STATE OF CALIFORNIA • DEPARTMENT OF TRANSPORTATION

TECHNICAL REPORT DOCUMENTATION PAGE

TR-0003 (REV 04/2024)

Lock Form

1. REPORT NUMBER	2. GOVERNMENT ASSOCIATION NUMBER	3. RECIPIENT'S CATALOG NUMBER
CA24-3703		

4. TITLE AND SUBTITLE	5. REPORT DATE
BRACE^2: Bridge Rapid Assessment Center for Extreme Events Phase I Final Report	May 2024
	6. PERFORMING ORGANIZATION CODE
7. AUTHOR	8. PERFORMING ORGANIZATION REPORT NO.
Chrystal Chern, Claudio Perez, and Khalid M. Mosalam	
9. PERFORMING ORGANIZATION NAME AND ADDRESS	10. WORK UNIT NUMBER
Pacific Earthquake Engineering Research Center University of California Berkeley, CA 94720	11. CONTRACT OR GRANT NUMBER
12. SPONSORING AGENCY AND ADDRESS	13. TYPE OF REPORT AND PERIOD COVERED
California Department of Transportation Division of Research, Innovation and System Information P.O. Box 942873 Sacramento, CA 94273-0001	Final Report 8/1/2020 – 1/31/2024
	14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES

16. ABSTRACT

This report documents the findings of the *BRACE*^{2} project. In order to accomodate the interdisciplinary nature of such an undertaking, this document is divided into four parts which are each designed for a particular target audience.

Part I develops a high-level overview of the *BRACE*^{2} web application that is targeted at decision makers and engineers. Chapter 1 introduces essential concepts that lay the foundation for structural health monitoring. Chapter 2 provides a user's guide for realizing these concepts with the health monitoring platform, the primary deliverable of the *BRACE*^{2} project. Finally, Chapter 3 documents an example scenario.

Part II is targeted at practicing engineers and also researchers. The concept of a predictor is developed in detail, which provides the fundamental abstraction through which structural health monitoring is performed. The predictors that perform bridge response simulation are Type I predictors. They consist of a family of structural analysis models. The general process for constructing a Type I predictor is described in Chapter 4. The predictors that perform data-driven bridge dynamics identification are Type II predictors. They consist of a family of system identification procedures. A practical guide to viewing, configuring, and interpreting the results of Type II predictors is provided in Chapter 5.

Part III documents the pilot studies that have been performed as part of Phase I of the *BRACE*^{2} project. This material will be a valuable resource for engineers that may be tasked with developing new digital twins to be added to the platform. These studies consist of 21 *partial* digital twins that are documented in Chapter 6 and one *complete* digital twin centered around Caltrans bridge No. 33-0214L (route 580/238 separation structure, or Hayward Bridge, for short) that is documented in Chapters 7 to 9. Specifically, Chapter 7 describes in depth the physical properties of the bridge, Chapter 8 gives a detailed account of the high-fidelity pilot model that was developed for the complete digital twin, and Chapter 9 documents an example simulation.

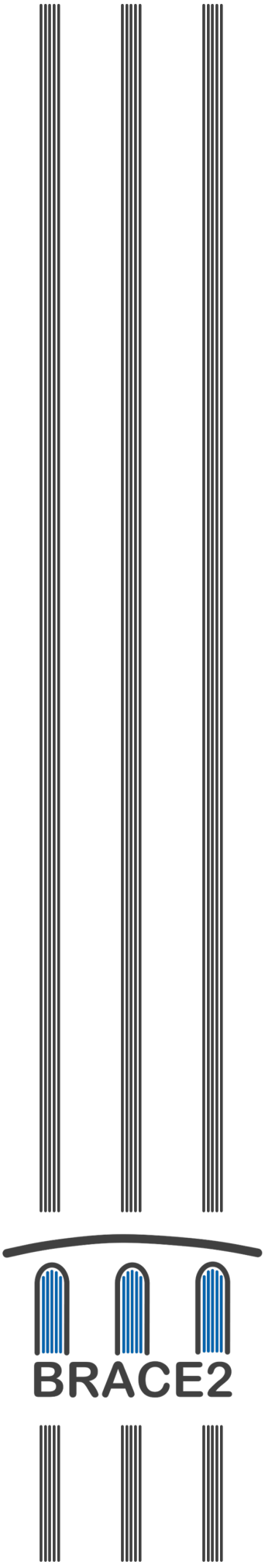
Part IV collects technical documentation which will aid in the successful transfer, deployment, and long-term operation of the health monitoring platform. The target audience includes system administrators (Chapters 10 and 11), future researchers (Chapters 12 and 13), and source code maintainers (Chapter 14).

17. KEY WORDS	18. DISTRIBUTION STATEMENT	
Structural health monitoring, digital twin, bridges, highway, system identification, structural analysis, data-driven, regional-scale		
19. SECURITY CLASSIFICATION *(of this report)*	20. NUMBER OF PAGES 179	21. COST OF REPORT CHARGED

DISCLAIMER STATEMENT

This document is disseminated in the interest of information exchange. The contents of this report reflect the views of the authors who are responsible for the facts and accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the State of California or the Federal Highway Administration. This publication does not constitute a standard, specification or regulation. This report does not constitute an endorsement by the Department of any product described herein.

For individuals with sensory disabilities, this document is available in alternate formats. For information, call (916) 654-8899, TTY 711, or write to California Department of Transportation, Division of Research, Innovation and System Information, MS-83, P.O. Box 942873, Sacramento, CA 94273-0001.



BRACE^2: Bridge Rapid Assessment Center for Extreme Events

Phase I Final Report

Chrystal Chern, Claudio Perez, and Khalid M. Mosalam

University of California, Berkeley

Pacific Earthquake Engineering Research Center

May 12, 2024

Contents

List of Figures

List of Tables

Executive Summary

This report documents the findings of the *BRACE*^{2} project. In order to accomodate the interdisciplinary nature of such an undertaking, this document is divided into four parts which are each designed for a particular target audience.

Part I develops a high-level overview of the *BRACE*^{2} web application that is targeted at decision makers and engineers. Chapter 1 introduces essential concepts that lay the foundation for structural health monitoring. Chapter 2 provides a user's guide for realizing these concepts with the health monitoring platform, the primary deliverable of the *BRACE*^{2} project. Finally, Chapter 3 documents an example scenario.

Part II is targeted at practicing engineers and also researchers. The concept of a predictor is developed in detail, which provides the fundamental abstraction through which structural health monitoring is performed. The predictors that perform bridge response simulation are Type I predictors. They consist of a family of structural analysis models. The general process for constructing a Type I predictor is described in Chapter 4. The predictors that perform data-driven bridge dynamics identification are Type II predictors. They consist of a family of system identification procedures. A practical guide to viewing, configuring, and interpreting the results of Type II predictors is provided in Chapter 5.

Part III documents the pilot studies that have been performed as part of Phase I of the *BRACE*^{2} project. This material will be a valuable resource for engineers that may be tasked with developing new digital twins to be added to the platform. These studies consist of 21 *partial* digital twins that are documented in Chapter 6 and one *complete* digital twin centered around Caltrans bridge No. `33-0214L` (route 580/238 separation structure, or Hayward Bridge, for short) that is documented in Chapters 7 to 9. Specifically, Chapter 7 describes in depth the physical properties of the bridge, Chapter 8 gives a detailed account of the high-fidelity pilot model that was developed for the complete digital twin, and Chapter 9 documents an example simulation.

Part IV collects technical documentation which will aid in the successful transfer, deployment, and long-term operation of the health monitoring platform. The target audience includes system administrators (Chapters 10 and 11), future researchers (Chapters 12 and 13), and source code maintainers (Chapter 14).

Acknowledgements

This research received funding support from California Department of Transportation (Caltarns) for the *Bridge Rapid Assessment Center for Extreme Events* (*BRACE*^{2}) project, Task Order 001 of the *PEERBridge Program* agreement 65A0774 to the Pacific Earthquake Engineering Research (PEER) Center. The authors thank the sponsor for this financial support. Opinions and findings expressed in this report are those of the authors and do not necessarily reflect those of the sponsor.

