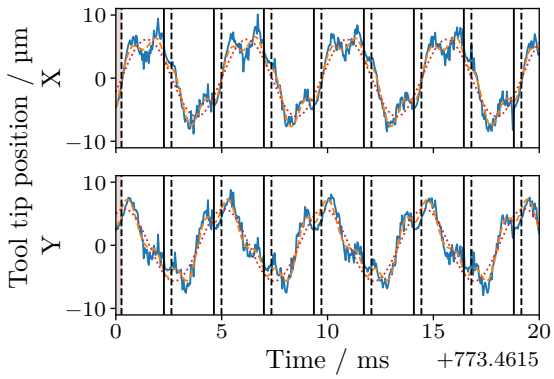


FIGURE 4.10: Example of fitting using the complete milling model (eq. (4.4)). Note that only part of one time series is shown; the model is actually fit to multiple time series simultaneously. The components in the X and Y directions are shown on separate axes. The position measurements are the solid blue lines, and the corresponding fitted positions $\boldsymbol{q}(t)$ are indicated with dashed orange lines. The start and end of each cut are indicated with solid and dashed vertical lines, respectively. The cut containing the explicitly parameterized nodes is indicated with a shaded background. The steady-state response $\boldsymbol{q}_{\text{ss}}(t)$ estimated from the data between cuts (Section 4.4.5) is shown with a dotted red line.

was represented by a shifted logistic curve:

$$\theta_1(0, \tilde{p}_8) = \tilde{p}_{19} - 0.05 \left(\frac{2\pi}{n_{\text{teeth}}} \right) \left(\frac{2}{1 + (1 + \mathrm{e}^{-p_{25}})} - 1 \right) \tag{4.86}$$

where $\tilde{p}_8$ is the value of p_8 estimated in Section 4.4.3, $\tilde{p}_{19}$ is the value of p_{19} estimated in Section 4.4.5, and p_{25} is an additional parameter for each time series.

To match the time series data, the initial conditions for each time series must be estimated simultaneously with the unknown model parameters. The initial conditions for a DDE with delay τ consist of the states $\boldsymbol{x}$ over an interval of length τ. For the approach presented here, this interval of states was represented by state values at the spectral element nodes for one cut. Specifically, the first spectral element segment was explicitly represented by the position $\boldsymbol{q}$ at each spectral element node in the segment, the velocity $\dot{\boldsymbol{q}}$ at the first node in the segment, and the velocity at the last node in the segment. The velocities between the first and last nodes were not necessary, because they affected neither the position measurements in the first segment nor the states of later segments. An initial guess for the explicitly repre-

sented nodes was obtained by fitting a polynomial of order $\left(\min\left\{n-1, \lfloor\sqrt{n}\rfloor\right\}\right)$ to the n time series measurements within the initial cut. Given the representation of the initial segment and the model parameters, an estimate of the state $\boldsymbol{x}$ at any time in the time series could be computed using the spectral element method, as described in Section 4.2.3. So, the model parameters and initial conditions could be estimated by finding the values of the parameters and initial conditions for which the computed trajectory most closely matched the time series data, as described in Section 4.4.2.

For the solving the least squares optimization problem to fit the time series, the `scipy.optimize.least_squares` SciPy library function [49] with the Trust Region Reflective algorithm was used. As in Section 4.4.5, only a subset of the parameters affected each term in the objective function, so sparse Jacobians were used to significantly speed up the optimization algorithm. Specifically, each initial condition segment and each parameter p_{25} affected only the terms in the corresponding time series. Once the parameters were estimated, the milling model could be used to make predictions. For example, the spectral element method was used to generate a stability chart for different spindle speeds and cutting depths.

4.5 Results and Discussion

The methods presented in this chapter were designed so that a machine operator could, in a few minutes, conduct a few short, low radial immersion test cuts; run the automated parameter estimation procedure; and then automatically generate a stability chart, like the one shown in fig. 4.11, using the estimated parameters. To demonstrate their applicability to a real milling system, the methods presented in this chapter were applied to time series collected using the experimental setup described in Section 4.3.

Given a set of a few time series, the methods described in Section 4.4 were

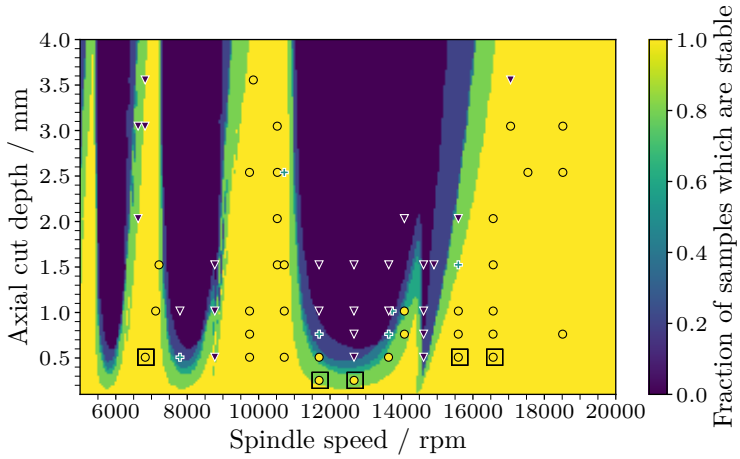
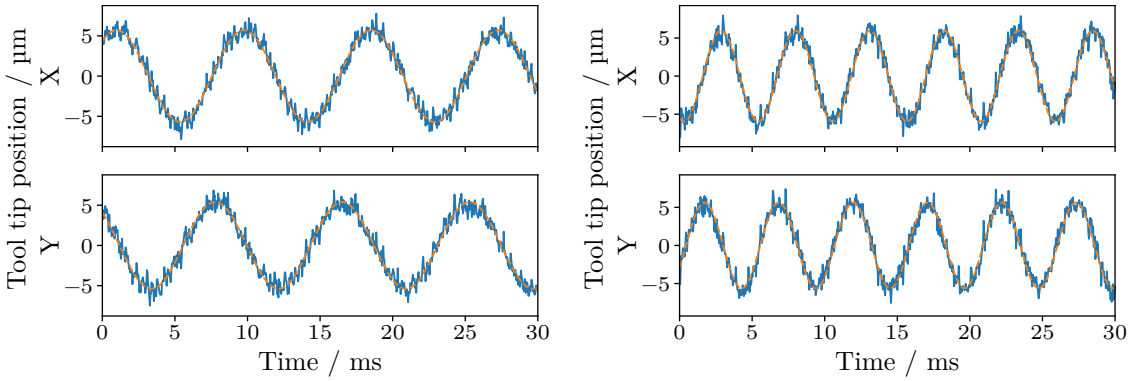


FIGURE 4.11: Stability charts for the estimated parameter values. For each point (Ω, b), the background color indicates the fraction of subsets of the data for which the estimated parameters predicted that the point would be stable. For comparison, the experimental stability assessments for this system from [60] are indicated with markers (stable: yellow circle, borderline: green cross, unstable: purple triangle). The values of Ω and b for the time series used for fitting are indicated with black boxes.

used to estimate the parameters of the dynamics model described in Section 4.2. Then, using these parameter estimates, the spectral element method described in Section 4.2.3 and Appendix E was used to compute approximate CMs over a grid of spindle speeds and axial cut depths. Each point in the grid was predicted to be stable if the magnitudes of all the CMs were less than one, and unstable otherwise. This procedure of estimating the parameters from a set of time series and predicting the stability over a grid of points was repeated for various sets of time series, in order to provide an indication of the variability in the predictions. Specifically, it was repeated for each subset of four unique time series from a set of five time series. The spindle speeds and axial cut depths for the five time series are indicated by black boxes in fig. 4.11. Note that all of these cases have low axial cut depths. Low axial cut depths are a good choice for test cuts in order to minimize the risk of damaging the tool, since the system is more likely to be stable, and even if it is unstable, the cutting forces are smaller. The grids of spectral element stability

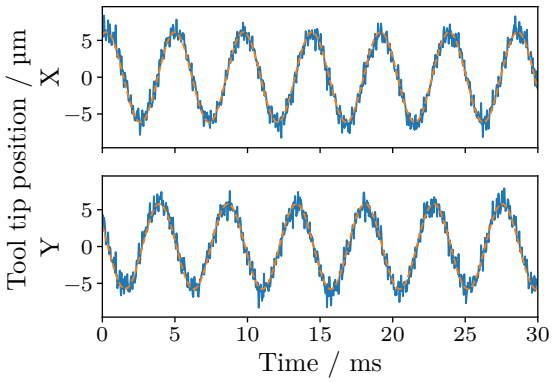
predictions corresponding to the subsets of time series were overlaid on each other to form the background color in fig. 4.11: the background color of each point indicates the fraction of the subsets of the time series for which the point was predicted to be stable. For comparison, true stability assessments obtained experimentally [60] are shown with triangular, cross-shaped, and circular markers. There is noticeable variability in the stability predictions, as indicated by the width of the transition between the background colors corresponding to the stable and unstable regions. However, they match the true stability assessments fairly closely. Given a stability chart like this, the operator could easily select a good operating point by finding a point which meets the desired criteria and is a reasonable distance from the unstable regions.

While the purpose of the methods is to provide stability predictions via parameter estimation, comparing the fitted trajectories to the experimental measurements provides some useful insight. For the same five time series which were used to generate fig. 4.11, the fitting procedure was run for all five time series together. For the fit of the data before cutting started (Section 4.4.4), fig. 4.12 shows the estimated $\boldsymbol{q}_{\mathrm{ss}}(t)$ for the data before cutting started. The good fit confirms that a single sinusoid is sufficient to represent the steady-state vibration. For the fit of the data between cuts (Section 4.4.5), fig. 4.13 shows the estimated $\boldsymbol{q}(t)$ between cuts and the updated estimate of $\boldsymbol{q}_{\mathrm{ss}}(t)$. The model for the vibration between cuts fits the data very well; this strongly supports the proposal of incorporating nonzero $\boldsymbol{f}_{\mathrm{u}}(t)$ into the model to handle cases where the tool is not perfectly centered about the axis of rotation. For fitting all the data during cutting (Section 4.4.6), fig. 4.14 shows the estimated $\boldsymbol{q}(t)$. While the model fits the most of the time series fairly well, some of the errors between the observations and fitted state are systematic, i.e. not just due to measurement noise. This suggests that the DDE model describing the dynamics during each cut could be improved further or, possibly, some of the "known" parameter

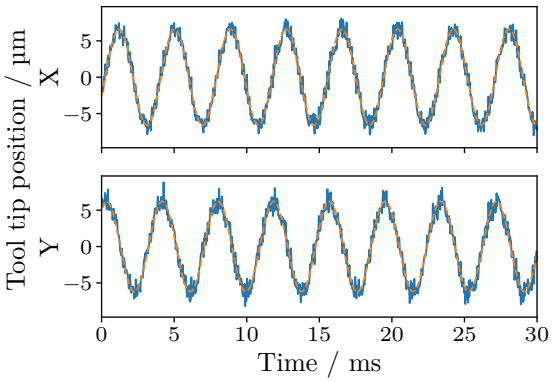


(a) $\Omega = 6842\,\text{rpm}$, $b = 0.51\,\text{mm}$.

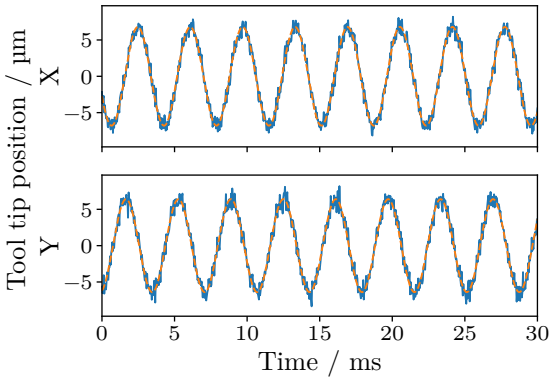
(b) $\Omega = 11\,722\,\text{rpm}$, $b = 0.25\,\text{mm}$.



(c) $\Omega = 12\,704\,\text{rpm}$, $b = 0.25\,\text{mm}$.

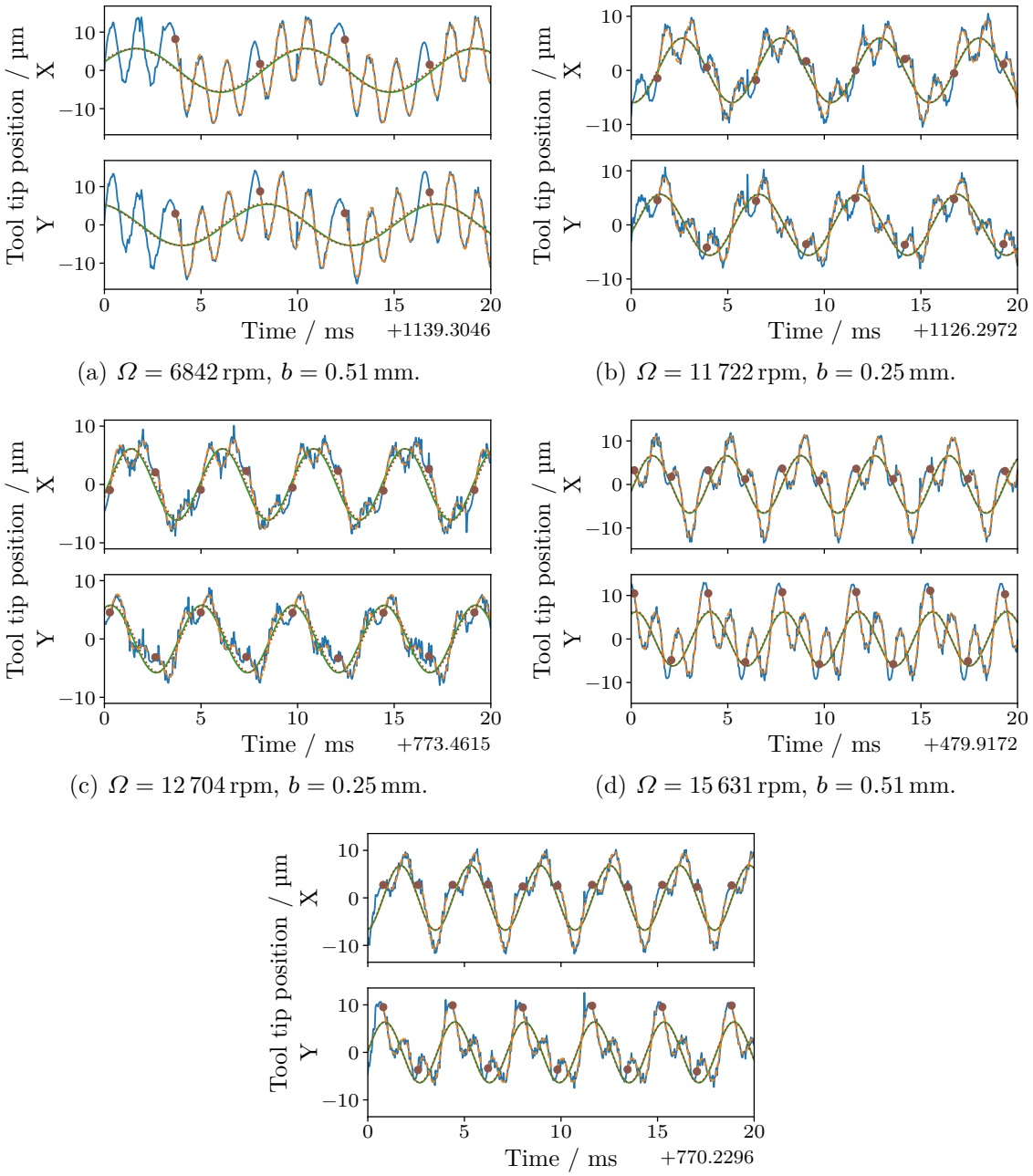


(d) $\Omega = 15\,631\,\text{rpm}$, $b = 0.51\,\text{mm}$.



(e) $\Omega = 16\,609\,\text{rpm}$, $b = 0.51\,\text{mm}$.

FIGURE 4.12: Fit to the observations before cutting started; only the first 30 ms is shown, due to space limitations. The components in the X and Y directions are shown on separate axes. The position measurements are the solid blue lines, and the corresponding fitted steady-state vibration $\boldsymbol{q}_{\text{ss}}(t)$ (Section 4.4.4) is indicated with dashed orange lines.



(a) $\Omega = 6842\,\text{rpm}$, $b = 0.51\,\text{mm}$.

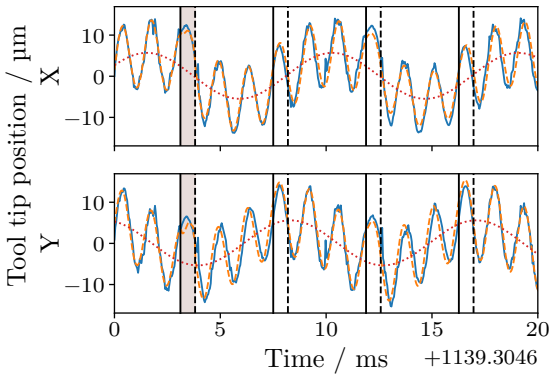
(b) $\Omega = 11\,722\,\text{rpm}$, $b = 0.25\,\text{mm}$.

(c) $\Omega = 12\,704\,\text{rpm}$, $b = 0.25\,\text{mm}$.

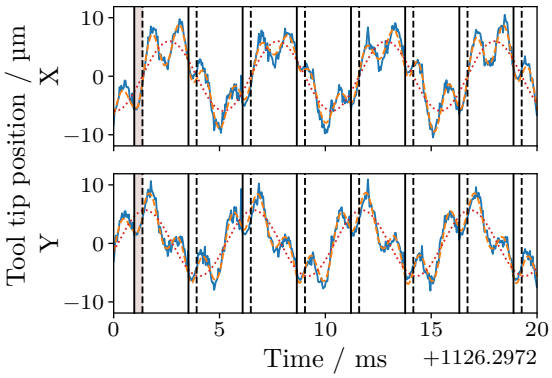
(d) $\Omega = 15\,631\,\text{rpm}$, $b = 0.51\,\text{mm}$.

(e) $\Omega = 16\,609\,\text{rpm}$, $b = 0.51\,\text{mm}$.

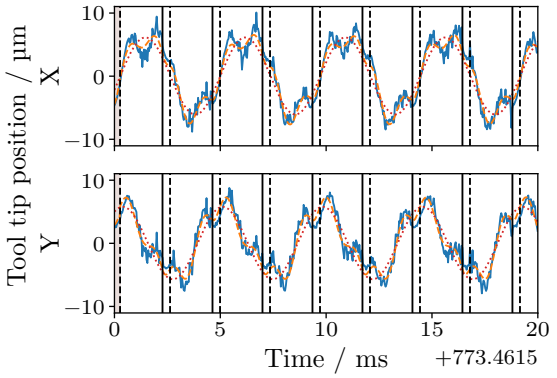
FIGURE 4.13: Fit to the observations between cuts; only the first 10% of the data is shown, due to space limitations. The components in the X and Y directions are shown on separate axes. The position measurements are the solid blue lines, and the corresponding fitted positions $\boldsymbol{q}(t)$ between cuts (Section 4.4.5) are indicated with dashed orange lines. The estimated initial condition for each interval between cuts is indicated with a brown circular marker. The steady-state response $\boldsymbol{q}_{\text{ss}}(t)$ estimated from the data before cutting started (Section 4.4.4) and the updated estimate of $\boldsymbol{q}_{\text{ss}}(t)$ from the data between cuts (Section 4.4.5) are shown with solid green and dotted red lines, respectively.



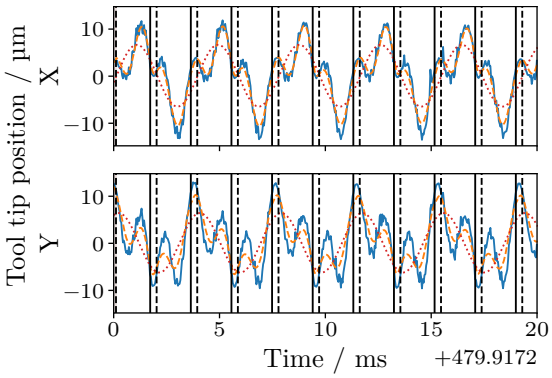
(a) $\Omega = 6842\,\text{rpm}$, $b = 0.51\,\text{mm}$.



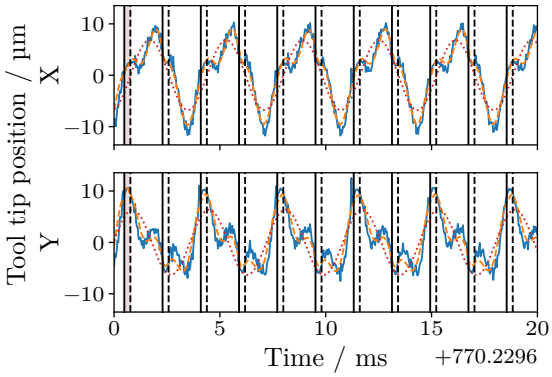
(b) $\Omega = 11\,722\,\text{rpm}$, $b = 0.25\,\text{mm}$.



(c) $\Omega = 12\,704\,\text{rpm}$, $b = 0.25\,\text{mm}$.



(d) $\Omega = 15\,631\,\text{rpm}$, $b = 0.51\,\text{mm}$.



(e) $\Omega = 16\,609\,\text{rpm}$, $b = 0.51\,\text{mm}$.

FIGURE 4.14: Fit using the complete milling model (eq. (4.4)); only the first 10% of the data is shown, due to space limitations. The components in the X and Y directions are shown on separate axes. The position measurements are the solid blue lines, and the corresponding fitted positions $\boldsymbol{q}(t)$ (Section 4.4.6) are indicated with dashed orange lines. The start and end of each cut are indicated with solid and dashed vertical lines, respectively. The cut containing the explicitly parameterized nodes is indicated with a shaded background. The steady-state response $\boldsymbol{q}_{\text{ss}}(t)$ estimated from the data between cuts (Section 4.4.5) is shown with a dotted red line.

values $(p_1, \ldots, p_6)$ were not quite correct. This also explains why some of the errors in the stability estimates in fig. 4.11 appear to be systematic, rather than due to measurement noise.

In summary, using the proposed methods, it is possible to quickly collect data from a small number of low radial immersion test cuts, using sensors attached to a milling machine, and then generate a stability chart which can be used to select appropriate parameters in the stable region. The results demonstrate that the model fits the data fairly well, although there is room for further improvement. The stability chart in fig. 4.11 is close enough to the correct stability boundary to be useful in practice. Fitting to subsets of the data provides an indication of the variability or uncertainty in the stability estimates. While interpreting the variability as uncertainty may be somewhat misleading because the errors are partially systematic, the variability information allows the operator to make a more informed decision. Given the stability chart in fig. 4.11, the operator could select a point a reasonable distance from the unstable regions which meets the operator's desired criteria.

4.6 Conclusion

This chapter described an automated method for estimating the parameters of a two-degree-of-freedom milling model from a small number of time series which could be collected using sensors attached to a milling machine. The milling model incorporated nonzero steady-state vibration of the milling tool, independent of cutting, to better match the motion of the tool in real time series. The parameters were strategically defined to be well-suited for estimation using optimization algorithms. The fitting method proceeded in stages which estimated a few parameters at a time in order to minimize the number of parameters which needed to be estimated simultaneously. Finally, the proposed method was evaluated on real experimental data. The results demonstrated that, despite some limitations of the model, the estimated sta-

bility chart matched the true stability boundary fairly closely and provided valuable information which would allow an operator to select appropriate machining settings.

5

Conclusions

This research provides significant advancements in methods for characteristic multiplier (CM) estimation and parameter estimation for dynamical systems from noisy time series data, with a particular focus on delay differential equations (DDEs) and machining applications. The primary motivation was to allow for mathematical models of milling to be applied in practice without the need for expensive, difficult, and time-consuming direct measurement of the process parameters using specialized equipment and expert engineers. This work introduced an automated process to estimate parameters and make stability predictions using time series measurements collected from sensors attached to the machine itself.

Chapter 2 investigated the limitations of existing CM estimation methods and introduced improvements to overcome those limitations. Specifically, these limitations included a need to estimate or constrain the steady-state response, bias due to ordinary least squares, and an inability to handle repeated CMs. This chapter showed that the limitations could be solved, respectively, by using an affine transition model on the state or state integral, using total least squares instead of ordinary least squares, and collecting multiple independent time series. Simulation results from a

variety of scenarios demonstrated that these improvements resulted in more accurate CM estimates than existing methods.

Chapter 3 introduced improvements to CM matching and trajectory matching methods for estimating the parameters of DDEs from time series data. For CM matching, this chapter incorporated the empirical CM estimation improvements from Chapter 2 and introduced a way to match multiple CM estimates for each time series. For trajectory matching, this chapter handled multivariate observations and prior knowledge about parameters in a principled way, used the spectral element method to provide a convenient representation of the initial interval and reduce the computational cost of computing the objective function, and fit multiple time series simultaneously. Simulation results demonstrated that these improved methods worked well in practice, although CM matching had some limitations which were not a problem for the trajectory matching method.

Finally, Chapter 4 introduced a new approach to estimate the parameters of a dynamics model for milling from time series data. Specifically, it combined and extended models from the literature into an updated model which more closely fit the time series data. It described a procedure to estimate the unknown parameters of the model in stages, so that it was not necessary to solve a global optimization algorithm for all the parameters simultaneously. The procedure incorporated extensions to the spectral element method, as described in Appendix E. Experimental results using data collected on a physical milling machine demonstrated that the proposed model and procedure successfully approximated the true stability boundaries.

5.1 Future Research Directions

While the methods presented in this work advance the state-of-the-art, further research could expand their applicability and increase their accuracy. This section describes possible future research directions corresponding to each chapter.

5.1.1 Empirical Characteristic Multiplier Estimation

This chapter proposed and evaluated various improvements to empirical CM estimation methods. Some directions for future research include the following.

1. The methods presented in this chapter assumed that the measurement sampling frequency was an integer multiple of the frequency of the limit cycle oscillation. It would be useful to investigate principled approaches to handle cases where this assumption is not met.

2. In some cases, the period of the limit cycle oscillation may be uncertain. It would be useful to investigate how to account for this uncertainty in the period when estimating the CMs.

3. For systems with significant nonlinearity, linearization about a limit cycle may be a good approximation only very close to the limit cycle. It would be useful to be able to identify these cases from the time series data. For example, correlation between the least squares residuals may indicate that the affine transition model is not a good match to the data. It may also be possible to compensate for such nonlinearities.

4. In some cases, some prior information about the dynamics of the system may be available. It would be useful to be able to incorporate prior information when estimating the CMs. For example, when a full model is available, trajectory matching can be used to estimate the model's unknown parameters, and then CMs can be computed from the model. It would be beneficial to find ways to incorporate partial prior information when a full model is unavailable.

5.1.2 Estimating Parameters of DDEs from Time Series

This chapter described improvements to CM matching and trajectory matching techniques for estimating parameters of DDE models from time series. Some directions for future research include the following.

1. Some of the CM estimates may be more certain than others. For example, the estimates of smaller CMs tend to have more uncertainty, since the components of the transient response corresponding to these CMs decay more rapidly. It would be useful to determine the uncertainty of the CM estimates and incorporate that uncertainty information into the objective function when using the CMs to estimate model parameters.

2. This chapter described state space reconstruction as one way to handle partial observability of the system state when estimating CMs. More investigation into how to best choose the components of the reconstructed state may improve the accuracy of the CM estimates.

3. For time series of chaotic behavior, the single-shooting trajectory matching approach described in this chapter may not work well, since small errors in the estimate of the initial conditions may lead to large errors in the trajectory. In these cases, it may be beneficial to use a multiple-shooting approach, i.e. splitting the time series into shorter time series for which the initial conditions are parameterized separately.

4. Some parameters, such as the time delay or phase, may be difficult to find using trajectory matching with local optimization algorithms. In these cases, it may be beneficial to investigate the use of global optimization algorithms to solve the optimization problem.

5. Extending these methods to systems with stochastic dynamics would be an interesting area for exploration.

5.1.3 Empirical Stability Prediction for Milling

This chapter combined and extended existing models for milling, adapted the spectral element method for the model, and described a procedure to estimate the model's parameters from time series data. Some directions for future research include the following.

1. This work was particularly challenging due to the limitations of the sensors. It would be useful to investigate using sensors which would provide a more detailed view of the state of the system. For example, a sensor to detect when the tool was cutting would be particularly useful, so that the phase of the cuts would not need to be estimated from the time series of position measurements. Also, it would be useful to measure the displacement at multiple heights along the tool in order to better understand its motion.

2. While the milling dynamics model used in this chapter is fairly sophisticated, it uses many approximations. Improvements to the dynamics model may allow it to better fit the time series data. For example, in some cases, it may be necessary to account for more modes of vibration.

3. The staged parameter estimation procedure described in this chapter is a good way to estimate the parameters without having to solve a difficult global optimization problem to estimate all of them simultaneously. However, it has the disadvantage that errors in the parameters estimated in earlier stages affect the estimates of the parameters estimated in later stages. Given the parameter estimates from this procedure, the estimates could be further refined in

a final stage by optimizing them all simultaneously. Since the procedure described in this chapter should provide good estimates of the parameters, a local optimization algorithm may be sufficient to refine them further.

4. It would be useful to extend this work to other machining operations, such as turning and drilling.

Appendix A

Stability of Limit Cycles

Consider a dynamical system

$$\frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}t} = \boldsymbol{f}(\boldsymbol{x}(t), t) \tag{A.1}$$

with a time-periodic equilibrium response with period T,

$$\boldsymbol{x}_{\mathrm{p}}(t) = \boldsymbol{x}_{\mathrm{p}}(t+T) \tag{A.2}$$

where $\boldsymbol{x}$ is a column vector of length d.

A.1 Dynamics of a Perturbation are a System of Linear Homogeneous ODEs

Expanding eq. (A.1) about the periodic response gives

$$\frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}t} = \boldsymbol{f}(\boldsymbol{x}_{\mathrm{p}}(t), t) + \boldsymbol{\nabla}_{\boldsymbol{x}}^{\top}\boldsymbol{f}(\boldsymbol{x}_{\mathrm{p}}(t), t)\left(\boldsymbol{x}(t) - \boldsymbol{x}_{\mathrm{p}}(t)\right) + \mathcal{O}\left(\left(\boldsymbol{x}(t) - \boldsymbol{x}_{\mathrm{p}}(t)\right)^2\right) \tag{A.3}$$

If the perturbation is defined as $\boldsymbol{\xi}(t) = \boldsymbol{x}(t) - \boldsymbol{x}_{\mathrm{p}}(t)$ and the higher-order terms are discarded, the result is

$$\frac{\mathrm{d}\left(\boldsymbol{x}_{\mathrm{p}}(t) + \boldsymbol{\xi}(t)\right)}{\mathrm{d}t} = \boldsymbol{f}(\boldsymbol{x}_{\mathrm{p}}(t), t) + \boldsymbol{\nabla}_{\boldsymbol{x}}^{\top}\boldsymbol{f}(\boldsymbol{x}_{\mathrm{p}}(t), t)\boldsymbol{\xi}(t) \tag{A.4}$$

$$\frac{\mathrm{d}\boldsymbol{x}_\mathrm{p}(t)}{\mathrm{d}t} + \frac{\mathrm{d}\boldsymbol{\xi}(t)}{\mathrm{d}t} = \frac{\mathrm{d}\boldsymbol{x}_\mathrm{p}(t)}{\mathrm{d}t} + \boldsymbol{\nabla}_{\boldsymbol{x}}^{\top}\boldsymbol{f}(\boldsymbol{x}_\mathrm{p}(t), t)\boldsymbol{\xi}(t) \tag{A.5}$$

$$\frac{\mathrm{d}\boldsymbol{\xi}(t)}{\mathrm{d}t} = \boldsymbol{\nabla}_{\boldsymbol{x}}^{\top}\boldsymbol{f}(\boldsymbol{x}_\mathrm{p}(t), t)\boldsymbol{\xi}(t) \tag{A.6}$$

Define $\boldsymbol{A}(t) = \boldsymbol{\nabla}_{\boldsymbol{x}}^{\top}\boldsymbol{f}(\boldsymbol{x}_\mathrm{p}(t), t)$, then

$$\frac{\mathrm{d}\boldsymbol{\xi}(t)}{\mathrm{d}t} = \boldsymbol{A}(t)\boldsymbol{\xi}(t) \tag{A.7}$$

where $\boldsymbol{A}(t)$ is time-periodic with period T, i.e. $\boldsymbol{A}(t) = \boldsymbol{A}(t+T)$, which can be shown as follows:

$$\boldsymbol{x}_\mathrm{p}(t) = \boldsymbol{x}_\mathrm{p}(t+T) \tag{A.8}$$

$$\frac{\mathrm{d}\boldsymbol{x}_\mathrm{p}(t)}{\mathrm{d}t} = \frac{\mathrm{d}\boldsymbol{x}_\mathrm{p}(t+T)}{\mathrm{d}t} \tag{A.9}$$

$$\boldsymbol{f}(\boldsymbol{x}_\mathrm{p}(t), t) = \boldsymbol{f}(\boldsymbol{x}_\mathrm{p}(t+T), t+T) \tag{A.10}$$

$$\boldsymbol{\nabla}_{\boldsymbol{x}}^{\top}\boldsymbol{f}(\boldsymbol{x}_\mathrm{p}(t), t) = \boldsymbol{\nabla}_{\boldsymbol{x}}^{\top}\boldsymbol{f}(\boldsymbol{x}_\mathrm{p}(t+T), t+T) \tag{A.11}$$

$$\boldsymbol{A}(t) = \boldsymbol{A}(t+T) \tag{A.12}$$

A.2 Floquet Theory

A fundamental matrix $\boldsymbol{\Phi}(t)$ for a system of d linear homogeneous differential equations, such as eq. (A.7), is a matrix-valued function

$$\boldsymbol{\Phi}(t) = \begin{bmatrix} \boldsymbol{\xi}^{(1)}(t) & \boldsymbol{\xi}^{(2)}(t) & \cdots & \boldsymbol{\xi}^{(d)}(t) \end{bmatrix} \tag{A.13}$$

where $\boldsymbol{\xi}^{(1)}(t), \ldots, \boldsymbol{\xi}^{(d)}(t)$ are linearly-independent solutions to eq. (A.7). By eq. (A.7), a fundamental matrix satisfies

$$\dot{\boldsymbol{\Phi}}(t) = \boldsymbol{A}(t)\boldsymbol{\Phi}(t) \tag{A.14}$$