The authors would like to express their appreciation for major support provided by Tom Ostrom, Division Chief, Caltrans – Division of Engineering Services (DES); for the technical and logistical support provided by the team led by Chris Traina, Office Chief – Office of Earthquake Engineering Analysis & Research (OEEAR) and by Sharon Yen, Senior Transportation Engineer – Division of Research, Innovation & System Information (DRISI). Caltrans senior engineers Pat Hipley, Mark Mahan, and Tom Shantz significantly contributed to the original ideas of the *BRACE*^{2} initial proposal. Efforts by California Geological Survey team led by Hamid Haddadi, Program Manager, California Strong Motion Instrumentation Program, were essential for data transfer from the Hayward Bridge to the PEER Server in real time. The *BRACE*^{2} team thanks Dr. Clifford Roblee, Senior Research Engineer, Caltrans and the g2F team shared valuable information about their findings that benefited the bridge models of *BRACE*^{2}. Technical and administrative contributions from the following current and early collaborators from the University of California, Berkeley (UCB), the Pacific Earthquake Engineering Research (PEER) Center, and the University of Nevada, Reno (UNR) are greatly appreciated:

- EuiHyun Choi, PhD Student, UCB,
- Zahra Faeli, Postdoc Researcher, UNR,
- Yuqing Gao, Postdoc Researcher, UCB,
- Selim Günay, Visiting Scholar, PEER,
- Fan Hu, PhD Student, UCB,
- Amarnath Kasalanati, Associate Director, PEER,
- Chenglong Li, PhD Student, UCB,
- Reza Mohammadi, PhD Student, UNR,
- Ramin Motamed, Associate Professor, UNR,
- Sifat Muin, Postdoc Researcher, UCB,
- Arpit Nema, Postdoc Researcher, PEER,
- Martin Neuenschwander, Visiting Scholar, PEER,
- Gaofeng Su, PhD Student, UCB,
- Nikolay Velkov, PhD Student, UCB, and
- Guanren Zhou, PhD Student, UCB.

Part I

Platform

Part I develops a high-level overview of the *BRACE*^{2} web application that is targeted at decision makers and engineers. Chapter 1 introduces essential concepts that lay the foundation for structural health monitoring. Chapter 2 provides a user's guide for realizing these concepts with the health monitoring platform, the primary deliverable of the *BRACE*^{2} project. Finally, Chapter 3 documents an example scenario.

Chapter 1

Framework Overview

The *BRACE*^{2} health monitoring platform provides a link between: (1) decision makers who are responsible for the safety of the public, (2) practicing engineers that understand the structural mechanics and are fit to develop prediction models, and (3) researchers or specialists in the field of structural health monitoring who are studying and developing new metrics for health assessment. The *BRACE*^{2} project objectives are achieved by designing a framework of the platform to work in terms of four key abstractions: assets, events, predictors, and metrics, Fig. 1. These abstractions each embody an essential component of a complete structural health monitoring framework.

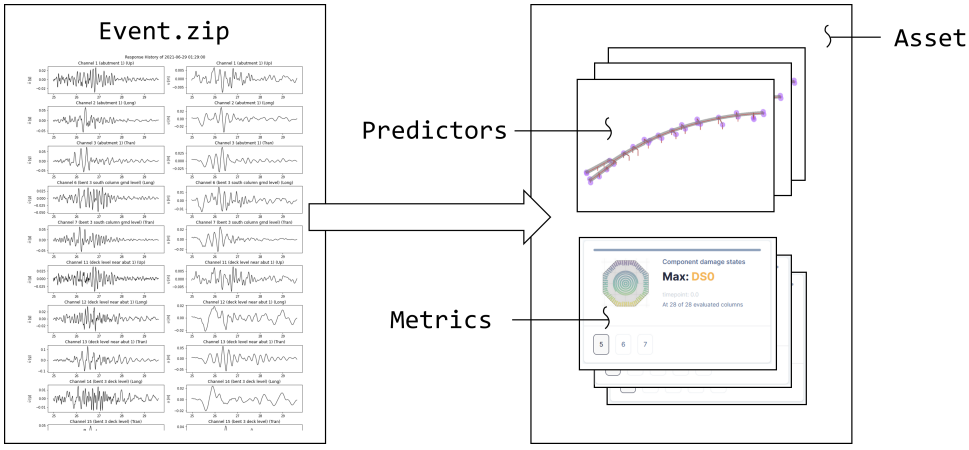


Fig. 1: The asset, predictor, event, and metric represent the central abstractions of the *BRACE*^{2} platform for bridge health monitoring.

An asset is a unique physical object for which there is interest in its structural health. For example, Caltrans bridge No. `33-0214L` is an asset belonging to the Caltrans *inventory*. The term metric is used to refer to those final quantities that are presented before a decision maker and serve as indicators of structural health or damage. These metrics are analogous to indicators like body temperature and heart rate that a medical specialist uses to diagnose a sick patient. They are computed, in real time or near real time, following events, e.g., a major earthquake or a man-made accident, due to pre-defined triggering mechanisms and their associated thresholds. Health metrics are generated by predictors, which either simulate the structural response or identify the structural properties. In order to conduct a meaningful structural health assessment, the assessor must have a

sound understanding of how each metric is composed and interpreted.

1.1 Assets

The asset abstraction represents the primary subject of structural health monitoring. Naturally, the inventory of assets consists of a wide variety of structural designs, each with distinct vulnerabilities and strengths. It is essential that the concept of an asset on the platform is able to account for these contextual details. For example, several older bridges in the Caltrans inventory are designed with continuous flared columns like that shown in Fig. 2. Structures with this design are particularly vulnerable to certain failure modes. An experimental investigation was conducted by Sanchez et al. [76] and a numerical investigation of the continuous detail is conducted by Soleimani et al. [84]. This issue is addressed by Caltrans [13]. In order to incorporate these details into the decision making process, the platform represents assets with a personalized *structure profile*. This profile has been designed to integrate with the Federal Highway Administration's National Bridge Inventory (NBI), as documented in Section 2.2.

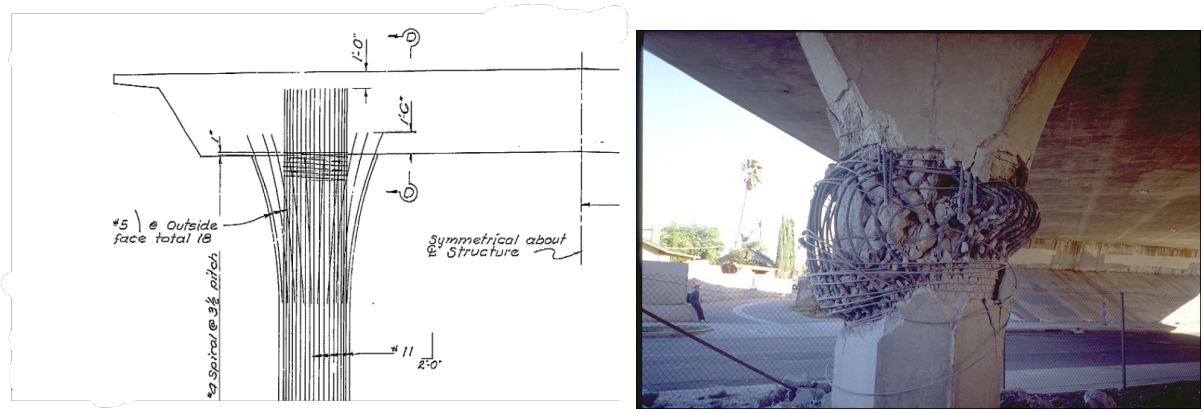


Fig. 2: Example of an important feature included in the inventory of assets. Left: Obsolete continuous flare detail from Caltrans bridge No. `04-0236`, Painter St. overcrossing. Right: Column failure of the Mission Gothic Bridge following the 1994 Northridge earthquake (Photo courtesy of PEER NISEE, [6]).

The first phase of the *BRACE*^{2} project was originally tasked to conduct a pilot study of five bridges, with particular focus on the *route 580/238 separation structure* (Caltrans bridge No. `33-0214L`). However, this study was voluntarily extended to a total of **22** bridges. These assets are further discussed in Chapter 6 of Part III where the original pilot study is also documented.

1.2 Predictors

The predictor abstraction refers to any analysis technique that an engineer can configure beforehand to be run when an event is triggered. This covers both methods based on conventional structural analysis and methods based on data fitting. There are several distinct procedures that fall within these classes. Because of the rate of active development in the field of structural health monitoring, the

development of these procedures is expected to continue growing. It is useful to consider prediction methods as belonging to one of two families: Type I predictors and Type II predictors. Refer to Fig. 3 for representative examples.