Quoting from [1], by Floquet's theorem, if $\boldsymbol{\Phi}(t)$ is a fundamental matrix of the system described by eqs. (A.7) and (A.12), then $\boldsymbol{\Phi}(t+T)$ is also a fundamental

matrix. Moreover, for every such $\boldsymbol{\Phi}(t)$ there exists a periodic nonsingular matrix $\boldsymbol{Q}(t)$ with period T and a constant matrix $\boldsymbol{R}$ such that

$$\boldsymbol{\Phi}(t) = \boldsymbol{Q}(t)\mathrm{e}^{t\boldsymbol{R}}, \quad \boldsymbol{Q}(t) = \boldsymbol{Q}(t+T) \tag{A.15}$$

Then, the transition from one period to the next is given by

$$\boldsymbol{\Phi}(t+T) = \boldsymbol{Q}(t+T)\mathrm{e}^{(t+T)\boldsymbol{R}} \tag{A.16}$$

$$= \boldsymbol{Q}(t+T)\mathrm{e}^{t\boldsymbol{R}}\mathrm{e}^{T\boldsymbol{R}} \tag{A.17}$$

$$= \boldsymbol{Q}(t)\mathrm{e}^{t\boldsymbol{R}}\mathrm{e}^{T\boldsymbol{R}} \tag{A.18}$$

$$= \boldsymbol{\Phi}(t)\mathrm{e}^{T\boldsymbol{R}} \tag{A.19}$$

The matrix $\mathrm{e}^{T\boldsymbol{R}}$ is known as the monodromy matrix of the fundamental matrix $\boldsymbol{\Phi}(t)$. Its eigenvalues are known as characteristic multipliers, and they describe the stability of the system. It can be shown via proof by induction that

$$\boldsymbol{\Phi}(t+jT) = \boldsymbol{\Phi}(t)\big(\mathrm{e}^{T\boldsymbol{R}}\big)^{j}, \quad j = 0, 1, ... \tag{A.20}$$

Let $\mathrm{e}^{T\boldsymbol{R}} = \boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^{-1}$, where $\boldsymbol{W}$ is the invertible matrix of eigenvectors of $\mathrm{e}^{T\boldsymbol{R}}$ and $\boldsymbol{M}$ is the diagonal matrix of eigenvalues of $\mathrm{e}^{T\boldsymbol{R}}$, i.e. the characteristic multipliers. Then,

$$\boldsymbol{\Phi}(t+jT) = \boldsymbol{\Phi}(t)\big(\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^{-1}\big)^{j}, \quad j = 0, 1, ... \tag{A.21}$$

$$\boldsymbol{\Phi}(t+jT) = \boldsymbol{\Phi}(t)\boldsymbol{W}\boldsymbol{M}^{j}\boldsymbol{W}^{-1}, \quad j = 0, 1, ... \tag{A.22}$$

$$\boldsymbol{\Phi}(t+jT)\boldsymbol{W} = \boldsymbol{\Phi}(t)\boldsymbol{W}\boldsymbol{M}^{j}, \quad j = 0, 1, ... \tag{A.23}$$

This result shows that the system is stable if and only if all the characteristic multipliers have magnitude less than 1.

While the monodromy matrix is dependent on the fundamental matrix chosen, the characteristic multipliers are the same for all choices of the fundamental matrix. This can be shown as follows. Given a fundamental matrix $\boldsymbol{\Phi}(t)$, any other fundamental

matrix $\boldsymbol{\Phi}'(t)$ can be expressed as $\boldsymbol{\Phi}'(t) = \boldsymbol{\Phi}(t)\boldsymbol{B}$ where $\boldsymbol{B}$ is a constant nonsingular matrix. Rearranging and substituting into eq. (A.19) gives

$$\boldsymbol{\Phi}'(t+T)\boldsymbol{B}^{-1} = \boldsymbol{\Phi}'(t)\boldsymbol{B}^{-1}\mathrm{e}^{T\boldsymbol{R}} \iff \boldsymbol{\Phi}'(t+T) = \boldsymbol{\Phi}'(t)\boldsymbol{B}^{-1}\mathrm{e}^{T\boldsymbol{R}}\boldsymbol{B} \tag{A.24}$$

from which it can be seen that the monodromy matrix of $\boldsymbol{\Phi}'(t)$, i.e. $\boldsymbol{B}^{-1}\mathrm{e}^{T\boldsymbol{R}}\boldsymbol{B}$, is similar to the monodromy matrix of $\boldsymbol{\Phi}(t)$, i.e. $\mathrm{e}^{T\boldsymbol{R}}$, and so it has the same eigenvalues.

A.3 Estimating a Monodromy Matrix

The question now is how to estimate a monodromy matrix from data. Consider a set $\left\{\boldsymbol{\Phi}^{(k)}(t) \mid k = 0, 1, ...\right\}$ of fundamental matrix solutions corresponding to a set of time points $\{t_k \mid k = 0, 1, ...\}$ which have the property that $\boldsymbol{\Phi}^{(k)}(t_k) = \boldsymbol{I}$. Let their corresponding monodromy matrices be $\mathrm{e}^{T\boldsymbol{R}^{(k)}}$. It can be shown that for any solution $\boldsymbol{\xi}^{(i)}(t)$ to eq. (A.7) and fundamental matrix $\boldsymbol{\Phi}^{(k)}$, there exists a constant vector $\boldsymbol{c}^{(k,i)}$ such that

$$\boldsymbol{\xi}^{(i)}(t) = \boldsymbol{\Phi}^{(k)}(t)\boldsymbol{c}^{(k,i)} \tag{A.25}$$

So, by substitution,

$$\boldsymbol{\xi}^{(i)}(t_k) = \boldsymbol{\Phi}^{(k)}(t_k)\boldsymbol{c}^{(k,i)} = \boldsymbol{I}\boldsymbol{c}^{(k,i)} = \boldsymbol{c}^{(k,i)} \implies \boldsymbol{\xi}^{(i)}(t) = \boldsymbol{\Phi}^{(k)}(t)\boldsymbol{\xi}^{(i)}(t_k) \tag{A.26}$$

Then, in subsequent periods, by eqs. (A.20) and (A.26),

$$\boldsymbol{\xi}^{(i)}(t_k + jT) = \boldsymbol{\Phi}^{(k)}(t_k + jT)\boldsymbol{\xi}^{(i)}(t_k), \qquad j = 0, 1, ... \tag{A.27}$$

$$= \boldsymbol{\Phi}^{(k)}(t_k)\left(\mathrm{e}^{T\boldsymbol{R}^{(k)}}\right)^j\boldsymbol{\xi}^{(i)}(t_k), \qquad j = 0, 1, ... \tag{A.28}$$

$$= \left(\mathrm{e}^{T\boldsymbol{R}^{(k)}}\right)^j\boldsymbol{\xi}^{(i)}(t_k), \qquad j = 0, 1, ... \tag{A.29}$$

It then can be shown via proof by induction that

$$\boldsymbol{\xi}^{(i)}(t_k + (j+1)T) = \mathrm{e}^{T\boldsymbol{R}^{(k)}}\boldsymbol{\xi}^{(i)}(t_k + jT), \quad j = 0, 1, ... \tag{A.30}$$

In other words, using the notation from eq. (2.3),

$$\boldsymbol{P}(t_k) = \mathrm{e}^{T\boldsymbol{R}^{(k)}} \tag{A.31}$$

In order to compute the characteristic multipliers, it's sufficient to compute the eigenvalues of $\mathrm{e}^{T\boldsymbol{R}^{(k)}}$ for a single value of k, since, as described in the previous section, the characteristic multipliers are identical for all choices of fundamental matrices. If there is no noise and the vectors $\left\{\boldsymbol{\xi}^{(i)}(t_k + jT) \mid j = 0, 1, ...\right\}$ are linearly independent, then it is sufficient to collect d pairs of values at a single phase from a single time series into square matrices:

$$\boldsymbol{\Xi}_{k,i,0} = \begin{bmatrix}\boldsymbol{\xi}^{(i)}(t_k) & \boldsymbol{\xi}^{(i)}(t_k + T) & \cdots & \boldsymbol{\xi}^{(i)}(t_k + (d-1)T)\end{bmatrix} \tag{A.32}$$

$$\boldsymbol{\Xi}_{k,i,1} = \begin{bmatrix}\boldsymbol{\xi}^{(i)}(t_k + T) & \boldsymbol{\xi}^{(i)}(t_k + 2T) & \cdots & \boldsymbol{\xi}^{(i)}(t_k + dT)\end{bmatrix} \tag{A.33}$$

$$\boldsymbol{\Xi}_{k,i,1} = \mathrm{e}^{T\boldsymbol{R}^{(k)}}\boldsymbol{\Xi}_{k,i,0} \tag{A.34}$$

and then compute the monodromy matrix with $\mathrm{e}^{T\boldsymbol{R}^{(k)}} = \boldsymbol{\Xi}_{k,i,1}{\boldsymbol{\Xi}_{k,i,0}}^{-1}$. In the presence of noise, it is common to collect additional pairs so that the monodromy matrix can be estimated by least squares. If the vectors from a single time series are linearly dependent, multiple time series, i.e. multiple values of i, must be collected as described in Section 2.3.3. Note that all the samples must be at the same phase t_k, since $\mathrm{e}^{T\boldsymbol{R}^{(k)}}$ is a function of k.

A.4 Linear Combinations Preserve Characteristic Multipliers

Little et al. [19] show that a moving integral of the state preserves the characteristic multipliers (CMs) when the mean of the steady-state response is zero. However, when working with experimental data, the integral can only be computed numerically, and the mean of the steady-state response is often nonzero. As a result, it is useful to demonstrate that the CMs are preserved by linear combinations and to show how to handle the case when the mean of the steady-state response is nonzero.

Define

$$\bar{\boldsymbol{x}}(t) = \sum_{\tau \in \mathcal{T}} \kappa(\tau) \boldsymbol{x}(t+\tau) \tag{A.35}$$

$$\bar{\boldsymbol{x}}_{\mathrm{p}}(t) = \sum_{\tau \in \mathcal{T}} \kappa(\tau) \boldsymbol{x}_{\mathrm{p}}(t+\tau) \tag{A.36}$$

where $\mathcal{T}$ is a set of time offsets and κ is a scalar coefficient function. For example, $\bar{\boldsymbol{x}}(t)$ is a left Riemann sum of $\boldsymbol{x}(t)$ over one period if $\mathcal{T} = \{iT/n \mid i = 0, 1, \ldots, n-1\}$ and $\kappa(\tau) = T/n$, but different choices of $\mathcal{T}$ and κ can be used to express a wide variety of other numerical integration rules. Expanding eq. (A.35),

$$\bar{\boldsymbol{x}}(t) = \sum_{\tau \in \mathcal{T}} \kappa(\tau) \boldsymbol{x}(t+\tau) \tag{A.37}$$

$$= \sum_{\tau \in \mathcal{T}} \kappa(\tau) \left[\boldsymbol{x}_{\mathrm{p}}(t+\tau) + \boldsymbol{\xi}(t+\tau) \right] \tag{A.38}$$

$$= \sum_{\tau \in \mathcal{T}} \kappa(\tau) \boldsymbol{x}_{\mathrm{p}}(t+\tau) + \sum_{\tau \in \mathcal{T}} \kappa(\tau) \boldsymbol{\xi}(t+\tau) \tag{A.39}$$

Rearranging and substituting,

$$\bar{\boldsymbol{x}}(t) - \bar{\boldsymbol{x}}_{\mathrm{p}}(t) = \sum_{\tau \in \mathcal{T}} \kappa(\tau) \boldsymbol{\xi}(t+\tau) \tag{A.40}$$

If $\boldsymbol{\Phi}(t)$ is a fundamental matrix solution, then it can be shown that there exists a constant vector $\boldsymbol{c}$ such that the perturbation can be written $\boldsymbol{\xi}(t) = \boldsymbol{\Phi}(t)\boldsymbol{c}$. Substituting this into eq. (A.40) and applying eq. (A.15),

$$\bar{\boldsymbol{x}}(t) - \bar{\boldsymbol{x}}_{\mathrm{p}}(t) = \sum_{\tau \in \mathcal{T}} \kappa(\tau) \boldsymbol{\Phi}(t+\tau) \boldsymbol{c} \tag{A.41}$$

$$= \sum_{\tau \in \mathcal{T}} \kappa(\tau) \left(\boldsymbol{Q}(t+\tau) \mathrm{e}^{(t+\tau)\boldsymbol{R}} \right) \boldsymbol{c} \tag{A.42}$$

$$= \left(\sum_{\tau \in \mathcal{T}} \kappa(\tau) \boldsymbol{Q}(t+\tau) \mathrm{e}^{(t+\tau)\boldsymbol{R}} \right) \boldsymbol{c} \tag{A.43}$$

$$= \left(\sum_{\tau \in \mathcal{T}} \kappa(\tau) \boldsymbol{Q}(t+\tau) \mathrm{e}^{(t+\tau)\boldsymbol{R}} \right) \mathrm{e}^{-t\boldsymbol{R}} \mathrm{e}^{t\boldsymbol{R}} \boldsymbol{c} \tag{A.44}$$

$$= \boldsymbol{\Psi}(t)\mathrm{e}^{t\boldsymbol{R}}\boldsymbol{c} \tag{A.45}$$

where

$$\boldsymbol{\Psi}(t) = \boldsymbol{G}(t)\mathrm{e}^{-t\boldsymbol{R}} \tag{A.46}$$

$$\boldsymbol{G}(t) = \sum_{\tau\in\mathcal{T}} \kappa(\tau)\boldsymbol{Q}(t+\tau)\mathrm{e}^{(t+\tau)\boldsymbol{R}} \tag{A.47}$$

It follows that

$$\boldsymbol{c} = \mathrm{e}^{-t\boldsymbol{R}}\boldsymbol{\Psi}(t)^{-1}\left(\bar{\boldsymbol{x}}(t) - \bar{\boldsymbol{x}}_{\mathrm{p}}(t)\right) \tag{A.48}$$

After one period,

$$\bar{\boldsymbol{x}}(t+T) - \bar{\boldsymbol{x}}_{\mathrm{p}}(t+T) = \boldsymbol{\Psi}(t+T)\mathrm{e}^{(t+T)\boldsymbol{R}}\boldsymbol{c} \tag{A.49}$$

$$= \boldsymbol{\Psi}(t+T)\mathrm{e}^{T\boldsymbol{R}}\mathrm{e}^{t\boldsymbol{R}}\boldsymbol{c} \tag{A.50}$$

$$= \boldsymbol{\Psi}(t+T)\mathrm{e}^{T\boldsymbol{R}}\mathrm{e}^{t\boldsymbol{R}}\left[\mathrm{e}^{-t\boldsymbol{R}}\boldsymbol{\Psi}(t)^{-1}\left(\bar{\boldsymbol{x}}(t) - \bar{\boldsymbol{x}}_{\mathrm{p}}(t)\right)\right] \tag{A.51}$$

$$= \boldsymbol{\Psi}(t+T)\mathrm{e}^{T\boldsymbol{R}}\boldsymbol{\Psi}(t)^{-1}\left(\bar{\boldsymbol{x}}(t) - \bar{\boldsymbol{x}}_{\mathrm{p}}(t)\right) \tag{A.52}$$

which can be written as

$$\bar{\boldsymbol{x}}(t+T) - \bar{\boldsymbol{x}}_{\mathrm{p}}(t+T) = \bar{\boldsymbol{P}}(t)\left(\bar{\boldsymbol{x}}(t) - \bar{\boldsymbol{x}}_{\mathrm{p}}(t)\right) \tag{A.53}$$

where $\bar{\boldsymbol{P}}(t)$ is the transition matrix for $\bar{\boldsymbol{x}}(t) - \bar{\boldsymbol{x}}_{\mathrm{p}}(t)$ given by

$$\bar{\boldsymbol{P}}(t) = \boldsymbol{\Psi}(t+T)\mathrm{e}^{T\boldsymbol{R}}\boldsymbol{\Psi}(t)^{-1} \tag{A.54}$$