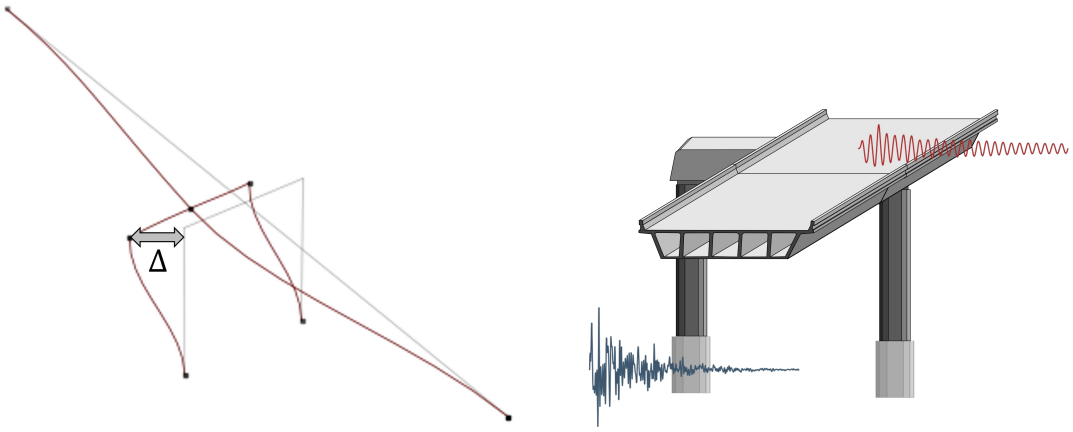


Fig. 3: Predictors are implemented as structural analysis models (Type I, Left) and data-driven procedures (Type II, Right).

1.2.1 Type I: Heirloom Methods

In the field of civil engineering, the most ubiquitous class of prediction techniques consists of those techniques that leverage structural analysis models. These techniques are particularly attractive because they are readily accessible to the large workforce of civil engineers that routinely develops models of this type for problems in design and rehabilitation. Structural analysis models are constructed by characterizing the properties of each structural component.

It is important to note that developing a finite element model for structural health monitoring is fundamentally distinct from the more typical practice of modeling for design or rehabilitation. Many common guidelines and best practices which have been developed for these latter purposes would not be generally well suited for structural health monitoring. Currently, models of this type that are employed on the platform leverage advanced finite elements that are typically not available through basic commercial products (e.g., [85]).

For each asset that is monitored on the health monitoring platform, several structural analysis models may be required. A pilot study for this class of prediction has been carried out for the route 580/238 separation structure (Caltrans bridge number `33-0214L`). This is presented in Part III.

1.2.2 Type II: Data-Driven Methods

Data-driven methods take advantage of measured response quantities to reduce the number of modeling parameters when an analyst is faced with the challenge of defining them in Type I predictors. This reduction can be quite significant, and with adequate data sources, can be accurately configured in just hours, as opposed to the weeks that an engineer may typically require to assemble a traditional structural analysis model of a bridge system. Among the broad range of methods that

fall in this category, the health monitoring platform mainly employs methods based on the dynamic equilibrium equations of motion, called *system identification* methods.

Methods of system identification (Chapters 5 and 13) use recorded response histories to characterize the system dynamics of the structure at hand, a bridge in this case. These dynamics are expressed in terms of either an input-output relationship in the frequency domain, called a transfer function, or a state space representation in the time domain, which is a set of first-order differential equations, expressed in a matrix form. Generally, the parameters of the dynamic equilibrium equations are used as identifying characteristics of the structure. Health monitoring then entails tracking the data-fitted identity of a structure over time.

All structural responses uploaded to the health monitoring platform are automatically analyzed by a collection of pre-configured system identification predictors. New predictors can be configured by interested knowledgable users in the future (Chapters 5 and 13).

1.3 Metrics

The platform provides metrics to decision makers in the form of computed numerical (tabulated or plotted) information that can inform judgment calls. The envisioned judgment calls can include closing a particular bridge, prioritizing locations to send early responders to for in-depth inspection, or declaring that no action is needed. Given this broad range of demands, it is useful to distinguish the following applications one may expect a metric to target:

Component condition. Applications include plans for local inspection and repair or retrofit.

Structure condition. Applications include operational decisions such as road closures or deployment of on-site inspection teams.

Inventory condition. Applications include analyzing transportation networks and prioritizing sites for inspection.

Engineers will be inclined to target component-level response metrics. Such metrics leverage familiar concepts like reinforcing bar strain and can be directly related to well-documented experimental phenomena like concrete confinement or cover spalling. However, these metrics are likely the least predictable. They are most sensitive to the material, geometry, and boundary condition variations, and there can be multiple possible damage states for the same static, and particularly dynamic, loading due to the path-dependent and random characteristics affecting the nonlinear behavior, including soil-foundation-structure interaction and damage propagation. The best use for such metrics is compiling the structure-level condition assessments. For example, evaluation of the overall capacity, stiffness, and ductility of a structure, or estimation of the deviation from the linear elastic range of behavior can give information about the serviceability of the structure that can then be used to make decisions regarding operational actions. Inventory level assessments are expected to be the most reliable and useful, but they depend on concepts that are not part of conventional engineering curricula, such as *regional-scale fragility* and *traffic flow dynamics*. On the health monitoring platform, health metric reports are grouped by structure and event, so that assessments can be made regarding the structural behavior at a global level.

1.4 Events and Evaluations

The event abstraction represents a situation that could place the health of a structure in jeopardy and for which an evaluation is desired. Whenever an event is processed at an asset, relevant predictors are automatically deployed to process the event data. This processing culminates in the creation of an evaluation, which is expected to be of immediate interest to the user or the decision maker.

In order to facilitate reliable and secure communication of event data, a secured REST API[1] has been implemented in the health monitoring platform. This API is documented in detail in Chapters 10 and 11. This feature is a critical component to enabling the event (Section 2.5) and evaluation (Section 2.4) pages to be updated in real time.

1.4.1 Data Sources

Throughout Phase I of the *BRACE*^{2} project, data has been provided in real time through a partnership with the California Geological Survey (CGS). Event data is received as a zipped archive of data files. In order to abstract away the formatting details of these files, data is parsed out and organized into an intermediate JSON[2] representation. This data processing is performed by the `quakeio` library (Chapter 14) as it arrives onto the health monitoring platform. For the realization of the needed partnership with CGS to receive the data in a timely manner, the CGS team has been provided credentials to access the Motion API of the *BRACE*^{2} platform (Chapter 11).

The data received from a station by CGS is the voltage reading of the seismic sensors in a digital binary format. Segmentation is performed at the recorder in the field when the amplitudes of motion at specific channels exceed a prescribed threshold. From a pilot investigation, the duration times for data streaming for unprocessed and processed data are within 5 to 6 seconds of each other, Table 1. Moreover, all data is received on the platform server within 18 to 67 seconds, Table 1.

Table 1: Type and duration (i.e., the time difference between the end of recording and when data is received by the *BRACE*^{2} server) of the streamed data.

Type	**Duration [min:sec]**
Bridge unprocessed data	1:02
Bridge processed data	1:07
Geotechnical array unprocessed data	0:18
Geotechnical array processed data	0:24

[1] A REST API is an application programming interface (API) that conforms to the constraints of REST architectural style and allows for interaction with "RESTful" web services. REST stands for REpresentational State Transfer.

[2] JSON is an open standard file and data interchange format that uses human-readable text to store and transmit data objects consisting of attribute-value pairs and arrays. It is commonly used for data interchange, including that of web applications with servers. JSON stands for JavaScript Object Notation.

Chapter 2

Platform Interface

The health monitoring platform is an application that has been developed by the *BRACE*^{2} team to facilitate structural health monitoring of instrumented bridges in real time. This application is the primary outcome of the *BRACE*^{2} project. This chapter serves as a user's guide to the application. Architectural, administrative, and implementation details are presented in Chapter 10. The application consists of three primary interfaces:

1. The *dashboard* renders a high-level snapshot of the bridges that are managed by the platform and available for the decision maker.
2. The *events page* is a comprehensive listing of the event history within the platform.
3. The *inventory page* provides a comprehensive listing of the monitored *assets* by the platform.
4. An *evaluation page* is a report on the health of an asset, generated in response to an event.

Items 1, 2, and 3 are accessible from the primary *navigation panel* (1, Fig. 4), which is accessible from every page within the platform. Evaluation pages (item 4) are accessible from within both the dashboard and events pages (items 1 and 2).