It is clear from this expression that if $\boldsymbol{\Psi}(t)$ was periodic (i.e. $\boldsymbol{\Psi}(t+T) = \boldsymbol{\Psi}(t)$), then $\bar{\boldsymbol{P}}(t)$ and $\mathrm{e}^{T\boldsymbol{R}}$ would be similar matrices and have identical eigenvalues. To show that this is, in fact, true, the first step is to find another expression for $\boldsymbol{G}(t+T)$,

$$\boldsymbol{G}(t+T) = \sum_{\tau\in\mathcal{T}} \kappa(\tau)\boldsymbol{Q}(t+\tau+T)\mathrm{e}^{(t+\tau+T)\boldsymbol{R}} \tag{A.55}$$

$$= \left(\sum_{\tau\in\mathcal{T}} \kappa(\tau)\boldsymbol{Q}(t+\tau+T)\mathrm{e}^{(t+\tau)\boldsymbol{R}}\right)\mathrm{e}^{T\boldsymbol{R}} \tag{A.56}$$

$$= \left(\sum_{\tau \in \mathcal{T}} \kappa(\tau) \boldsymbol{Q}(t+\tau) \mathrm{e}^{(t+\tau)\boldsymbol{R}} \right) \mathrm{e}^{T\boldsymbol{R}} \tag{A.57}$$

where the periodicity of $\boldsymbol{Q}(t)$ was enforced in the last step. Substituting eq. (A.47), the result is

$$\boldsymbol{G}(t+T) = \boldsymbol{G}(t)\mathrm{e}^{T\boldsymbol{R}} \tag{A.58}$$

Now, to find an expression for $\boldsymbol{\Psi}(t+T)$, substituting into eq. (A.46) and applying eq. (A.58) yields

$$\boldsymbol{\Psi}(t+T) = \boldsymbol{G}(t+T)\mathrm{e}^{-(t+T)\boldsymbol{R}} \tag{A.59}$$

$$= \boldsymbol{G}(t+T)\mathrm{e}^{-T\boldsymbol{R}}\mathrm{e}^{-t\boldsymbol{R}} \tag{A.60}$$

$$= \boldsymbol{G}(t)\mathrm{e}^{T\boldsymbol{R}}\mathrm{e}^{-T\boldsymbol{R}}\mathrm{e}^{-t\boldsymbol{R}} \tag{A.61}$$

$$= \boldsymbol{G}(t)\mathrm{e}^{-t\boldsymbol{R}} \tag{A.62}$$

which by eq. (A.46) shows that

$$\boldsymbol{\Psi}(t+T) = \boldsymbol{\Psi}(t) \tag{A.63}$$

and therefore by eq. (A.54),

$$\bar{\boldsymbol{P}}(t) = \boldsymbol{\Psi}(t)\mathrm{e}^{T\boldsymbol{R}}\boldsymbol{\Psi}(t)^{-1} \tag{A.64}$$

Therefore, $\bar{\boldsymbol{P}}(t)$ and $\mathrm{e}^{T\boldsymbol{R}}$ are similar matrices and have identical eigenvalues, so the change of basis from $\boldsymbol{\xi}(t) = \boldsymbol{x}(t) - \boldsymbol{x}_{\mathrm{p}}(t)$ to $\bar{\boldsymbol{x}}(t) - \bar{\boldsymbol{x}}_{\mathrm{p}}(t)$ preserves the characteristic multipliers. In other words, when applying any of the CM estimation methods, $\bar{\boldsymbol{x}}(t) - \bar{\boldsymbol{x}}_{\mathrm{p}}(t)$ can be used analogously to $\boldsymbol{x}(t) - \boldsymbol{x}_{\mathrm{p}}(t)$.

Appendix B

Details of Higher-Dimensional System for Evaluating CM Estimation

The higher-dimensional system considered in Section 2.4.2 is a linear system with time-periodic coefficients of period 1. The dynamics of the system are

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}(t)\boldsymbol{x}(t) + \boldsymbol{g}(t) \tag{B.1}$$

where

$$\boldsymbol{A}(t) = \left(\dot{\boldsymbol{Q}}(t) + \boldsymbol{Q}(t)\boldsymbol{R}\right)\left(\boldsymbol{Q}(t)\right)^{-1} \tag{B.2}$$

$$\boldsymbol{g}(t) = -\boldsymbol{A}(t)\boldsymbol{x}_{\mathrm{p}}(t) + \dot{\boldsymbol{x}}_{\mathrm{p}}(t) \tag{B.3}$$

$$\boldsymbol{R} = \boldsymbol{W}\left(\ln \boldsymbol{M}\right)\boldsymbol{W}^{-1} \tag{B.4}$$

The elements of the steady-state periodic solution $\boldsymbol{x}_{\mathrm{p}}(t)$ and the periodic matrix $\boldsymbol{Q}(t)$ are described by Fourier series:

$$\left[\boldsymbol{x}_{\mathrm{p}}\right]_i = \tfrac{1}{2}\left[\boldsymbol{d}_0\right]_i + \sum_{k=1}^{3}\left[\boldsymbol{d}_{\mathrm{a},k}\right]_i \cos\left(k\Omega t - \left[\boldsymbol{d}_{\mathrm{p},k}\right]_i\right) \tag{B.5}$$

$$\left[\boldsymbol{Q}(t)\right]_{i,j} = \tfrac{1}{2}\left[\boldsymbol{D}_0\right]_{i,j} + \sum_{k=1}^{3}\left[\boldsymbol{D}_{\mathrm{a},k}\right]_{i,j} \cos\left(k\Omega t - \left[\boldsymbol{D}_{\mathrm{p},k}\right]_{i,j}\right) \tag{B.6}$$

The parameter values are:

$$\Omega = 2\pi$$

$$\boldsymbol{M} = \begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 0.75\mathrm{e}^{\mathrm{i}} & 0 & 0 \\ 0 & 0 & 0.75\mathrm{e}^{-\mathrm{i}} & 0 \\ 0 & 0 & 0 & 0.9 \end{bmatrix}$$

$$\boldsymbol{W} = \begin{bmatrix} 0.02 & 0.31\mathrm{e}^{0.31\mathrm{i}} & 0.31\mathrm{e}^{-0.31\mathrm{i}} & 0.58 \\ 0.90 & 0.42\mathrm{e}^{-2.97\mathrm{i}} & 0.42\mathrm{e}^{2.97\mathrm{i}} & -0.39 \\ -0.71 & 0.83\mathrm{e}^{1.59\mathrm{i}} & 0.83\mathrm{e}^{-1.59\mathrm{i}} & -0.09 \\ 0.90 & 0.41 & 0.41 & -0.73 \end{bmatrix}$$

$$\boldsymbol{d}_0 = \begin{bmatrix} -0.37 \\ -0.37 \\ 0.15 \\ 0.94 \end{bmatrix} \qquad \boldsymbol{d}_{\mathrm{a},1} = \begin{bmatrix} 0.77 \\ 0.60 \\ 0.50 \\ 0.21 \end{bmatrix} \qquad \boldsymbol{d}_{\mathrm{a},2} = \begin{bmatrix} 0.40 \\ 0.46 \\ 0.04 \\ 0.07 \end{bmatrix} \qquad \boldsymbol{d}_{\mathrm{a},3} = \begin{bmatrix} 0.19 \\ 0.17 \\ 0.12 \\ 0.13 \end{bmatrix}$$

$$\boldsymbol{d}_{\mathrm{p},1} = \begin{bmatrix} 1.79 \\ 0.16 \\ -0.62 \\ -2.36 \end{bmatrix} \qquad \boldsymbol{d}_{\mathrm{p},2} = \begin{bmatrix} -1.29 \\ -2.21 \\ -1.29 \\ 1.47 \end{bmatrix} \qquad \boldsymbol{d}_{\mathrm{p},3} = \begin{bmatrix} 1.69 \\ 2.92 \\ 2.18 \\ -1.96 \end{bmatrix}$$

$$\boldsymbol{D}_0 = \begin{bmatrix} -0.097 & -0.297 & -0.238 & 0.25 \\ -0.22 & -0.015 & 0.481 & 0.462 \\ 0.225 & 0.041 & -0.223 & -0.339 \\ 0.47 & 0.016 & -0.384 & 0.123 \end{bmatrix}$$

$$\boldsymbol{D}_{\mathrm{a},1} = \begin{bmatrix} 0.194 & 0.01 & 0.016 & 0.148 \\ 0.127 & 0.037 & 0.197 & 0.048 \\ 0.215 & 0.069 & 0.18 & 0.054 \\ 0.241 & 0.224 & 0.006 & 0.207 \end{bmatrix} \qquad \boldsymbol{D}_{\mathrm{p},1} = \begin{bmatrix} -1.60 & 2.08 & -2.11 & 1.20 \\ -3.10 & -2.48 & 1.42 & 2.31 \\ -0.99 & 3.12 & -2.68 & 1.24 \\ -0.50 & 0.54 & -0.85 & -2.45 \end{bmatrix}$$

$$\boldsymbol{D}_{\mathrm{a},2} = \begin{bmatrix} 0.077 & 0.066 & 0.08 & 0.033 \\ 0.064 & 0.102 & 0.024 & 0.010 \\ 0.110 & 0.001 & 0.104 & 0.080 \\ 0.019 & 0.053 & 0.084 & 0.111 \end{bmatrix} \qquad \boldsymbol{D}_{\mathrm{p},2} = \begin{bmatrix} 1.69 & -2.75 & -0.78 & -2.02 \\ -1.49 & 0.84 & 0.97 & 0.83 \\ 0.27 & -1.61 & -1.52 & -2.33 \\ 1.04 & 2.13 & -0.32 & -1.86 \end{bmatrix}$$

$$\boldsymbol{D}_{\mathrm{a},3} = \begin{bmatrix} 0.057 & 0.029 & 0.053 & 0.052 \\ 0.047 & 0.043 & 0.050 & 0.053 \\ 0.029 & 0.040 & 0.018 & 0.050 \\ 0.030 & 0.037 & 0.057 & 0.041 \end{bmatrix} \qquad \boldsymbol{D}_{\mathrm{p},3} = \begin{bmatrix} -1.81 & 2.05 & -1.15 & -0.65 \\ -0.50 & -0.75 & -0.43 & 1.95 \\ -1.91 & -1.53 & 1.65 & -0.78 \\ -0.28 & 1.42 & -0.83 & -1.36 \end{bmatrix}$$

Appendix C

Maximum a Posteriori Estimation as a Least Squares Problem

Consider maximum a posteriori (MAP) estimation of unknown parameters of an ordinary differential equation (ODE) or delay differential equation (DDE) model given noisy measurements of the system's state. Assume that time series of noisy measurements are available, where each measurement $\boldsymbol{z}_j^{[i]}$ in time series i is a known function of the system state $\boldsymbol{x}_j^{[i]}$ with additive Gaussian noise,

$$\boldsymbol{z}_j^{[i]} = \boldsymbol{h}\left(\boldsymbol{x}_j^{[i]}\right) + \boldsymbol{\epsilon}_j^{[i]}, \quad \boldsymbol{\epsilon}_j^{[i]} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}) \tag{C.1}$$

and that all the noise vectors $\boldsymbol{\epsilon}_j^{[i]}$ for all the samples are mutually independent. In other words, assume a multivariate normal density for each measurement,

$$p\left(\boldsymbol{z}_j^{[i]} \mid \boldsymbol{x}_j^{[i]}\right) = \left((2\pi)^{d_{\boldsymbol{z}}} \det \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\left(\boldsymbol{z}_j^{[i]} - \boldsymbol{h}\left(\boldsymbol{x}_j^{[i]}\right)\right)^{\top} \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}^{-1} \left(\boldsymbol{z}_j^{[i]} - \boldsymbol{h}\left(\boldsymbol{x}_j^{[i]}\right)\right)\right) \tag{C.2}$$

where $d_{\boldsymbol{z}}$ is the number of elements in each $\boldsymbol{z}_j^{[i]}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}$ is the covariance of the noise.

Let $\boldsymbol{\theta}$ be a vector containing all of the unknown model parameters and unknown

initial conditions necessary to approximately describe the trajectories of the system corresponding to the time series. Let $\boldsymbol{y}_j^{[i]}$ be the point on the i^{th} approximate trajectory which corresponds to $\boldsymbol{x}_j^{[i]}$. For ODE models, the initial condition at a single instant in time is sufficient for each trajectory, and the trajectories can be obtained numerically or analytically. For DDE models, the spectral element method provides a convenient way to represent the initial segment of each trajectory and compute the approximate trajectories. In summary, each $\boldsymbol{y}_j^{[i]}$ approximating $\boldsymbol{x}_j^{[i]}$ is a known function of $\boldsymbol{\theta}$:

$$\boldsymbol{x}_j^{[i]} \approx \boldsymbol{y}_j^{[i]} = \boldsymbol{f}_j^{[i]}(\boldsymbol{\theta}) \tag{C.3}$$

Assume also that the prior knowledge of $\boldsymbol{\theta}$ can be expressed as a multivariate normal density, $\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}})$, i.e.

$$p(\boldsymbol{\theta}) = \left((2\pi)^{d_{\boldsymbol{\theta}}} \det \boldsymbol{\Sigma}_{\boldsymbol{\theta}}\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\mu}_{\boldsymbol{\theta}})^{\top} \boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1} (\boldsymbol{\theta} - \boldsymbol{\mu}_{\boldsymbol{\theta}})\right) \tag{C.4}$$

where $d_{\boldsymbol{\theta}}$ is the number of elements in $\boldsymbol{\theta}$.

The objective is to find the value of $\boldsymbol{\theta}$ at the maximum posterior density:

$$\underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, p\left(\boldsymbol{\theta} \mid \boldsymbol{z}_1^{[1]}, ...\right) \tag{C.5}$$

Applying Bayes' theorem, this is equivalent to

$$\underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \frac{p\left(\boldsymbol{z}_1^{[1]}, ... \mid \boldsymbol{\theta}\right) p\left(\boldsymbol{\theta}\right)}{\int_{\boldsymbol{\Theta}} p\left(\boldsymbol{z}_1^{[1]}, ... \mid \boldsymbol{\theta}'\right) p\left(\boldsymbol{\theta}'\right) \mathrm{d}\boldsymbol{\theta}'} \tag{C.6}$$

which is equivalent to

$$\underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, p\left(\boldsymbol{z}_1^{[1]}, ... \mid \boldsymbol{\theta}\right) p\left(\boldsymbol{\theta}\right) \tag{C.7}$$

since the denominator of eq. (C.6) is always positive and does not depend on $\boldsymbol{\theta}$. Using eq. (C.3), since all $\boldsymbol{y}_j^{[i]}$ are known functions of $\boldsymbol{\theta}$, this can be approximately

expressed as

$$\underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, p\left(\boldsymbol{z}_1^{[1]}, \ldots \mid \boldsymbol{y}_1^{[1]}, \ldots\right) p(\boldsymbol{\theta}) \tag{C.8}$$

This can be simplified as follows. First, since the noise vectors are assumed to be independent, the joint likelihood is equal to the product of the likelihoods of the individual noise vectors,

$$p\left(\boldsymbol{z}_1^{[1]}, \ldots \mid \boldsymbol{y}_1^{[1]}, \ldots\right) = \prod_{i,j} p\left(\boldsymbol{z}_j^{[i]} \mid \boldsymbol{y}_j^{[i]}\right) \tag{C.9}$$

Then, since the posterior density is always nonnegative and the natural logarithm is a monotonically increasing function, maximimizing the posterior density is equivalent to maximizing its natural logarithm, which can then be expanded into a sum:

$$\underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, p\left(\boldsymbol{z}_1^{[1]}, \ldots \mid \boldsymbol{y}_1^{[1]}, \ldots\right) p(\boldsymbol{\theta}) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left(\prod_{i,j} p\left(\boldsymbol{z}_j^{[i]} \mid \boldsymbol{y}_j^{[i]}\right)\right) p(\boldsymbol{\theta}) \tag{C.10}$$

$$= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \ln\left(\left(\prod_{i,j} p\left(\boldsymbol{z}_j^{[i]} \mid \boldsymbol{y}_j^{[i]}\right)\right) p(\boldsymbol{\theta})\right) \tag{C.11}$$

$$= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left(\sum_{i,j} \ln p\left(\boldsymbol{z}_j^{[i]} \mid \boldsymbol{y}_j^{[i]}\right) + \ln p(\boldsymbol{\theta})\right) \tag{C.12}$$