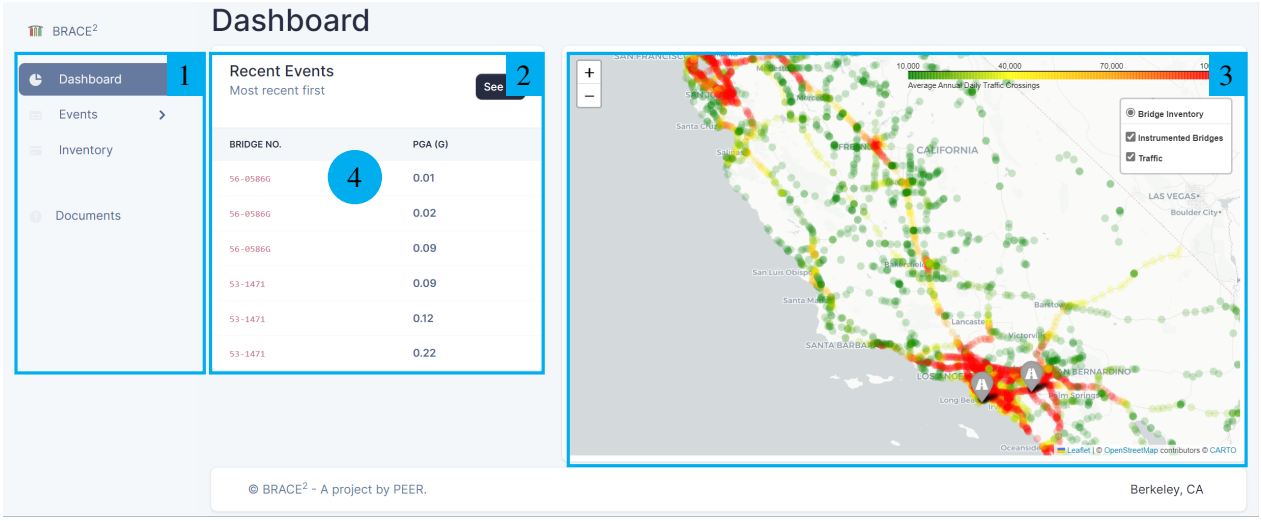


Fig. 4: Dashboard components of the decision-making interface.

2.1 Dashboard

The dashboard (Fig. 4) is the primary "entry point" of the interface. Its purpose is to identify and provide fast access to any evaluations that are of immediate interest. The primary components of the dashboard are a "recent event table" and an "inventory map."

The recent event table (2, Fig. 4) is a listing of links to evaluations that have been recently generated. This table is a filtered subset of the full event table (Section 2.4), which restricts focus only to those stations which are likely of interest at the current time. This filtered view allows the dashboard to serve as a concise landing spot from where decision makers can quickly find relevant reports in times of emergency.

The dashboard map (3, Fig. 4) presents locations of assets (gray balloons) with evaluations listed in the recent event table. Moreover, the map displays the most recent AADT (Average Annual Daily Traffic crossings) data collected from the traffic network (red to green markers and color bar) to assist the decision makers in planning their actions, taking into consideration the network operation and constraints. The AADT data is collected from the most recent (at the time of this report, 2022) publicly available Traffic Counts (a.k.a. Traffic Volumes) report provided by the California Traffic Census Program[3]. The data should be updated by the platform maintainer as new reports become available, to ensure stability of the data structure.

2.2 Inventory

The *inventory* page provides a listing of the assets that are monitored on the platform. An asset is a unique physical object for which there is interest in its structural health. For example, the Caltrans bridge No. `33-0214L` asset is an element of the inventory. The inventory page links assets to the various resources pertaining to them. One such resource is an asset's page on the following link: strongmotioncenter.org, which provides the applicable seismic monitoring metadata that is maintained and updated by the providers of the instrumentation data (California Geological Survey (CGS) in this case) for the physical structure. The current inventory includes the assets shown in Fig. 5. The assets currently include bridges (a total of 22 bridges, not all of them shown in the screenshot table of Fig. 5) that are instrumented by Caltrans and CGS.

A digital twin is a computational model of a unique physical asset that persistently reflects the assets state, context, and behavior. A single digital twin consists of one or more predictors (Part II). *Partial* digital twins are configured for assets that have instrumentation data periodically uploaded to the platform for only high levels of excitation; they contain only Type II predictors. *Full* digital twins are configured for assets that have instrumentation data streamed to the platform for moderate-to-severe levels of excitation; they contain both Type I predictors and Type II predictors. In Fig. 5, partial and full twins are distinguished by their map markers.

[3]Refer to `https://gis.data.ca.gov/datasets/d8833219913c44358f2a9a71bda57f76_0/about`. It is to be noted that the *BRACE*^{2} platform has the capability of checking for new reports and automatically downloading them. However, this is not necessary because such reports are only published once (or less) per year. Moreover, there is some risk of the data format changing and the platform needing to adjust for such an automatic update. Therefore, manual update is deemed sufficient.

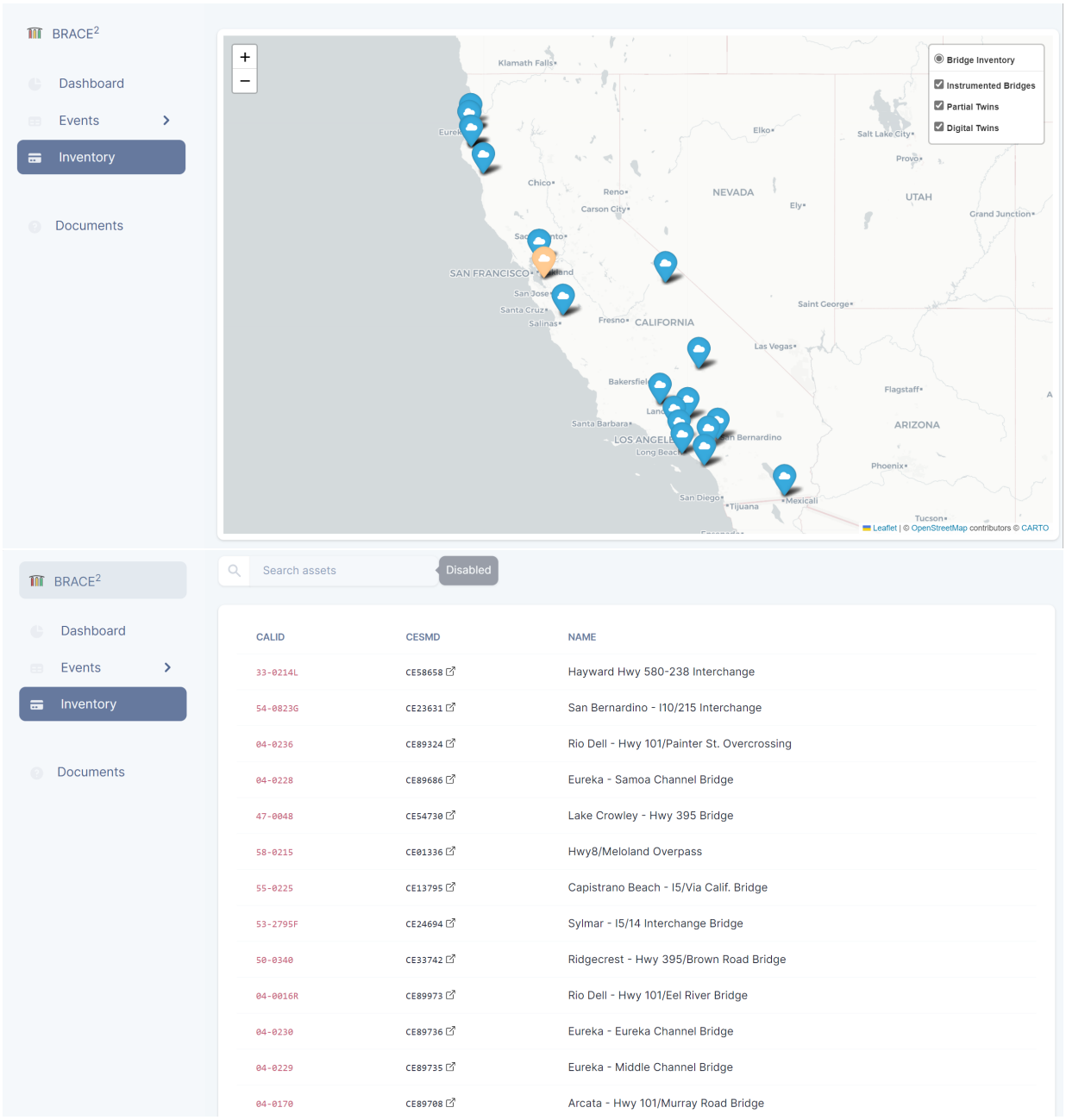


Fig. 5: The inventory page presents the collection of assets with provided references to their structure profiles.

2.3 Structure Profile

Every asset on the *BRACE*^{2} platform is given a *structure profile* page that is accessible from both the Inventory map and the Inventory table. This page displays a sensor map and an Event History summary table providing the peak recorded acceleration and analysis results from Type II predictors (Fig. 6). Following the sensor map, an asset's structure profile additionally provides personalized data that is curated from sources like CESMD and the Federal HighWay Administration's National

Bridge Inventory (Fig. 7).

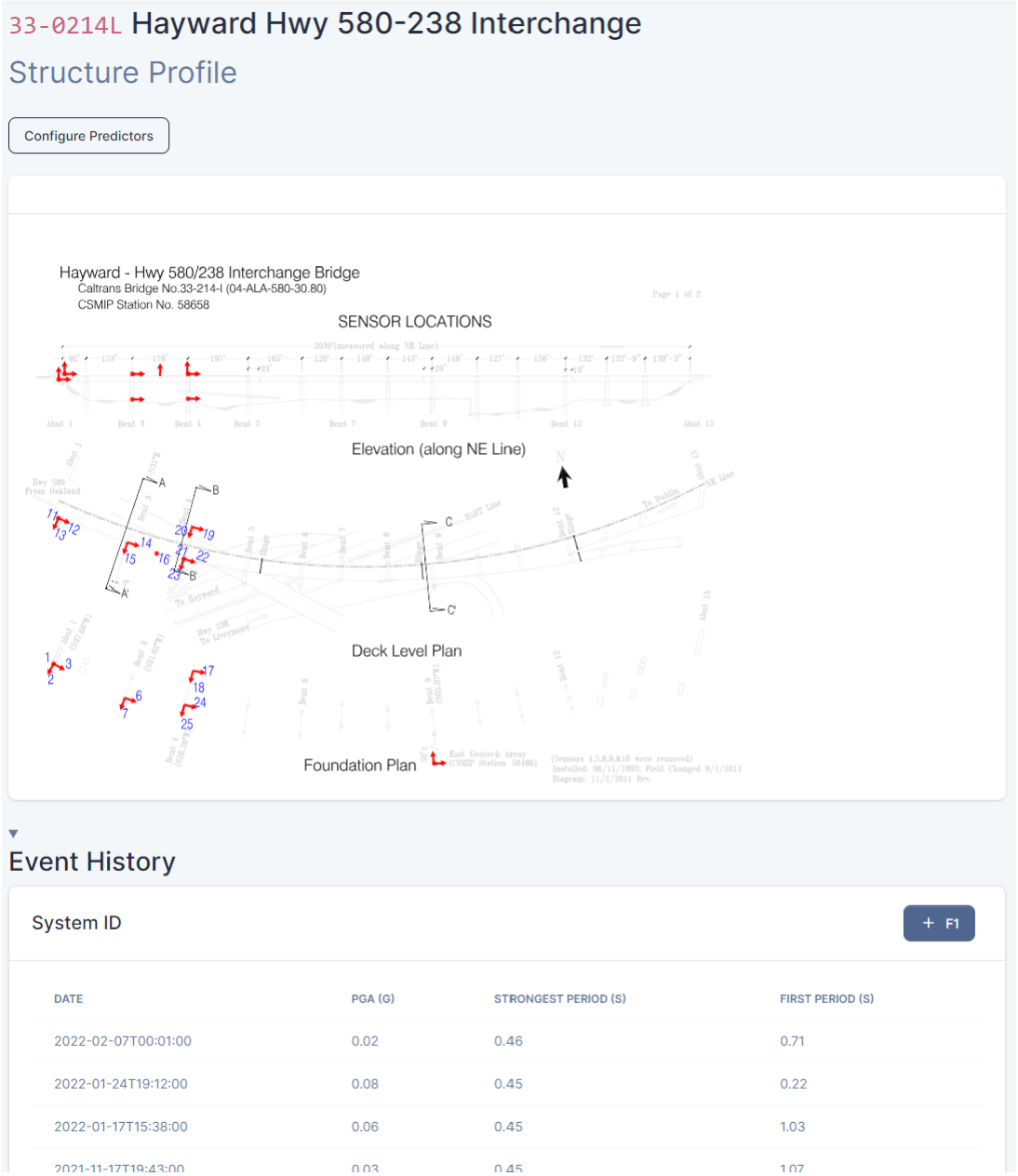


Fig. 6: An asset's structure profile is headed by its CESMD sensor map and event history.

Comprehensive bridge reports tabulating the asset data are available from the structure profile (Section 2.3). These reports can be printed to PDF; most browsers provide the printing functionality with the command `ctrl+P`; see Fig. 8. Two examples of comprehensive bridge reports generated by this process are available in Section 14.1.6.6.

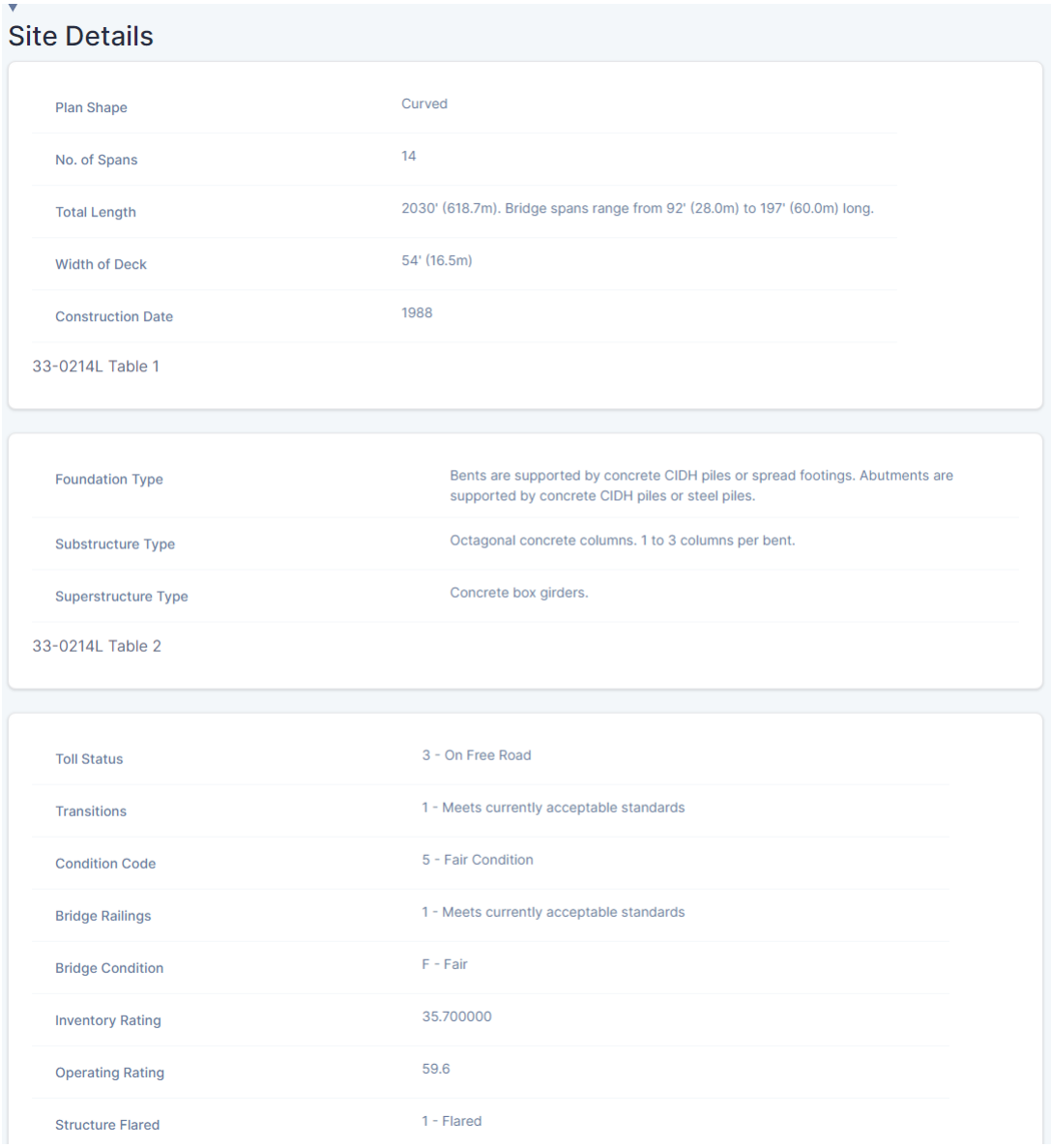


Fig. 7: Following its sensor map, an asset's structure profile provides tabulated data from CESMD and the National Bridge Inventory.