Let

$$\boldsymbol{r}_j^{[i]} = \boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\left(\boldsymbol{z}_j^{[i]} - \boldsymbol{h}\left(\boldsymbol{y}_j^{[i]}\right)\right) \tag{C.13}$$

where $\boldsymbol{L}_{\boldsymbol{\epsilon}}$ is a factor in the Cholesky decomposition $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}^{-1} = \boldsymbol{L}_{\boldsymbol{\epsilon}}\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}$, so that

$$\left(\boldsymbol{z}_j^{[i]} - \boldsymbol{h}\left(\boldsymbol{x}_j^{[i]}\right)\right)^{\top} \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}^{-1}\left(\boldsymbol{z}_j^{[i]} - \boldsymbol{h}\left(\boldsymbol{x}_j^{[i]}\right)\right) \tag{C.14}$$

$$\approx \left(\boldsymbol{z}_j^{[i]} - \boldsymbol{h}\left(\boldsymbol{y}_j^{[i]}\right)\right)^{\top} \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}^{-1}\left(\boldsymbol{z}_j^{[i]} - \boldsymbol{h}\left(\boldsymbol{y}_j^{[i]}\right)\right) \tag{C.15}$$

$$= \left(\boldsymbol{z}_j^{[i]} - \boldsymbol{h}\left(\boldsymbol{y}_j^{[i]}\right)\right)^{\top} \boldsymbol{L}_{\boldsymbol{\epsilon}}\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\left(\boldsymbol{z}_j^{[i]} - \boldsymbol{h}\left(\boldsymbol{y}_j^{[i]}\right)\right) \tag{C.16}$$

$$= \left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\left(\boldsymbol{z}_j^{[i]} - \boldsymbol{h}\left(\boldsymbol{y}_j^{[i]}\right)\right)\right)^{\top} \boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\left(\boldsymbol{z}_j^{[i]} - \boldsymbol{h}\left(\boldsymbol{y}_j^{[i]}\right)\right) \tag{C.17}$$

$$= \left(\boldsymbol{r}_j^{[i]}\right)^{\top} \boldsymbol{r}_j^{[i]} \tag{C.18}$$

Similarly, let

$$\boldsymbol{r}_{\boldsymbol{\theta}} = \boldsymbol{L}_{\boldsymbol{\theta}}^{\top}\left(\boldsymbol{\theta} - \boldsymbol{\mu}_{\boldsymbol{\theta}}\right) \tag{C.19}$$

where $\boldsymbol{L}_{\boldsymbol{\theta}}$ is a factor in the Cholesky decomposition $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1} = \boldsymbol{L}_{\boldsymbol{\theta}}\boldsymbol{L}_{\boldsymbol{\theta}}^{\top}$, so that

$$\left(\boldsymbol{\theta} - \boldsymbol{\mu}_{\boldsymbol{\theta}}\right)^{\top} \boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1}\left(\boldsymbol{\theta} - \boldsymbol{\mu}_{\boldsymbol{\theta}}\right) = \left(\boldsymbol{r}_{\boldsymbol{\theta}}\right)^{\top} \boldsymbol{r}_{\boldsymbol{\theta}} \tag{C.20}$$

Then, substituting eqs. (C.2), (C.4), (C.18), and (C.20) into eq. (C.12), the optimization problem becomes

$$\underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left(\begin{array}{l} \displaystyle\sum_{i,j} \ln\left(\left((2\pi)^{d_{\boldsymbol{z}}} \det \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\right)^{-1/2} \exp\left(-\frac{1}{2}\left(\boldsymbol{r}_j^{[i]}\right)^{\top} \boldsymbol{r}_j^{[i]}\right)\right) \\ \quad + \ln\left(\left((2\pi)^{d_{\boldsymbol{\theta}}} \det \boldsymbol{\Sigma}_{\boldsymbol{\theta}}\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\left(\boldsymbol{r}_{\boldsymbol{\theta}}\right)^{\top} \boldsymbol{r}_{\boldsymbol{\theta}}\right)\right) \end{array} \right) \tag{C.21}$$

$$= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left(\begin{array}{l} \displaystyle\sum_{i,j} \left(\ln\left((2\pi)^{d_{\boldsymbol{z}}} \det \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\right)^{-1/2} - \frac{1}{2}\left(\boldsymbol{r}_j^{[i]}\right)^{\top} \boldsymbol{r}_j^{[i]}\right) \\ \quad + \ln\left((2\pi)^{d_{\boldsymbol{\theta}}} \det \boldsymbol{\Sigma}_{\boldsymbol{\theta}}\right)^{-\frac{1}{2}} - \frac{1}{2}\left(\boldsymbol{r}_{\boldsymbol{\theta}}\right)^{\top} \boldsymbol{r}_{\boldsymbol{\theta}} \end{array} \right) \tag{C.22}$$

$$= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left(\begin{array}{l} \displaystyle\sum_{i,j} \ln\left((2\pi)^{d_{\boldsymbol{z}}} \det \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\right)^{-1/2} - \frac{1}{2}\sum_{i,j}\left(\boldsymbol{r}_j^{[i]}\right)^{\top} \boldsymbol{r}_j^{[i]} \\ \quad + \ln\left((2\pi)^{d_{\boldsymbol{\theta}}} \det \boldsymbol{\Sigma}_{\boldsymbol{\theta}}\right)^{-\frac{1}{2}} - \frac{1}{2}\left(\boldsymbol{r}_{\boldsymbol{\theta}}\right)^{\top} \boldsymbol{r}_{\boldsymbol{\theta}} \end{array} \right) \tag{C.23}$$

Subtracting the terms which do not depend on $\boldsymbol{\theta}$ yields

$$\underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left(-\frac{1}{2}\sum_{i,j}\left(\boldsymbol{r}_j^{[i]}\right)^{\top} \boldsymbol{r}_j^{[i]} - \frac{1}{2}\left(\boldsymbol{r}_{\boldsymbol{\theta}}\right)^{\top} \boldsymbol{r}_{\boldsymbol{\theta}}\right) \tag{C.24}$$

Dividing by $-1/2$ changes this into an equivalent minimization problem:

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left(\sum_{i,j} \left(\boldsymbol{r}_j^{[i]} \right)^\top \boldsymbol{r}_j^{[i]} + \left(\boldsymbol{r}_{\boldsymbol{\theta}} \right)^\top \boldsymbol{r}_{\boldsymbol{\theta}} \right) \tag{C.25}$$

This is an unconstrained, nonlinear, least-squares optimization problem. Solving it finds the MAP estimate of $\boldsymbol{\theta}$ according to the prior, measurements, and models.

Appendix D

Guessing the Initial Period for Trajectory Matching

When fitting a delay differential equation (DDE) model to time series, it can be advantageous to start with a good initial guess of the unknowns. Guessing the unknown parameter values generally relies on prior knowledge of the system, but a good guess of the unknown initial states of the system can be obtained from the initial data. Assuming the spectral element method is used to approximate the solution of the model, one approach is to find the maximum likelihood estimate of the nodes in the first period.

Using a similar derivation similar to Appendix C, an optimization problem can be written for the approximation of the first period for each time series i, using only the first period of data:

$$\underset{\boldsymbol{Y}_{\text{init}}^{[i]}}{\operatorname{argmin}} \sum_{\left\{j \mid t_j^{[i]} \in [0, T^{[i]}]\right\}} \left(\boldsymbol{r}_j^{[i]}\right)^\top \boldsymbol{r}_j^{[i]} \tag{D.1}$$

where $t_j^{[i]}$ is the time of the j^{th} sample in time series i; $T^{[i]}$ is the length of the period; $\eta_0^{[i]}, \ldots, \eta_m^{[i]}$ are the times of the spectral element nodes in the first period;

$\boldsymbol{y}^{[i]}(t) \approx \boldsymbol{x}^{[i]}(t)$ is the spectral element approximation of the true state; and

$$\boldsymbol{r}_j^{[i]} = \boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\left(\boldsymbol{z}_j^{[i]} - \boldsymbol{h}\left(\boldsymbol{y}_j^{[i]}\right)\right) \tag{D.2}$$

$$\boldsymbol{Y}_{\text{init}}^{[i]} = \left[\boldsymbol{y}^{[i]}\left(\eta_0^{[i]}\right) \quad \ldots \quad \boldsymbol{y}^{[i]}\left(\eta_m^{[i]}\right)\right] \tag{D.3}$$

Equation (D.1) provides a good guess for the first period which can then be used for the model fitting optimization problem.

Conveniently, if $\boldsymbol{h}(\boldsymbol{y}) = \boldsymbol{H}\boldsymbol{y}$, and $\boldsymbol{H}$ is invertible, this can be expressed as an ordinary least squares problem, for which an analytical solution is available. This can be shown as follows.

For the first period, $\boldsymbol{y}_j^{[i]}$ can be written as

$$\boldsymbol{y}_j^{[i]} = \boldsymbol{Y}_{\text{init}}^{[i]}\boldsymbol{\phi}_j^{[i]}, \quad \forall\, t_j^{[i]} \in \left[0, T^{[i]}\right] \tag{D.4}$$

where $\boldsymbol{\phi}_j^{[i]}$ is a column vector of the values of the spectral element trial functions at time $t_j^{[i]}$. So,

$$\boldsymbol{r}_j^{[i]} = \boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\left(\boldsymbol{z}_j^{[i]} - \boldsymbol{h}\left(\boldsymbol{Y}_{\text{init}}^{[i]}\boldsymbol{\phi}_j^{[i]}\right)\right), \quad \forall\, t_j^{[i]} \in \left[0, T^{[i]}\right] \tag{D.5}$$

$$= \boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\left(\boldsymbol{z}_j^{[i]} - \boldsymbol{H}\boldsymbol{Y}_{\text{init}}^{[i]}\boldsymbol{\phi}_j^{[i]}\right), \quad \forall\, t_j^{[i]} \in \left[0, T^{[i]}\right] \tag{D.6}$$

$$= \boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{z}_j^{[i]} - \boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{H}\boldsymbol{Y}_{\text{init}}^{[i]}\boldsymbol{\phi}_j^{[i]}, \quad \forall\, t_j^{[i]} \in \left[0, T^{[i]}\right] \tag{D.7}$$

Let v_i be the number of measurements in the initial period for time series i, then the minimization problem can be written as

$$\underset{\boldsymbol{Y}_{\text{init}}^{[i]}}{\text{argmin}} \left\| \begin{bmatrix} \left(\boldsymbol{r}_1^{[i]}\right)^{\top} \\ \vdots \\ \left(\boldsymbol{r}_{v_i}^{[i]}\right)^{\top} \end{bmatrix} \right\|^2 \tag{D.8}$$

$$= \underset{\boldsymbol{Y}_{\text{init}}^{[i]}}{\text{argmin}} \left\| \begin{bmatrix} \left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{z}_1^{[i]} - \boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{H}\boldsymbol{Y}_{\text{init}}^{[i]}\boldsymbol{\phi}_1^{[i]}\right)^{\top} \\ \vdots \\ \left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{z}_{v_i}^{[i]} - \boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{H}\boldsymbol{Y}_{\text{init}}^{[i]}\boldsymbol{\phi}_{v_i}^{[i]}\right)^{\top} \end{bmatrix} \right\|^2 \tag{D.9}$$

$$= \left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{H}\right)^{-1}\left(\begin{array}{c}\\ \underset{\left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{H}\boldsymbol{Y}_{\text{init}}^{[i]}\right)^{\top}}{\text{argmin}}\\ \\ \end{array}\left\|\begin{bmatrix}\left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{z}_{1}^{[i]}-\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{H}\boldsymbol{Y}_{\text{init}}^{[i]}\boldsymbol{\phi}_{1}^{[i]}\right)^{\top}\\ \vdots\\ \left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{z}_{v_i}^{[i]}-\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{H}\boldsymbol{Y}_{\text{init}}^{[i]}\boldsymbol{\phi}_{v_i}^{[i]}\right)^{\top}\end{bmatrix}\right\|^{2}\right)^{\top} \tag{D.10}$$

$$= \left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{H}\right)^{-1}\left(\begin{array}{c}\\ \underset{\left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{H}\boldsymbol{Y}_{\text{init}}^{[i]}\right)^{\top}}{\text{argmin}}\\ \\ \end{array}\left\|\begin{bmatrix}\left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{z}_{1}^{[i]}\right)^{\top}\\ \vdots\\ \left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{z}_{v_i}^{[i]}\right)^{\top}\end{bmatrix}-\begin{bmatrix}\left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{H}\boldsymbol{Y}_{\text{init}}^{[i]}\boldsymbol{\phi}_{1}^{[i]}\right)^{\top}\\ \vdots\\ \left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{H}\boldsymbol{Y}_{\text{init}}^{[i]}\boldsymbol{\phi}_{v_i}^{[i]}\right)^{\top}\end{bmatrix}\right\|^{2}\right)^{\top} \tag{D.11}$$

$$= \left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{H}\right)^{-1}\left(\begin{array}{c}\\ \underset{\left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{H}\boldsymbol{Y}_{\text{init}}^{[i]}\right)^{\top}}{\text{argmin}}\\ \\ \end{array}\left\|\begin{bmatrix}\left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{z}_{1}^{[i]}\right)^{\top}\\ \vdots\\ \left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{z}_{v_i}^{[i]}\right)^{\top}\end{bmatrix}-\begin{bmatrix}\left(\boldsymbol{\phi}_{1}^{[i]}\right)^{\top}\left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{H}\boldsymbol{Y}_{\text{init}}^{[i]}\right)^{\top}\\ \vdots\\ \left(\boldsymbol{\phi}_{v_i}^{[i]}\right)^{\top}\left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{H}\boldsymbol{Y}_{\text{init}}^{[i]}\right)^{\top}\end{bmatrix}\right\|^{2}\right)^{\top} \tag{D.12}$$

$$= \left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{H}\right)^{-1}\left(\begin{array}{c}\\ \underset{\left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{H}\boldsymbol{Y}_{\text{init}}^{[i]}\right)^{\top}}{\text{argmin}}\\ \\ \end{array}\left\|\begin{bmatrix}\left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{z}_{1}^{[i]}\right)^{\top}\\ \vdots\\ \left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{z}_{v_i}^{[i]}\right)^{\top}\end{bmatrix}-\begin{bmatrix}\left(\boldsymbol{\phi}_{1}^{[i]}\right)^{\top}\\ \vdots\\ \left(\boldsymbol{\phi}_{v_i}^{[i]}\right)^{\top}\end{bmatrix}\left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{H}\boldsymbol{Y}_{\text{init}}^{[i]}\right)^{\top}\right\|^{2}\right)^{\top} \tag{D.13}$$

where $\|\cdot\|$ is the Frobenius norm. The minimization problem is an ordinary least squares problem, so, using the well-known solution,

$$\left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{H}\right)^{-1}\left(\begin{array}{c}\\ \underset{\left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{H}\boldsymbol{Y}_{\text{init}}^{[i]}\right)^{\top}}{\text{argmin}}\\ \\ \end{array}\left\|\begin{bmatrix}\left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{z}_{1}^{[i]}\right)^{\top}\\ \vdots\\ \left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{z}_{v_i}^{[i]}\right)^{\top}\end{bmatrix}-\begin{bmatrix}\left(\boldsymbol{\phi}_{1}^{[i]}\right)^{\top}\\ \vdots\\ \left(\boldsymbol{\phi}_{v_i}^{[i]}\right)^{\top}\end{bmatrix}\left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{H}\boldsymbol{Y}_{\text{init}}^{[i]}\right)^{\top}\right\|^{2}\right)^{\top} \tag{D.14}$$

$$= \left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{H}\right)^{-1}\left(\left(\begin{bmatrix}\left(\boldsymbol{\phi}_{1}^{[i]}\right)^{\top}\\ \vdots\\ \left(\boldsymbol{\phi}_{v_i}^{[i]}\right)^{\top}\end{bmatrix}^{\top}\begin{bmatrix}\left(\boldsymbol{\phi}_{1}^{[i]}\right)^{\top}\\ \vdots\\ \left(\boldsymbol{\phi}_{v_i}^{[i]}\right)^{\top}\end{bmatrix}\right)^{-1}\begin{bmatrix}\left(\boldsymbol{\phi}_{1}^{[i]}\right)^{\top}\\ \vdots\\ \left(\boldsymbol{\phi}_{v_i}^{[i]}\right)^{\top}\end{bmatrix}^{\top}\begin{bmatrix}\left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{z}_{1}^{[i]}\right)^{\top}\\ \vdots\\ \left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{z}_{v_i}^{[i]}\right)^{\top}\end{bmatrix}\right)^{\top} \tag{D.15}$$

$$= \left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{H}\right)^{-1} \begin{bmatrix} \left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{z}_1^{[i]}\right)^{\top} \\ \vdots \\ \left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{z}_{v_i}^{[i]}\right)^{\top} \end{bmatrix}^{\top} \begin{bmatrix} \left(\boldsymbol{\phi}_1^{[i]}\right)^{\top} \\ \vdots \\ \left(\boldsymbol{\phi}_{v_i}^{[i]}\right)^{\top} \end{bmatrix} \left(\left(\begin{bmatrix} \left(\boldsymbol{\phi}_1^{[i]}\right)^{\top} \\ \vdots \\ \left(\boldsymbol{\phi}_{v_i}^{[i]}\right)^{\top} \end{bmatrix}^{\top} \begin{bmatrix} \left(\boldsymbol{\phi}_1^{[i]}\right)^{\top} \\ \vdots \\ \left(\boldsymbol{\phi}_{v_i}^{[i]}\right)^{\top} \end{bmatrix} \right)^{-1} \right)^{\top} \tag{D.16}$$