2.4 Events

The events page, Fig. 9, is representative of the link that *BRACE*^{2} provides between the seismic recording station associated with the Center for Engineering Strong Motion Data (CESMD) identification (ID), and the physical structure associated with a bridge ID. This provides decision makers with an interface to the database of seismic events that is kept on the platform in order to maintain a structured history of past health evaluations. This page is not intended to duplicate the functionality provided by strongmotioncenter.org, which should be utilized for obtaining all *seismic* metadata for a particular event. Instead, this page facilitates access to the relevant information for the *BRACE*^{2} decision-making process. This page is updated in real time with motions provided by CGS. Ad-

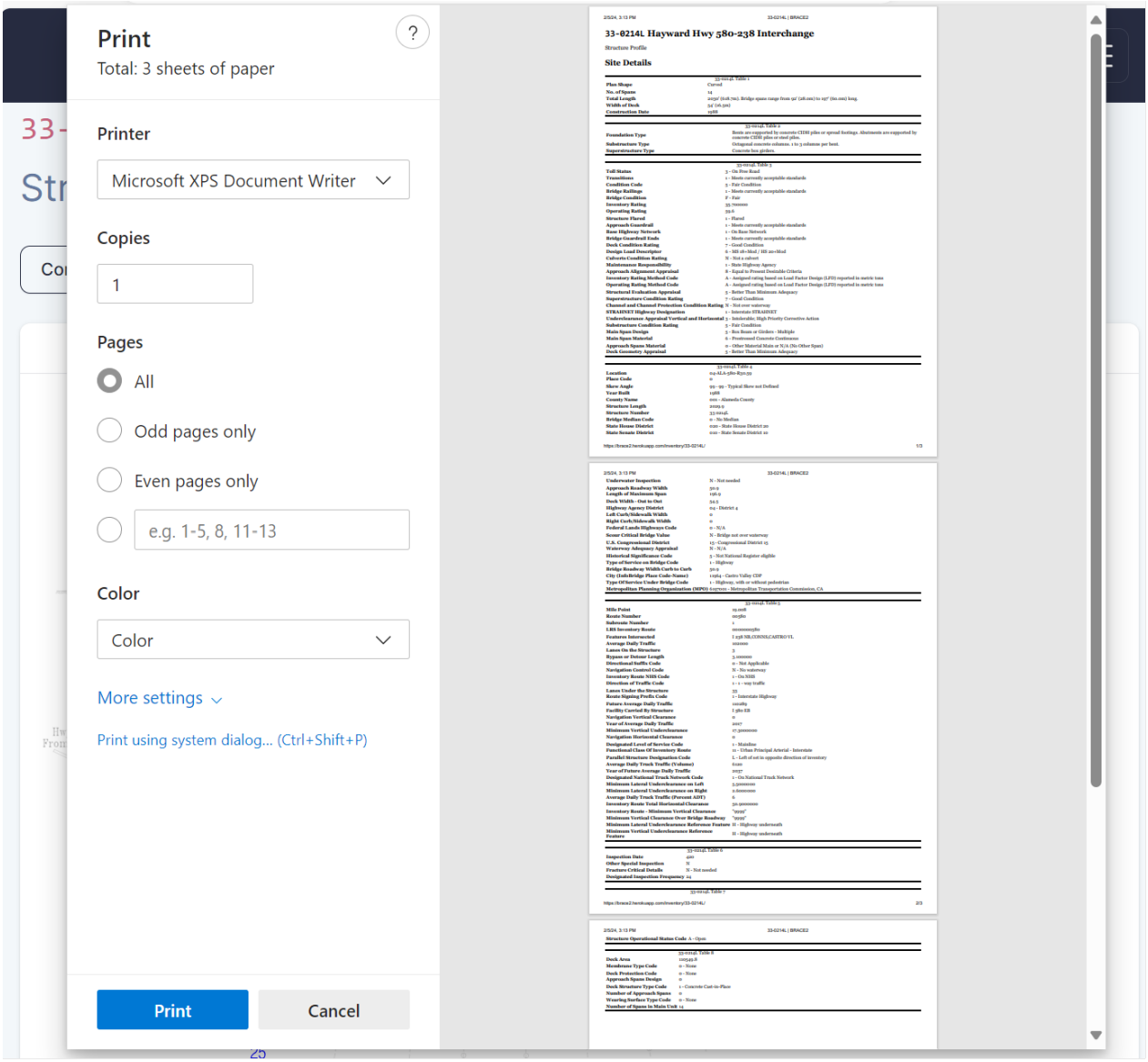


Fig. 8: Print the bridge report (tabulated asset data) from the structure profile.

ditionally, an *email service* has been configured to automatically notify relevant parties when new events occur and are subsequently automatically created as events on the *BRACE*^{2} platform.

2.5 Evaluations and Metrics

An *evaluation* page is a report that is generated on the *BRACE*^{2} platform by automatically assembling the results from several analysis models for a certain asset and a given event. The results of these analysis models are condensed into metrics that can be rapidly leveraged by decision-makers to make judgments concerning the structural health of the asset in question. Based on these metrics of the evaluation, condition assessment and decision-making can be effectively performed by the end-user, i.e., the decision maker.

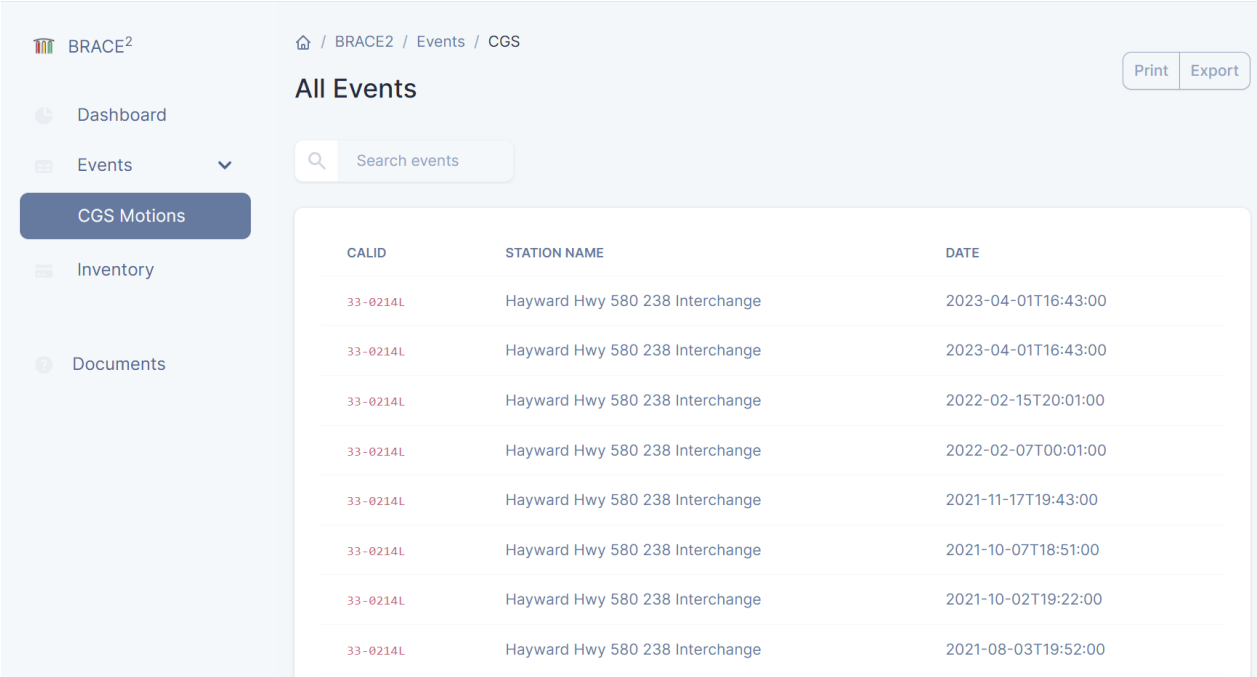


Fig. 9: The events page includes a summary table of all past events.

All evaluation pages include a map of the sensor layout of the structure (Fig. 10). For the full digital twin, the rendering of the Type I predictor is also provided. The metric cards on the evaluation page (1, Fig. 11), collect relevant metrics of an evaluation, e.g., accelerations and drifts, component damage state, or frequency content, and presents them alongside renderings of the analysis models and sensor configuration. The computed metrics from analysis models are provided at all components of interest, e.g., all columns of a bridge. On the other hand, the directly recorded metrics are available for only the components that are instrumented; these are reported in the "Sensor Data" card (4, Fig. 11).

The *BRACE*^{2} platform is designed to allow more than one predictor to report each metric. Each predictor can be selected from the *predictor selection* region of each metric card (e.g., 2, Fig. 11). Presently, OpenSees [56, 54] is being used as the predictor for Type I predictors. Weighing the importance of each prediction of a given metric requires the judgment of the decision maker. Factors that will have to be taken into account in such decisions include stability, performance, resolution, and primary assessment class. Standard predictor configurations for both Type I predictors and Type II predictors are described in Part II.

Because the analysis models can vary significantly in their computational time, a progress bar is displayed on each metric card (3, Fig. 11), indicating how many analysis models are still under evaluation. For example, if three analysis models have been deployed to compute the *component damage states* metric, but only two have been completed by the time the decision maker has opened the evaluation page, the progress bar will indicate an incomplete analysis until the remaining third model has completed its run.

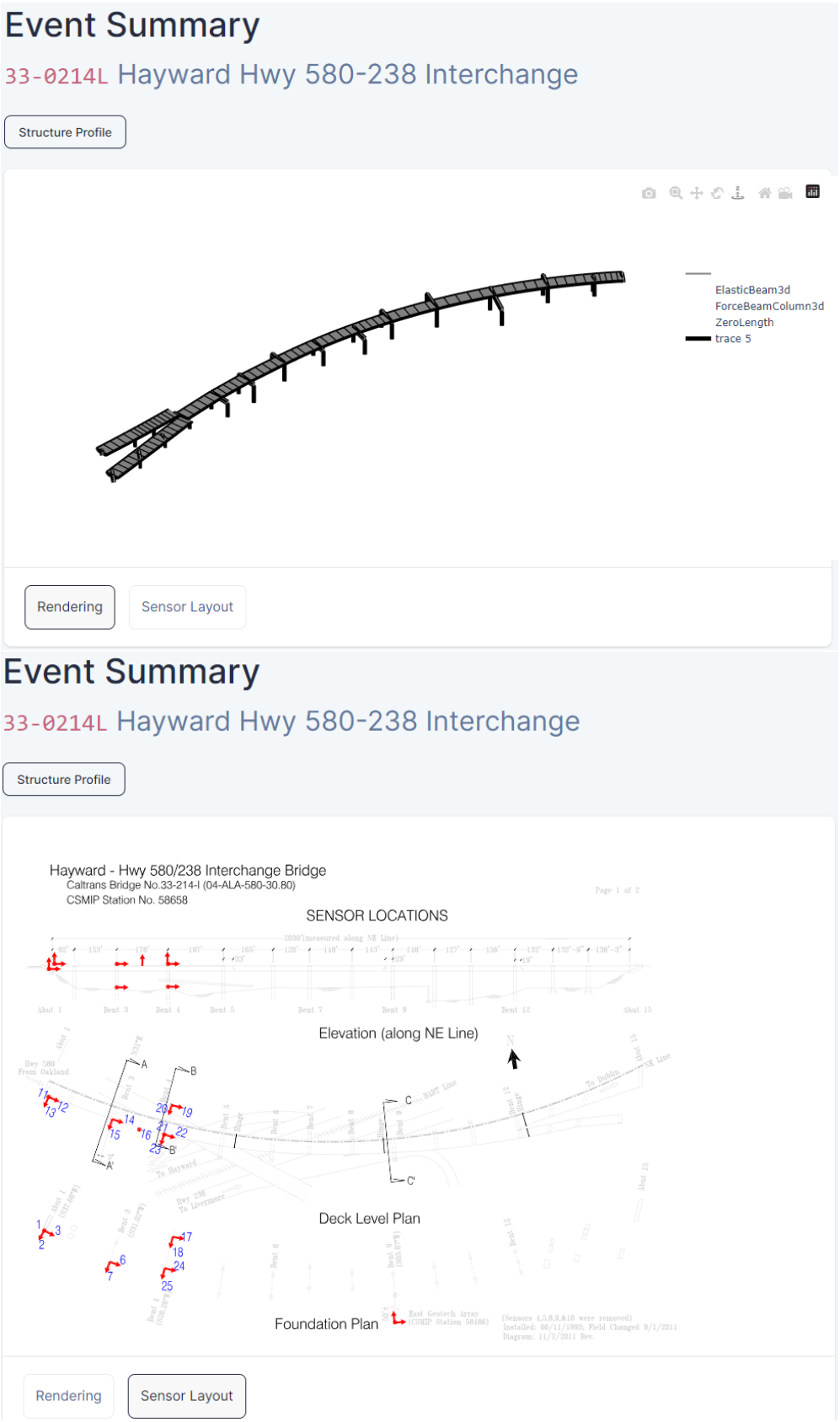


Fig. 10: The Type I predictor rendering and sensor layout views on an evaluation page.

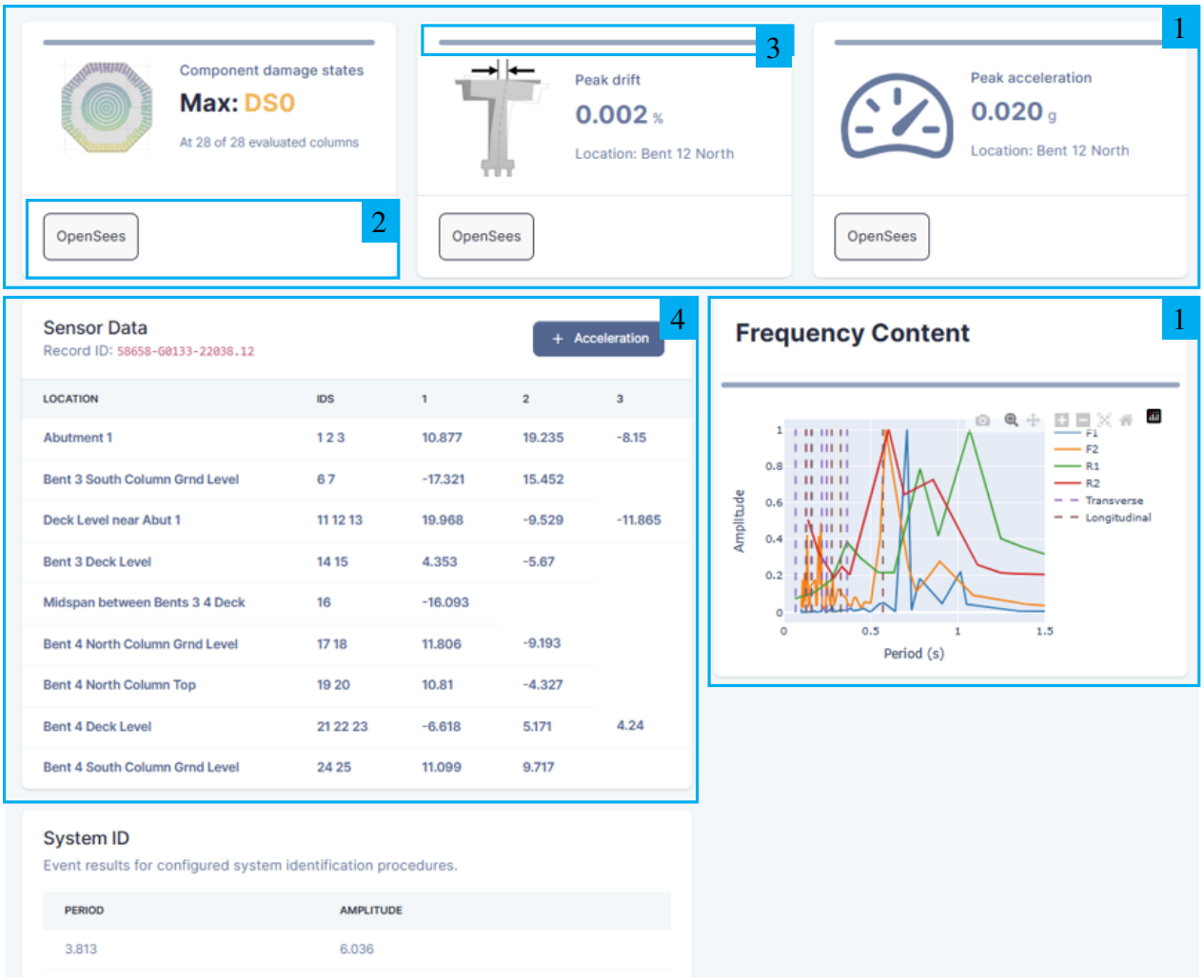


Fig. 11: An evaluation page presenting the metrics obtained for a sample event.

Chapter 3

An Example Scenario

An example platform workflow from the decision maker's point of view is described as follows:

1. An earthquake event occurs.
2. Servers hosted by CGS process any channel data and leverage the secure *BRACE*^{2} API to "post" this data to the *BRACE*^{2} server.
3. When a post is made to the *BRACE*^{2} server (i.e., step 2), an email is sent to those with platform access and various prediction methods are *automatically deployed* to produce the event evaluation reports.
4. The decision maker logs on to the platform and lands on the "Dashboard" (Fig. 4).
5. From the dashboard, the decision maker accesses the evaluation page of the most recently evaluated event by clicking the top row of the "Recently Evaluated Assets" table (4, Fig. 4), where the results of the evaluation have been automatically populated.
6. On the evaluation page, the decision maker checks the health metrics (1, Fig. 11) estimated by the deployed analysis models and period shift estimated by the system identification methods. It is noted that the system identification methods and the period shift estimation (refer to section Section 5.3) directly make use of the sensor data and not the results of the analytical models. The results of individual analysis models appear on the evaluation page in real-time as the analyses are completed, such that the results for the models that take less time to analyze are available as soon as possible.

Part II

Predictors

Part II is targeted at practicing engineers and also researchers. The concept of a predictor is developed in detail, which provides the fundamental abstraction through which structural health monitoring is performed. The predictors that perform bridge response simulation are Type I predictors. They consist of a family of structural analysis models. The general process for constructing a Type I predictor is described in Chapter 4. The predictors that perform data-driven bridge dynamics identification are Type II predictors. They consist of a family of system identification procedures. A practical guide to viewing, configuring, and interpreting the results of Type II predictors is provided in Chapter 5.