$$= \boldsymbol{H}^{-1}\left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\right)^{-1} \begin{bmatrix} \boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{z}_1^{[i]} & \cdots & \boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\boldsymbol{z}_{v_i}^{[i]} \end{bmatrix} \begin{bmatrix} \left(\boldsymbol{\phi}_1^{[i]}\right)^{\top} \\ \vdots \\ \left(\boldsymbol{\phi}_{v_i}^{[i]}\right)^{\top} \end{bmatrix} \left(\left(\begin{bmatrix} \left(\boldsymbol{\phi}_1^{[i]}\right)^{\top} \\ \vdots \\ \left(\boldsymbol{\phi}_{v_i}^{[i]}\right)^{\top} \end{bmatrix}^{\top} \begin{bmatrix} \left(\boldsymbol{\phi}_1^{[i]}\right)^{\top} \\ \vdots \\ \left(\boldsymbol{\phi}_{v_i}^{[i]}\right)^{\top} \end{bmatrix} \right)^{\top} \right)^{-1} \tag{D.17}$$

$$= \boldsymbol{H}^{-1}\left(\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top}\right)^{-1}\boldsymbol{L}_{\boldsymbol{\epsilon}}^{\top} \begin{bmatrix} \boldsymbol{z}_1^{[i]} & \cdots & \boldsymbol{z}_{v_i}^{[i]} \end{bmatrix} \begin{bmatrix} \left(\boldsymbol{\phi}_1^{[i]}\right)^{\top} \\ \vdots \\ \left(\boldsymbol{\phi}_{v_i}^{[i]}\right)^{\top} \end{bmatrix} \left(\begin{bmatrix} \left(\boldsymbol{\phi}_1^{[i]}\right)^{\top} \\ \vdots \\ \left(\boldsymbol{\phi}_{v_i}^{[i]}\right)^{\top} \end{bmatrix}^{\top} \begin{bmatrix} \left(\boldsymbol{\phi}_1^{[i]}\right)^{\top} \\ \vdots \\ \left(\boldsymbol{\phi}_{v_i}^{[i]}\right)^{\top} \end{bmatrix} \right)^{-1} \tag{D.18}$$

$$= \boldsymbol{H}^{-1} \begin{bmatrix} \boldsymbol{z}_1^{[i]} & \cdots & \boldsymbol{z}_{v_i}^{[i]} \end{bmatrix} \begin{bmatrix} \left(\boldsymbol{\phi}_1^{[i]}\right)^{\top} \\ \vdots \\ \left(\boldsymbol{\phi}_{v_i}^{[i]}\right)^{\top} \end{bmatrix} \left(\begin{bmatrix} \boldsymbol{\phi}_1^{[i]} & \cdots & \boldsymbol{\phi}_{v_i}^{[i]} \end{bmatrix} \begin{bmatrix} \left(\boldsymbol{\phi}_1^{[i]}\right)^{\top} \\ \vdots \\ \left(\boldsymbol{\phi}_{v_i}^{[i]}\right)^{\top} \end{bmatrix} \right)^{-1} \tag{D.19}$$

Appendix E

Adapting the Spectral Element Method

Consider a system with piecewise dynamics,

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}(t,\boldsymbol{p})\boldsymbol{x}(t) + \boldsymbol{B}(t,\boldsymbol{p})\boldsymbol{x}(t-\tau) + \boldsymbol{c}(t,\boldsymbol{p}) \qquad \text{if } 0 \le t - t_{\mathrm{b}}(t) < t_{\mathrm{d}} \quad \text{(E.1)}$$

$$\boldsymbol{x}(t) = \boldsymbol{D}(t - t_{\mathrm{s}}(t), \boldsymbol{p}) \left(\boldsymbol{x}(t_{\mathrm{s}}(t)) - \boldsymbol{e}(t_{\mathrm{s}}(t), \boldsymbol{p})\right) + \boldsymbol{e}(t,\boldsymbol{p}) \quad \text{if } t_{\mathrm{d}} \le t - t_{\mathrm{b}}(t) < \tau \quad \text{(E.2)}$$

where $\boldsymbol{A}(t,\boldsymbol{p})$, $\boldsymbol{B}(t,\boldsymbol{p})$, $\boldsymbol{c}(t,\boldsymbol{p})$, and $\boldsymbol{e}(t,\boldsymbol{p})$ are functions with period T; T is an integer multiple of τ; and

$$t_{\mathrm{b}}(t) = \tau \lfloor (t - t_{\mathrm{i}})/\tau \rfloor + t_{\mathrm{i}} \quad \text{(E.3)}$$

$$t_{\mathrm{s}}(t) = t_{\mathrm{b}}(t) + t_{\mathrm{d}} \quad \text{(E.4)}$$

Time is split into segments of duration τ, starting at an initial time t_{i}. Within each segment of duration τ, the system follows eq. (E.1) for the initial interval of duration t_{d}, and then it switches to eq. (E.2) for the remaining interval of duration $\tau - t_{\mathrm{d}}$. Note that the milling model used in Chapter 4 is of this form—eq. (4.4) corresponds to eq. (E.1), and eq. (4.36) corresponds to eq. (E.2)—but the spectral element method described here can also be used for any other system with dynamics of this form.

The solution for a system with dynamics described by eqs. (E.1) and (E.2) can be approximated using an extended version of the spectral element method presented

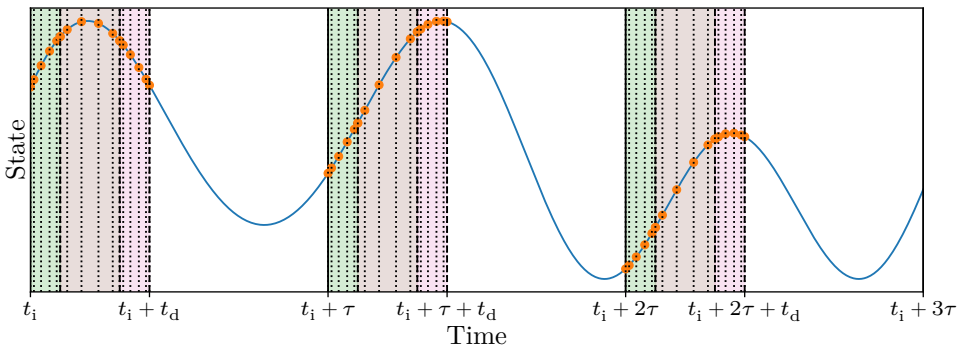


FIGURE E.1: Illustration of the spectral element segments, elements, and nodes for a system with piecewise dynamics described by eqs. (E.1) and (E.2). The state of the system is indicated with a curved blue line. The divisions between segments are indicated with solid vertical lines. The end of each element is indicated with a vertical dashed line. The location of each node is indicated with a vertical line (solid, dashed, or dotted), and the corresponding value of the state is indicated with an orange circular marker. The intervals of time following eq. (E.1) have a shaded background, while the intervals of time following eq. (E.2) have a white background.

in [10]. The approach and derivation used here is similar, but is extended to handle nonzero $\boldsymbol{c}(t, \boldsymbol{p})$, piecewise dynamics, and coefficients whose period is an integer multiple of τ.

The approximate solution is represented by the value of the solution at discrete nodes, as illustrated by the circular markers in fig. E.1. The initial interval of duration t_d in each segment is split into n_e elements with indices $\{0, \ldots, n_\mathrm{e} - 1\}$. Within each element, the solution is approximated by a polynomial of order n_o which interpolates between $n_\mathrm{o} + 1$ nodes with indices $\{0, \ldots, n_\mathrm{o}\}$. Let $t_{i,j,k}$ be the time corresponding to node k in element j in segment i. The times of the nodes within segments are consistent, i.e. $t_{i,j,k} = t_{i-1,j,k} + \tau$, and consecutive elements within a segment are contiguous, i.e. $t_{i,j,0} = t_{i,j-1,n_\mathrm{o}}$ for $j > 0$. For element j in segment i, the solution is approximated by

$$\boldsymbol{x}(t) \approx \sum_{k=0}^{n_\mathrm{o}} \boldsymbol{a}_{i,j,k} \phi_k(\nu_{i,j}(t)), \qquad t \in \left[t_{i,j,0}, t_{i,j,n_\mathrm{o}}\right] \tag{E.5}$$

where $\boldsymbol{a}_{i,j,k}$ is the solution at time $t_{i,j,k}$; $\nu_{i,j}(t) = (t - t_{i,j,0})/(t_{i,j,n_\mathrm{o}} - t_{i,j,0})$ is a

function shifting and scaling the time t to a coordinate in $[0,1]$ within the element; and $\phi_k : [0,1] \to \mathbb{R}$ are Lagrange trial functions of order n_o. The nodes are chosen to be the Legendre–Gauss–Lobatto (LGL) points of the specified order within the element, and the trial functions are chosen to be Lagrange polynomials, such that

$$\phi_k(\nu_{i,j}(t_{i,j,m})) = \begin{cases} 1 & k = m \\ 0 & k \neq m \end{cases} \tag{E.6}$$

For times following the initial interval of duration t_d in each segment i, the dynamics follow eq. (E.2), so the solution in this interval for segment i is approximated by

$$\begin{aligned} \boldsymbol{x}(t) \approx \boldsymbol{D}(t - t_{i,n_\mathrm{e}-1,n_\mathrm{o}}, \boldsymbol{p}) \left(\boldsymbol{a}_{i,n_\mathrm{e}-1,n_\mathrm{o}} - \boldsymbol{e}(t_{i,n_\mathrm{e}-1,n_\mathrm{o}}, \boldsymbol{p}) \right) + \boldsymbol{e}(t, \boldsymbol{p}), \\ t \in \left[t_{i,n_\mathrm{e}-1,n_\mathrm{o}}, t_{i+1,0,0} \right] \end{aligned} \tag{E.7}$$

Applying this to the full interval of duration $\tau - t_\mathrm{d}$ between the end of the last element in previous segment and the start of the current segment yields

$$\boldsymbol{a}_{i,0,0} = \boldsymbol{D}(\tau - t_\mathrm{d}, \boldsymbol{p}) \left(\boldsymbol{a}_{i-1,n_\mathrm{e}-1,n_\mathrm{o}} - \boldsymbol{e}(t_{i-1,n_\mathrm{e}-1,n_\mathrm{o}}, \boldsymbol{p}) \right) + \boldsymbol{e}(t_{i,0,0}, \boldsymbol{p}) \tag{E.8}$$

$$= \boldsymbol{D}(\tau - t_\mathrm{d}, \boldsymbol{p}) \boldsymbol{a}_{i-1,n_\mathrm{e}-1,n_\mathrm{o}} + \tilde{\boldsymbol{d}}_i \tag{E.9}$$

where

$$\tilde{\boldsymbol{d}}_i = \boldsymbol{e}(t_{i,0,0}, \boldsymbol{p}) - \boldsymbol{D}(\tau - t_\mathrm{d}, \boldsymbol{p}) \boldsymbol{e}(t_{i-1,n_\mathrm{e}-1,n_\mathrm{o}}, \boldsymbol{p}) \tag{E.10}$$

Approximate expressions for the time derivative and delayed state within each element can be derived from eq. (E.5). Note that the value of the derivative $\nu'_{i,j}(t)$ is independent of i and t; this derivative will be denoted ν'_j. Then, for element j in segment i,

$$\dot{\boldsymbol{x}}(t) \approx \sum_{k=0}^{n_\mathrm{o}} \boldsymbol{a}_{i,j,k} \nu'_j \phi'_k(\nu_{i,j}(t)), \qquad t \in \left[t_{i,j,0}, t_{i,j,n_\mathrm{o}} \right] \tag{E.11}$$

$$\boldsymbol{x}(t - \tau) \approx \sum_{k=0}^{n_\mathrm{o}} \boldsymbol{a}_{i-1,j,k} \phi_k(\nu_{i,j}(t)), \qquad t \in \left[t_{i,j,0}, t_{i,j,n_\mathrm{o}} \right] \tag{E.12}$$

Substituting into eq. (E.1) and rearranging yields, for $t \in \left[t_{i,j,0}, t_{i,j,n_\mathrm{o}}\right]$,

$$\begin{aligned}
&\left(\sum_{k=0}^{n_\mathrm{o}} \boldsymbol{a}_{i,j,k} \nu_j' \phi_k'(\nu_{i,j}(t))\right) - \boldsymbol{A}(t,\boldsymbol{p}) \left(\sum_{k=0}^{n_\mathrm{o}} \boldsymbol{a}_{i,j,k} \phi_k(\nu_{i,j}(t))\right) \\
&\approx \boldsymbol{B}(t,\boldsymbol{p}) \left(\sum_{k=0}^{n_\mathrm{o}} \boldsymbol{a}_{i-1,j,k} \phi_k(\nu_{i,j}(t))\right) + \boldsymbol{c}(t,\boldsymbol{p})
\end{aligned} \tag{E.13}$$

Simplifying, this becomes, for $t \in \left[t_{i,j,0}, t_{i,j,n_\mathrm{o}}\right]$,

$$\begin{aligned}
&\sum_{k=0}^{n_\mathrm{o}} \left(\boldsymbol{I} \nu_j' \phi_k'(\nu_{i,j}(t)) - \boldsymbol{A}(t,\boldsymbol{p}) \phi_k(\nu_{i,j}(t))\right) \boldsymbol{a}_{i,j,k} \\
&\approx \left(\sum_{k=0}^{n_\mathrm{o}} \boldsymbol{B}(t,\boldsymbol{p}) \phi_k(\nu_{i,j}(t)) \boldsymbol{a}_{i-1,j,k}\right) + \boldsymbol{c}(t,\boldsymbol{p})
\end{aligned} \tag{E.14}$$

The method of weighted residuals can be applied to obtain multiple equations which enforce approximate equality of eq. (E.14). As in [10], the first n_o Legendre polynomials are used as the test functions. They are denoted ψ_ℓ and have indices $\ell \in \{0, \ldots, n_\mathrm{o} - 1\}$. For each element j in segment i:

$$\begin{aligned}
&\int_0^1 \left(\sum_{k=0}^{n_\mathrm{o}} \left(\boldsymbol{I} \nu_j' \phi_k'(\sigma) - \boldsymbol{A}\left(\nu_{i,j}^{-1}(\sigma), \boldsymbol{p}\right) \phi_k(\sigma)\right) \boldsymbol{a}_{i,j,k}\right) \psi_\ell(\sigma) \,\mathrm{d}\sigma \\
&= \int_0^1 \left(\left(\sum_{k=0}^{n_\mathrm{o}} \boldsymbol{B}\left(\nu_{i,j}^{-1}(\sigma), \boldsymbol{p}\right) \phi_k(\sigma) \boldsymbol{a}_{i-1,j,k}\right) + \boldsymbol{c}\left(\nu_{i,j}^{-1}(\sigma), \boldsymbol{p}\right)\right) \psi_\ell(\sigma) \,\mathrm{d}\sigma
\end{aligned} \tag{E.15}$$

where $\nu_{i,j}^{-1}(\sigma) = t_{i,j,0} + \sigma \cdot (t_{i,j,n_\mathrm{o}} - t_{i,j,0})$ denotes the inverse function of $\nu_{i,j}$. Rearranging, this becomes

$$\begin{aligned}
&\sum_{k=0}^{n_\mathrm{o}} \left(\int_0^1 \left(\boldsymbol{I} \nu_j' \phi_k'(\sigma) - \boldsymbol{A}\left(\nu_{i,j}^{-1}(\sigma), \boldsymbol{p}\right) \phi_k(\sigma)\right) \psi_\ell(\sigma) \,\mathrm{d}\sigma\right) \boldsymbol{a}_{i,j,k} \\
&= \sum_{k=0}^{n_\mathrm{o}} \left(\int_0^1 \boldsymbol{B}\left(\nu_{i,j}^{-1}(\sigma), \boldsymbol{p}\right) \phi_k(\sigma) \psi_\ell(\sigma) \,\mathrm{d}\sigma\right) \boldsymbol{a}_{i-1,j,k} + \int_0^1 \boldsymbol{c}\left(\nu_{i,j}^{-1}(\sigma), \boldsymbol{p}\right) \psi_\ell(\sigma) \,\mathrm{d}\sigma
\end{aligned} \tag{E.16}$$

which can be written as

$$\sum_{k=0}^{n_\mathrm{o}} \boldsymbol{N}_{i,j,k,\ell} \boldsymbol{a}_{i,j,k} = \sum_{k=0}^{n_\mathrm{o}} \boldsymbol{P}_{i,j,k,\ell} \boldsymbol{a}_{i-1,j,k} + \boldsymbol{d}_{i,j,\ell} \tag{E.17}$$

where

$$\boldsymbol{N}_{i,j,k,\ell} = \int_0^1 \left(\boldsymbol{I} \nu_j' \phi_k'(\sigma) - \boldsymbol{A}(\nu_{i,j}^{-1}(\sigma), \boldsymbol{p}) \phi_k(\sigma) \right) \psi_\ell(\sigma) \, \mathrm{d}\sigma \tag{E.18}$$

$$\boldsymbol{P}_{i,j,k,\ell} = \int_0^1 \boldsymbol{B}(\nu_{i,j}^{-1}(\sigma), \boldsymbol{p}) \phi_k(\sigma) \psi_\ell(\sigma) \, \mathrm{d}\sigma \tag{E.19}$$

$$\boldsymbol{d}_{i,j,\ell} = \int_0^1 \boldsymbol{c}(\nu_{i,j}^{-1}(\sigma), \boldsymbol{p}) \psi_\ell(\sigma) \, \mathrm{d}\sigma \tag{E.20}$$

The integrals can be approximated by LGL quadrature using LGL weights w_m, which is convenient because the nodes were chosen to be LGL points:

$$\boldsymbol{N}_{i,j,k,\ell} \approx \sum_{m=1}^{n_\mathrm{o}} \left(\boldsymbol{I} \nu_j' \phi_k'(\sigma_m) - \boldsymbol{A}\left(t_{i,j,m}, \boldsymbol{p}\right) \phi_k(\sigma_m) \right) \psi_\ell(\sigma_m) w_m \tag{E.21}$$

$$\boldsymbol{P}_{i,j,k,\ell} \approx \sum_{m=1}^{n_\mathrm{o}} \boldsymbol{B}\left(t_{i,j,m}, \boldsymbol{p}\right) \phi_k(\sigma_m) \psi_\ell(\sigma_m) w_m \tag{E.22}$$

$$\boldsymbol{d}_{i,j,\ell} \approx \sum_{m=1}^{n_\mathrm{o}} \boldsymbol{c}\left(t_{i,j,m}, \boldsymbol{p}\right) \psi_\ell(\sigma_m) w_m \tag{E.23}$$

where $\sigma_m = \nu_{i,j}(t_{i,j,m})$. Substituting eq. (E.6), this simplifies to

$$\boldsymbol{N}_{i,j,k,\ell} \approx \left(\sum_{m=1}^{n_\mathrm{o}} \nu_j' \phi_k'(\sigma_m) \psi_\ell(\sigma_m) w_m \right) \boldsymbol{I} - \boldsymbol{A}\left(t_{i,j,k}, \boldsymbol{p}\right) \phi_k(\sigma_k) \psi_\ell(\sigma_k) w_k \tag{E.24}$$

$$\boldsymbol{P}_{i,j,k,\ell} \approx \boldsymbol{B}\left(t_{i,j,k}, \boldsymbol{p}\right) \phi_k(\sigma_k) \psi_\ell(\sigma_k) w_k \tag{E.25}$$

$$\boldsymbol{d}_{i,j,\ell} \approx \sum_{m=1}^{n_\mathrm{o}} \boldsymbol{c}(t_{i,j,m}, \boldsymbol{p}) \psi_\ell(\sigma_m) w_m \tag{E.26}$$

Combining eq. (E.8) with eq. (E.17) for all the elements and trial functions yields the following relationship between the solution in segment i and the solution in

segment $i-1$:

$$\bar{\boldsymbol{N}}_i \bar{\boldsymbol{a}}_i = \bar{\boldsymbol{P}}_i \bar{\boldsymbol{a}}_{i-1} + \bar{\boldsymbol{d}}_i \tag{E.27}$$

For example, for $n_\mathrm{e} = 2$,

$$\bar{\boldsymbol{N}}_i = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{N}_{i,0,0,0} & \boldsymbol{N}_{i,0,1,0} & \cdots & \boldsymbol{N}_{i,0,n_\mathrm{o},0} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{N}_{i,0,0,n_\mathrm{o}-1} & \boldsymbol{N}_{i,0,1,n_\mathrm{o}-1} & \cdots & \boldsymbol{N}_{i,0,n_\mathrm{o},n_\mathrm{o}-1} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{N}_{i,1,0,0} & \boldsymbol{N}_{i,1,1,0} & \cdots & \boldsymbol{N}_{i,1,n_\mathrm{o},0} \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \vdots & \ddots & \vdots & \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{N}_{i,1,0,n_\mathrm{o}-1} & \boldsymbol{N}_{i,1,1,n_\mathrm{o}-1} & \cdots & \boldsymbol{N}_{i,1,n_\mathrm{o},n_\mathrm{o}-1} \end{bmatrix} \tag{E.28}$$

$$\bar{\boldsymbol{P}}_i = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{D}(\tau - t_\mathrm{d}, \boldsymbol{p}) \\ \boldsymbol{P}_{i,0,0,0} & \boldsymbol{P}_{i,0,1,0} & \cdots & \boldsymbol{P}_{i,0,n_\mathrm{o},0} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{P}_{i,0,0,n_\mathrm{o}-1} & \boldsymbol{P}_{i,0,1,n_\mathrm{o}-1} & \cdots & \boldsymbol{P}_{i,0,n_\mathrm{o},n_\mathrm{o}-1} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{P}_{i,1,0,0} & \boldsymbol{P}_{i,1,1,0} & \cdots & \boldsymbol{P}_{i,1,n_\mathrm{o},0} \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \vdots & \ddots & \vdots & \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{P}_{i,1,0,n_\mathrm{o}-1} & \boldsymbol{P}_{i,1,1,n_\mathrm{o}-1} & \cdots & \boldsymbol{P}_{i,1,n_\mathrm{o},n_\mathrm{o}-1} \end{bmatrix} \tag{E.29}$$

$$\bar{\boldsymbol{a}}_i = \begin{bmatrix} \boldsymbol{a}_{i,0,0} \\ \boldsymbol{a}_{i,0,1} \\ \vdots \\ \boldsymbol{a}_{i,0,n_\mathrm{o}} = \boldsymbol{a}_{i,1,0} \\ \boldsymbol{a}_{i,1,1} \\ \vdots \\ \boldsymbol{a}_{i,1,n_\mathrm{o}} \end{bmatrix}, \quad \bar{\boldsymbol{a}}_{i-1} = \begin{bmatrix} \boldsymbol{a}_{i-1,0,0} \\ \boldsymbol{a}_{i-1,0,1} \\ \vdots \\ \boldsymbol{a}_{i-1,0,n_\mathrm{o}} = \boldsymbol{a}_{i-1,1,0} \\ \boldsymbol{a}_{i-1,1,1} \\ \vdots \\ \boldsymbol{a}_{i-1,1,n_\mathrm{o}} \end{bmatrix}, \quad \bar{\boldsymbol{d}}_i = \begin{bmatrix} \tilde{\boldsymbol{d}}_i \\ \boldsymbol{d}_{i,0,0} \\ \vdots \\ \boldsymbol{d}_{i,0,n_\mathrm{o}-1} \\ \boldsymbol{d}_{i,1,0} \\ \vdots \\ \boldsymbol{d}_{i,1,n_\mathrm{o}-1} \end{bmatrix} \tag{E.30}$$

Equation (E.27) can be solved for $\bar{a}_i$ to obtain

$$\bar{\boldsymbol{a}}_i = \bar{\boldsymbol{Q}}_i \bar{\boldsymbol{a}}_{i-1} + \bar{\boldsymbol{r}}_i \qquad \text{where} \qquad \bar{\boldsymbol{Q}}_i = \bar{\boldsymbol{N}}_i^{-1} \bar{\boldsymbol{P}}_i, \quad \bar{\boldsymbol{r}}_i = \bar{\boldsymbol{N}}_i^{-1} \bar{\boldsymbol{d}}_i \tag{E.31}$$

Observe that $\bar{\boldsymbol{Q}}_i = \bar{\boldsymbol{Q}}_{i+T/\tau}$ and $\bar{\boldsymbol{r}}_i = \bar{\boldsymbol{r}}_{i+T/\tau}$ due to the periodicity of $\boldsymbol{A}(t, \boldsymbol{p})$, $\boldsymbol{B}(t, \boldsymbol{p})$, $\boldsymbol{c}(t, \boldsymbol{p})$, and $\boldsymbol{e}(t, \boldsymbol{p})$. As a result, $\bar{\boldsymbol{Q}}_1, \ldots, \bar{\boldsymbol{Q}}_{T/\tau}$ and $\bar{\boldsymbol{r}}_1, \ldots, \bar{\boldsymbol{r}}_{T/\tau}$ are sufficient to approximately describe the dynamics of the system. For example, given $\bar{\boldsymbol{a}}_0$,

they can be used to compute $\bar{\boldsymbol{a}}_1, \bar{\boldsymbol{a}}_2, ...$, which can be interpolated using eqs. (E.5) and (E.7) to find the approximate solution at an arbitrary future time.

For computation of the characteristic multipliers (CMs), the equations described by eq. (E.31) for all $i = 1, \dots, T/\tau$ can be combined into a single equation describing the mapping over an entire period T,

$$\bar{\boldsymbol{a}}_{(i+1)\cdot(T/\tau)} = \breve{\boldsymbol{Q}}_{1,\dots,T/\tau} \bar{\boldsymbol{a}}_{i\cdot(T/\tau)} + \breve{\boldsymbol{r}}_{1,\dots,T/\tau} \tag{E.32}$$

where $\breve{\boldsymbol{Q}}_{1,\dots,T/\tau} = \bar{\boldsymbol{Q}}_{T/\tau} \cdots \bar{\boldsymbol{Q}}_1$ is a constant matrix product and $\breve{\boldsymbol{r}}_{1,\dots,T/\tau}$ is a constant vector. The equilibrium point is given by

$$\bar{\boldsymbol{a}}^* = \left(\boldsymbol{I} - \breve{\boldsymbol{Q}}_{1,\dots,T/\tau} \right)^{-1} \breve{\boldsymbol{r}}_{1,\dots,T/\tau} \tag{E.33}$$

which can be substituted to obtain

$$\bar{\boldsymbol{a}}_{(i+1)\cdot(T/\tau)} - \bar{\boldsymbol{a}}^* = \breve{\boldsymbol{Q}}_{1,\dots,T/\tau} \left(\bar{\boldsymbol{a}}_{i\cdot(T/\tau)} - \bar{\boldsymbol{a}}^* \right) \tag{E.34}$$

So, the system converges, i.e. $\lim_{i\to\infty} \bar{\boldsymbol{a}}_{i\cdot(T/\tau)} = \bar{\boldsymbol{a}}^*$, if the all eigenvalues of $\breve{\boldsymbol{Q}}_{1,\dots,T/\tau}$ have magnitude less than 1, and diverges otherwise. In other words, the eigenvalues of $\breve{\boldsymbol{Q}}_{1,\dots,T/\tau}$ are the CMs of the spectral element approximation. They approximately describe the stability of the system being approximated.

Bibliography

[1] L. Meirovitch. *Methods of Analytical Dynamics*. Dover Publications, Inc., 2003.

[2] S. C. Sinha, C. C. Chou, and H. H. Denman. "Stability analysis of systems with periodic coefficients: An approximate approach". *Journal of Sound and Vibration* 64.4 (1979), pp. 515–527. DOI: 10.1016/0022-460X(79)90801-0.

[3] S. Sinha and D.-H. Wu. "An efficient computational scheme for the analysis of periodic systems". *Journal of Sound and Vibration* 151.1 (1991), pp. 91–117. DOI: 10.1016/0022-460X(91)90654-3.

[4] B. Balachandran and M. X. Zhao. "A Mechanics Based Model for Study of Dynamics of Milling Operations". *Meccanica* 35 (Mar. 2000), pp. 89–109. DOI: 10.1023/A:1004887301926.

[5] T. Insperger and G. Stépán. "Semi-discretization method for delayed systems". *International Journal for Numerical Methods in Engineering* 55.5 (Oct. 2002), pp. 503–518. DOI: 10.1002/nme.505.

[6] T. Insperger and G. Stépán. "Updated semi-discretization method for periodic delay-differential equations with discrete delay". *International Journal for Numerical Methods in Engineering* 61.1 (Sept. 2004), pp. 117–141. DOI: 10.1002/nme.1061.

[7] P. V. Bayly et al. "Stability of Interrupted Cutting by Temporal Finite Element Analysis". *Journal of Manufacturing Science and Engineering* 125.2 (2003), pp. 220–225. DOI: 10.1115/1.1556860.

[8] B. P. Mann and B. R. Patel. "Stability of Delay Equations Written as State Space Models". *Journal of Vibration and Control* 16.7–8 (2010), pp. 1067–1085. DOI: 10.1177/1077546309341111.

[9] D. J. Tweten et al. "On the comparison of semi-analytical methods for the stability analysis of delay differential equations". *Journal of Sound and Vibration* 331.17 (2012), pp. 4057–4071. DOI: 10.1016/j.jsv.2012.04.009.

[10] F. A. Khasawneh and B. P. Mann. "A spectral element approach for the stability of delay systems". *International Journal for Numerical Methods in Engineering* 87.6 (Aug. 2011), pp. 566–592. DOI: 10.1002/nme.3122.

[11] B. P. Mann and K. A. Young. "An Empirical Approach for Delayed Oscillator Stability and Parametric Identification". *Proceedings of the Royal Society A* 462.2071 (2006), pp. 2145–2160. DOI: 10.1098/rspa.2006.1677.

[12] A. Petrie and X. Zhao. "Estimating eigenvalues of dynamical systems from time series with applications to predicting cardiac alternans". *Proceedings of the Royal Society A* 468.2147 (Nov. 2012), pp. 3649–3666. DOI: 10.1098/rspa.2012.0098.

[13] Y. Hurmuzlu and C. Basdogan. "On the Measurement of Dynamic Stability of Human Locomotion". *Journal of Biomechanical Engineering* 116.1 (Feb. 1994), pp. 30–36. DOI: 10.1115/1.2895701.

[14] D. B. Marghitu and P. Nalluri. "Nonlinear Dynamic Stability of Normal and Arthritic Greyhounds". *Nonlinear Dynamics* 12 (1997), pp. 237–250. DOI: 10.1023/A:1008257328720.

[15] S. T. Trickey, L. N. Virgin, and E. H. Dowell. "The stability of limit–cycle oscillations in a nonlinear aeroelastic system". *Proceedings of the Royal Society of London A* 458.2025 (Sept. 2002), pp. 2203–2226. DOI: 10.1098/rspa.2002.0965.

[16] N. Choi, H. Cho, and B. Lee. "Development of Floquet Multiplier Estimator to Determine Nonlinear Oscillatory Behavior in Power System Data Measurement". *Energies* 12.10 (2019), p. 1824. DOI: 10.3390/en12101824.

[17] P. V. Bayly and L. N. Virgin. "An empirical study of the stability of periodic motion in the forced spring-pendulum". *Proceedings of the Royal Society of London A* 443.1918 (1993), pp. 391–408. DOI: 10.1098/rspa.1993.0152.

[18] K. D. Murphy et al. "Measuring the Stability of Periodic Attractors Using Perturbation-Induced Transients: Applications to Two Non-Linear Oscillators". *Journal of Sound and Vibration* 172.1 (Apr. 1994), pp. 85–102. DOI: 10.1006/jsvi.1994.1160.

[19] J. A. Little, J. D. Turner, and B. P. Mann. "Improving empirical characteristic multiplier estimation through a change of basis". *Journal of Sound and Vibration* 488 (2020). DOI: 10.1016/j.jsv.2020.115613.

[20] T. Insperger et al. "Stability of up-milling and down-milling, part 1: alternative analytical methods". *International Journal of Machine Tools and Manufacture* 43.1 (2003), pp. 25–34. DOI: 10.1016/S0890-6955(02)00159-1.

[21] B. P. Mann et al. “Chatter vibration and surface location error prediction for helical end mills”. *International Journal of Machine Tools and Manufacture* 48.3 (2008), pp. 350–361. DOI: 10.1016/j.ijmachtools.2007.10.003.

[22] D. Montgomery and Y. Altintas. “Mechanism of Cutting Force and Surface Generation in Dynamic Milling”. *Journal of Engineering for Industry* 113.2 (1991), pp. 160–168. DOI: 10.1115/1.2899673.

[23] W. T. Corpus and W. J. Endres. “A high order solution for the added stability lobes in intermittent machining”. *Proceedings of the ASME Symposium on Machining Processes*. 2000, pp. 871–878.

[24] M. A. Davies et al. “Stability Prediction for Low Radial Immersion Milling”. *Journal of Manufacturing Science and Engineering* 124.2 (2002), pp. 217–225. DOI: 10.1115/1.1455030.

[25] X.-H. Long, B. Balachandran, and B. P. Mann. “Dynamics of milling processes with variable time delays”. *Nonlinear Dynamics* 47.4 (2007), pp. 49–63. DOI: 10.1007/s11071-006-9058-4.

[26] T. L. Schmitz and B. P. Mann. “Closed-form solutions for surface location error in milling”. *International Journal of Machine Tools and Manufacture* 46.12 (2006), pp. 1369–1377. DOI: 10.1016/j.ijmachtools.2005.10.007.

[27] B. P. Mann et al. “Limit cycles, bifurcations, and accuracy of the milling process”. *Journal of Sound and Vibration* 277.1 (2004), pp. 31–48. DOI: 10.1016/j.jsv.2003.08.040.

[28] M. J. Bünner et al. “Tool to recover scalar time-delay systems from experimental time series”. *Physical Review E* 54 (4 Oct. 1996), R3082–R3085. DOI: 10.1103/PhysRevE.54.R3082.

[29] M. J. Bünner et al. “Recovery of the time-evolution equation of time-delay systems from time series”. *Physical Review E* 56 (5 Nov. 1997), pp. 5083–5089. DOI: 10.1103/PhysRevE.56.5083.

[30] S. P. Ellner et al. “Inferring mechanism from time-series data: Delay-differential equations”. *Physica D: Nonlinear Phenomena* 110.3 (1997), pp. 182–194. DOI: 10.1016/S0167-2789(97)00123-1.

[31] H. Voss and J. Kurths. “Reconstruction of non-linear time delay models from data by the use of optimal transformations”. *Physics Letters A* 234.5 (1997), pp. 336–344. DOI: 10.1016/S0375-9601(97)00598-7.

[32] M. D. Prokhorov, V. I. Ponomarenko, and A. S. Karavaev. “Equations of a time-delay system under external action reconstructed from time series”. *Technical Physics Letters* 30 (2004), pp. 78–81. DOI: 10.1134/1.1646723.

[33] V. I. Ponomarenko and M. D. Prokhorov. "Reconstruction of the equation of a system with two delay times from time series". *Technical Physics Letters* 30.11 (2004), pp. 940–943. DOI: 10.1134/1.1829349.

[34] M. D. Prokhorov et al. "Reconstruction of time-delayed feedback systems from time series". *Physica D: Nonlinear Phenomena* 203.3 (2005), pp. 209–223. DOI: 10.1016/j.physd.2005.03.013.

[35] V. I. Ponomarenko and M. D. Prokhorov. "Recovery of equations of coupled time-delay systems from time series". *Technical Physics Letters* 31 (2005), pp. 64–67. DOI: 10.1134/1.1859503.

[36] M. D. Prokhorov and V. I. Ponomarenko. "Reconstructing model equations of the chains of coupled delay-feedback systems from time series". *Technical Physics Letters* 34.4 (2008), pp. 331–334. DOI: 10.1134/S1063785008040184.

[37] M. J. Bünner et al. "Recovery of scalar time-delay systems from time series". *Physics Letters A* 211.6 (1996), pp. 345–349. DOI: 10.1016/0375-9601(96)00014-X.

[38] R. Hegger et al. "Identifying and Modeling Delay Feedback Systems". *Physical Review Letters* 81 (3 July 1998), pp. 558–561. DOI: 10.1103/PhysRevLett.81.558.

[39] A. S. Karavaev, V. I. Ponomarenko, and M. D. Prokhorov. "Reconstruction of scalar time-delay system models". *Technical Physics Letters* 27.5 (2001), pp. 414–418. DOI: 10.1134/1.1376769.

[40] S. N. Wood. "Partially specified ecological models". *Ecological Monographs* 71.1 (2001), pp. 1–25. DOI: 10.1890/0012-9615(2001)071[0001:PSEM]2.0.CO;2.

[41] W. Horbelt, J. Timmer, and H. U. Voss. "Parameter estimation in nonlinear delayed feedback systems from noisy data". *Physics Letters A* 299.5 (2002), pp. 513–521. DOI: 10.1016/S0375-9601(02)00748-X.

[42] V. Deshmukh. "Parametric Estimation for Delayed Nonlinear Time-Varying Dynamical Systems". *Journal of Computational and Nonlinear Dynamics* 6.4 (2011). DOI: 10.1115/1.4003626.

[43] C. Dai et al. "Seeker optimization algorithm for parameter estimation of time-delay chaotic systems". *Physical Review E* 83 (3 Mar. 2011), p. 036203. DOI: 10.1103/PhysRevE.83.036203.

[44] I. V. Sysoev, V. I. P. B. P. Bezruchko, and M. D. Prokhorov. "Reconstruction of parameters and unobserved variables of a semiconductor laser with optical feedback from intensity time series". *Physical Review E* 101 (4 Apr. 2020), p. 042218. DOI: 10.1103/PhysRevE.101.042218.

[45] J. D. Turner and B. P. Mann. "Empirical characteristic multiplier estimation: A comparison of methods and improvements". *Journal of Sound and Vibration* 511 (2021), p. 116053. DOI: 10.1016/j.jsv.2021.116053.

[46] J. Ahn and N. Hogan. "Improved Assessment of Orbital Stability of Rhythmic Motion with Noise". *PLoS ONE* 10.3 (2015). DOI: 10.1371/journal.pone.0119596.

[47] M. Casdagli et al. "State space reconstruction in the presence of noise". *Physica D: Nonlinear Phenomena* 51.1 (1991), pp. 52–98. DOI: 10.1016/0167-2789(91)90222-U.

[48] J. F. Gibson et al. "An analytic approach to practical state space reconstruction". *Physica D: Nonlinear Phenomena* 57.1 (1992). DOI: 10.1016/0167-2789(92)90085-2.

[49] P. Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[50] P. T. Boggs and J. E. Rogers. "Orthogonal Distance Regression". *Statistical Analysis of Measurement Error Models and Applications*. Vol. 112. Contemporary Mathematics. AMS–IMS–SIAM. 1990, pp. 183–194.

[51] S. P. Ellner, Y. Seifu, and R. H. Smith. "Fitting population dynamic models to time-series data by gradient matching". *Ecology* (2002). DOI: 10.1890/0012-9658(2002)083[2256:FPDMTT]2.0.CO;2.

[52] V. I. Ponomarenko and M. D. Prokhorov. "Recovery of systems with a linear filter and nonlinear delay feedback in periodic regimes". *Physical Review E* 78 (6 Dec. 2008), p. 066207. DOI: 10.1103/PhysRevE.78.066207.

[53] M. D. Prokhorov and V. I. Ponomarenko. "Reconstruction of time-delay systems using small impulsive disturbances". *Physical Review E* 80 (6 Dec. 2009), p. 066206. DOI: 10.1103/PhysRevE.80.066206.

[54] F. Sorrentino. "Identification of delays and discontinuity points of unknown systems by using synchronization of chaos". *Physical Review E* 81 (6 June 2010), p. 066218. DOI: 10.1103/PhysRevE.81.066218.

[55] I. V. Sysoev et al. "Recovery of couplings and parameters of elements in networks of time-delay systems from time series". *Physical Review E* 94 (5 Nov. 2016), p. 052207. DOI: 10.1103/PhysRevE.94.052207.

[56] D. Schuhmacher, B.-T. Vo, and B.-N. Vo. "A Consistent Metric for Performance Evaluation of Multi-Object Filters". *IEEE Transactions on Signal Processing* 56.8 (2008), pp. 3447–3457. DOI: 10.1109/TSP.2008.920469.

[57] C. H. Papadimitriou and K. Steiglitz. “Combinatorial Optimization: Algorithms and Complexity”. Dover Publications, 1982. Chap. Weighted Matching, pp. 247–270.

[58] Y. Ishida and T. Yamamoto. “Linear and Nonlinear Rotordynamics: A Modern Treatment with Applications”. John Wiley & Sons, Inc., 2012. Chap. Vibrations of an Asymmetrical Shaft and an Asymmetrical Rotor, pp. 93–113.

[59] C. E. Rasmussen and C. K. I. Williams. “Gaussian Processes for Machine Learning”. MIT Press, 2006. Chap. Regression, pp. 7–31.

[60] B. P. Mann et al. “Simultaneous Stability and Surface Location Error Predictions in Milling”. *Journal of Manufacturing Science and Engineering* 127.3 (July 2005), pp. 446–453. DOI: 10.1115/1.1948394.

[61] G. Yücesan and Y. Altintaş. “Improved modelling of cutting force coefficients in peripheral milling”. *International Journal of Machine Tools and Manufacture* 34.4 (1994), pp. 473–487. DOI: 10.1016/0890-6955(94)90079-5.

[62] Y. Altintas. *Manufacturing Automation: Metal Cutting Mechanics, Machine Tool Vibrations, and CNC Design*. 2nd ed. Cambridge University Press, 2012.

[63] B. P. Mann et al. “Stability of up-milling and down-milling, part 2: experimental verification”. *International Journal of Machine Tools and Manufacture* 43.1 (2003), pp. 35–40. DOI: 10.1016/S0890-6955(02)00160-8.

[64] B. P. Mann. “Dynamic Models of Milling and Broaching”. PhD thesis. Saint Louis, Missouri: Washington University, Sever Institute of Technology, Department of Mechanical Engineering, May 2003.

[65] S. A. Tobias. *Machine Tool Vibration*. London: Blackie, 1965.

[66] J. Tlusty et al. *Selbsterregte schwingungen an werkzeugmaschinen*. Berlin: VEB Verlag Technik, 1962.

[67] H. Merritt. “Theory of self-excited machine tool chatter”. *Journal of Engineering for Industry* 87.4 (1965), pp. 447–454. DOI: 10.1115/1.3670861.

[68] J. R. Pratt and A. H. Nayfeh. “Design and modeling for chatter control”. *Nonlinear Dynamics* 19.1 (1999), pp. 49–69. DOI: 10.1023/A:1008322520352.

[69] F. Koenisberger and J. Tlusty. “Machine Tool Structures”. Vol. 1. Pergamon Press, 1970. Chap. Stability Against Chatter. DOI: 10.1016/C2013-0-02227-X.

[70] R. L. Kegg. “Cutting dynamics in machine tool chatter”. *Journal of Engineering for Industry* 87.4 (1965), pp. 464–470. DOI: 10.1115/1.3670863.

[71] R. Shridar, R. E. Hohn, and G. W. Long. "A stability algorithm for the general milling process". *Journal of Engineering for Industry* 90.2 (1968), pp. 330–334. DOI: 10.1115/1.3604637.

[72] N. H. Hanna and S. A. Tobias. "A theory of nonlinear regenerative chatter". *Journal of Engineering for Industry* 96.1 (1974), pp. 247–255. DOI: 10.1115/1.3438305.

[73] J. Tlusty and F. Ismail. "Special aspects of chatter in milling". *Journal of Vibration, Acoustics, Stress,and Reliability in Design* 105.1 (1983), pp. 24–32. DOI: 10.1115/1.3269061.

[74] I. Grabec. "Chaotic dynamics of the cutting process". *International Journal of Machine Tools and Manufacture* 28.1 (1988), pp. 19–32. DOI: 10.1016/0890-6955(88)90004-1.

[75] G. Stépán. *Retarded Dynamical Sytems: Stability and Characteristic Functions*. John Wiley & Sons, 1989.

[76] I. Minis and R. Yanushevsky. "A New Theoretical Approach for the Prediction of Machine Tool Chatter in Milling". *Journal of Engineering for Industry* 115.1 (1993), pp. 1–8. DOI: 10.1115/1.2901633.

[77] S. Smith and J. Tlusty. "An Overview of Modeling and Simulation of the Milling Process". *Journal of Engineering for Industry* 113.2 (1991), pp. 169–175. DOI: 10.1115/1.2899674.

[78] Y. Altintaş and E. Budak. "Analytical Prediction of Stability Lobes in Milling". *CIRP Annals* 44.1 (1995), pp. 357–362. DOI: 10.1016/S0007-8506(07)62342-7.

[79] H. Schulz and T. Moriwaki. "High-speed Machining". *CIRP Annals* 41.2 (1992), pp. 637–643. DOI: 10.1016/S0007-8506(07)63250-8.

[80] A. H. Nayfeh, C. M. Chin, and J. Pratt. "Dynamics and Chaos in Manufacturing Processes". Ed. by F. C. Moon. New York, NY: John Wiley & Sons, Inc., 1997. Chap. Applications of perturbation methods to tool chatter dynamics.

[81] B. Balachandran. "Nonlinear dynamics of milling processes". *Philosophical Transiactions of the Royal Society A* 359.1781 (2001), pp. 793–819. DOI: 10.1098/rsta.2000.0755.

[82] M. A. Davies and B. Balachandran. "Impact Dynamics in Milling of Thin-Walled Structures". *Nonlinear Dynamics* 22 (2000), pp. 375–392. DOI: 10.1023/A:1008364405411.

[83] Y. Altintas. "Analytical Prediction of Three Dimensional Chatter Stability in Milling". *JSME International Journal* 44.3 (2001), pp. 717–723. DOI: 10.1299/jsmec.44.717.

[84] R. G. Landers and A. G. Ulsoy. "Nonlinear Feed Effect in Machining Chatter Analysis". *Journal of Manufacturing Science and Engineering* 130.1 (2008). DOI: 10.1115/1.2783276.

[85] B. R. Patel, B. P. Mann, and K. A. Young. "Uncharted islands of chatter instability in milling". *International Journal of Machine Tools and Manufacture* 48.1 (2008), pp. 124–134. DOI: 10.1016/j.ijmachtools.2007.06.009.

[86] E. Butcher and B. Mann. "Delay Differential Equations: Recent Advances and New Directions". Ed. by D. E. Gilsinn, T. Kalmár-Nagy, and B. Balachandran. Boston, MA: Springer, 2009. Chap. Stability Analysis and Control of Linear Periodic Delayed Systems Using Chebyshev and Temporal Finite Element Methods, pp. 93–129. DOI: 10.1007/978-0-387-85595-0_4.

[87] F. A. Khasawneh et al. "Increased Stability of Low-Speed Turning Through a Distributed Force and Continuous Delay Model". *Journal of Computational and Nonlinear Dynamics* 4.4 (2009). DOI: 10.1115/1.3187153.

[88] O. A. Bobrenkov et al. "Analysis of milling dynamics for simultaneously engaged cutting teeth". *Journal of Sound and Vibration* 329.5 (2010), pp. 585–606. DOI: 10.1016/j.jsv.2009.09.032.

[89] E. A. Butcher and O. A. Bobrenkov. "On the Chebyshev spectral continuous time approximation for constant and periodic delay differential equations". *Communications in Nonlinear Science and Numerical Simulation* 16.3 (2011), pp. 1541–1554. DOI: 10.1016/j.cnsns.2010.05.037.

[90] A. G. Ulsoy and R. Gitik. "On the Convergence of the Matrix Lambert W Approach to Solution of Systems of Delay Differential Equations". *Journal of Dynamic Systems, Measurement, and Control* 142.2 (2019). DOI: 10.1115/1.4045368.

[91] R. G. Landers et al. "Peripheral milling of thin titanium plates: modelling, analysis, and process planning". *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 225.6 (2011), pp. 783–798. DOI: 10.1177/09544054JEM1994.

[92] M. H. Kurdi et al. "A Robust Semi-Analytical Method for Calculating the Response Sensitivity of a Time Delay System". *Journal of Vibration and Acoustics* 130.6 (2008). DOI: 10.1115/1.2981093.

[93] C. M. Mack et al. "Burst and principal components analyses of MEA data for 16 chemicals describe at least three effects classes". *NeuroToxicology* 40 (Jan. 2014), pp. 75–85. DOI: 10.1016/j.neuro.2013.11.008.

[94] J. D. Turner and Y. Zhu. "Three-dimensional structures from electrically-activated self-folding of polymer sheets". *Summer Undergraduate Research Symposium*. North Carolina State University. Raleigh, NC, July 2014.

[95] J. D. Turner and B. P. Mann. "Sensitivity of Final Field Position to the Punt Initial Conditions in American Football". *Proceedings of the ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Vol. 8: 28th Conference on Mechanical Vibration and Noise. ASME. Charlotte, North Carolina, Aug. 2016. DOI: 10.1115/DETC2016-60235.

[96] J. D. Turner et al. "A nonlinear model for the characterization and optimization of athletic training and performance". *Biomedical Human Kinetics* 9.1 (2017), pp. 82–93. DOI: 10.1515/bhk-2017-0013.

[97] J. D. Turner, L. H. Manring, and B. P. Mann. "Reinforcement Learning for Active Damping of Harmonically Excited Pendulum with Highly Nonlinear Actuator". *Nonlinear Structures and Systems*. Ed. by G. Kerschen, M. R. W. Brake, and L. Renson. Vol. 1. Conference Proceedings of the Society for Experimental Mechanics Series. Society for Experimental Mechanics. Springer, 2020, pp. 119–123. DOI: 10.1007/978-3-030-12391-8_15.

[98] X.-S. Wang, J. D. Turner, and B. P. Mann. "Constrained attractor selection using deep reinforcement learning". *Journal of Vibration and Control* (2020). DOI: 10.1177/1077546320930144.

Biography

James D. Turner first became involved with scientific research under the supervision of Dr. Timothy Shafer at the U.S. Environmental Protection Agency [93]. He graduated as a valedictorian from North Carolina State University (NCSU) in 2015 with a B.S. in Mechanical Engineering and Minors in Spanish and Computer Programming, where he was awarded Goldwater and Park Scholarships. While at NCSU, he conducted research under Dr. Yong Zhu [94] and interned for Applied Research Associates and Deere & Company.

James began his Ph.D. studies at Duke University in 2015, under the direction of Dr. Brian Mann. He was awarded James B. Duke, Pratt/Gardner, and National Defense Science and Engineering Graduate (NDSEG) Fellowships. While at Duke, he contributed to research in nonlinear systems and reinforcement learning [19, 45, 95–98]. Additionally, he conducted research on multitarget tracking under the guidance of Dr. James McMahon (U.S. Naval Research Laboratory) and Dr. Michael Zavlanos (Duke University) for a M.S. degree in Computer Science.

Following graduation, James will work for the U.S. Naval Research Laboratory in Washington, D.C., as a full-time researcher.