Chapter 4

Type I Prediction: Structural Modeling

4.1 Premise

The state of a structure as it experiences an event is assumed to be characterized by its displacement $\mathbf{u}$ and velocity $\dot{\mathbf{u}}$, which is governed by Newton's laws of motion:

$$\mathbf{m}\ddot{\mathbf{u}} = \mathbf{f}^{\text{ext}} - \mathbf{f}^{\text{int}}(\mathbf{u}, \dot{\mathbf{u}}), \tag{4.1}$$

where $\ddot{\mathbf{u}}$ is the acceleration associated with the state of the structure, $\mathbf{f}^{\text{int}}$ characterizes the internal resisting force produced by the structure in response to its configuration, and $\mathbf{f}^{\text{ext}}$ is the external loading that is imposed by the event. Type I predictors are premised on the idea that if the form of $\mathbf{f}^{\text{int}}$ is *assumed* and the loading $\mathbf{f}^{\text{ext}}$ is *given*, then $\mathbf{u}$ and $\dot{\mathbf{u}}$ can be approximately *determined* by methods of integration of the second order system of differential equations in Eq. (4.1). On the health monitoring platform, this is realized by employing a *finite element* approximation of $\mathbf{m}$ and $\mathbf{f}^{\text{int}}$ and constructing an approximation to $\mathbf{f}^{\text{ext}}$ from the sensor data.

In the field of civil engineering, the most ubiquitous class of prediction techniques is those that leverage a structural analysis model. These techniques are readily accessible to the large workforce of civil engineers that routinely develops models of this type for applications in design and rehabilitation. Structural analysis models are constructed by characterizing the properties of each structural component of an asset. Currently, models of this type that are employed on the *BRACE*^{2} platform leverage advanced finite elements that are typically not available through basic commercial products (see, e.g., [85]).

Type I predictors can be divided into *local*, or *component* models, and *global*, or *structure* models. The finite element model depicted in Fig. 12 is an example of one such local model, which can be used for carrying out specialized analysis procedures to determine the overall element properties, or even acting as a super-component in a larger global model. This model, in particular, is used to derive the warping constant that is used by the elastic beam elements employed by the low-fidelity global models (Section 4.2). The remainder of this chapter develops various aspects of finite element modeling, which require special consideration in the context of structural health monitoring, as implemented on the *BRACE*^{2} health monitoring platform. Section 4.4 discusses the formal theoretical structure that is inherent in representing continuous systems by the discrete problem of Eq. (4.1) and Section 4.5 discusses the generalization of this theory to models that exhibit hysteresis.

Fig. 12: Finite element model used to obtain the warping constant for the superstructure of Caltrans Bridge No. 58-0215 Caltrans (Meloland Road Overpass).

4.2 Modeling

Several models are developed for each bridge with varying fidelities to target a wide range of events. In order to accurately represent the structural response to a broad range of excitation intensities, several Type I predictors may be required. For events of low to moderate intensity, a low-fidelity linear model, e.g., Fig. 13, often suffices. However, in order to compute damage indicators following high intensity events, a high-fidelity nonlinear model is required that is able to account for and report on inelastic and geometrically nonlinear response. Because these nonlinear models tend to be computationally intensive, a lower-fidelity model may be utilized to report partial metrics pending the completion of the full nonlinear analysis.

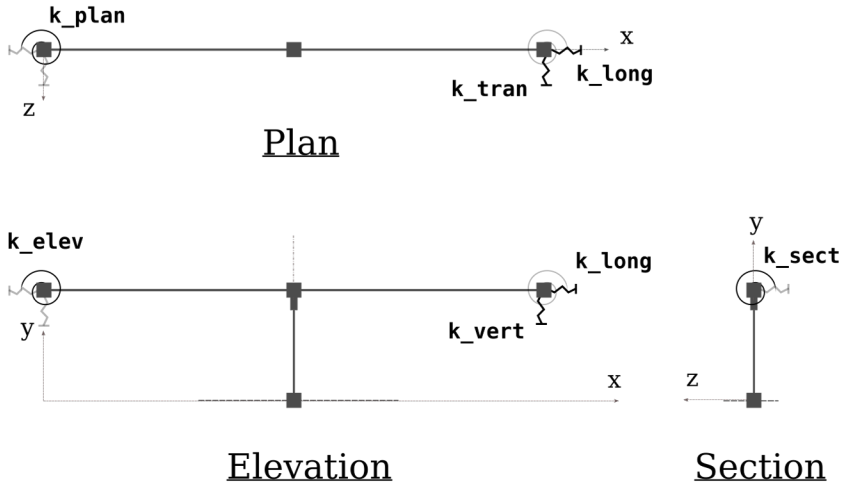


Fig. 13: A low-fidelity model of Caltrans bridge No. 58-0215 (Meloland Road Overpass).

4.2.1 Low Fidelity Models

A low-fidelity model may be used to explore and validate the behavior of an idealized structure in the absence of structural degradation such as cracking and creep. Mass is generally assigned to the structure using consistent mass distribution in all involved frame elements.

Superstructure

In a low-fidelity model, the superstructure is generally modeled using the linear-elastic Euler-Bernoulli beam model. Parameters of this model (i.e., the moment of inertia, cross-sectional area, and elastic modulus) are taken as those of the gross concrete section, neglecting reinforcement.

The girder may be assumed homogeneous and details like reduced-strength concrete specified for the deck need not be accounted for [12]. An independent planar finite element model may be required to calculate the torsion constant of the superstructure girders (see, e.g., Fig. 12). Similarly, cap beams are generally modeled as elastic beam-column elements with moment of inertia incorporating an additional flange where the superstructure coincides with the cap beam.

Substructure

A low-fidelity bent model may similarly employ linear-elastic columns with linear geometry. The elements are joined to the superstructure at the centroid of the girder, and a rigid offset is imposed at this joint with a length equal to the vertical distance between the centroid and the soffit of the girder. For some structures, it may be sufficient to impose fixed displacement boundary conditions at the base of the column, but this decision should be made judiciously. More sophisticated soil/foundation modeling like those described in Section 4.2.2.3 may be required for certain structures even at the low-fidelity level.

Abutments

In simple overpass structures, special care should be considered in modeling the abutment response, even at the low-fidelity level. In a study of Caltrans bridge No. `58-0215` (Meloland Road Overpass), Werner et al. [98] concluded that the abutments and embankments are the primary contributors to the transverse response, meanwhile vertical response characteristics were dominated by properties of the deck. Levine and Scott [47] build on the study of Werner et al. [98] to produce an estimate for linear rotational stiffnesses at the abutments using a system identification procedure developed by Beck [8] . The model used in this study is a simple linear 3D beam assemblage with fixed translational boundary conditions at the abutments. Wilson and Tan [100] used a *single-input, single-output* (SISO) structural identification technique to further explore the abutment response.

Goel and Chopra [26] considered another Caltrans bridge No. `04-0236` (Painter Street Overcrossing) and note that the linear elastic models proposed in the aforementioned publications are insufficient as they fail to account for reductions in the abutment stiffness values that arise under the influence of earthquake motions. This study also evaluated procedures from Caltrans, AASHTO-83, and ATC-6. In another study, Zhang and Makris [103] built upon the work of Wilson and Tan [100], Werner et al. [98], and Goel and Chopra [26]. Finally, Arici and Mosalam [5] reviewed most of the aforementioned studies and sought, among several tasks, to examine the predictive capabilities of identified linear models using a *multi-input, single-output* (MISO) approach.

4.2.2 High-Fidelity Models

A high-fidelity model is able to provide more useful metrics to the health monitoring platform, such as the component damage state card at the top left corner of Fig. 11. However, this requires careful consideration of the inelastic material response throughout the structure. Material properties such as constitutive relationships of the concrete and the reinforcing steel are determined from common engineering practice and/or testing for the materials used in the structure's construction. Generally, these materials include unconfined concrete, confined concrete, steel for reinforcing bars, and elastomers for bearings. They may also include advanced materials such as Fiber Reinforced Polymers

(FRP) using carbon or glass fibers for retrofitted structures.

4.2.2.1 Columns

Because bridge damage states often include flexural yielding (plastic hinging) and failures at the columns, refer to examples in Fig. 14, it is important that a high-fidelity column model is capable of accurately reporting this inelastic behavior. To this end, frame elements that are expected to exhibit inelasticity should be modeled by inelastic elements capable of exhibiting hysteresis. For this purpose, a *fiber section* may be used to collocate material points over the cross-section. Accordingly, uniaxial material models generally suffice for shear-free formulations like the Euler-Bernoulli theory, but multiaxial material response should be considered if shear flexibility is expected. Finite element formulations that apply mixed interpolations tend to furnish an ideal balance between computational expense and accuracy [66].

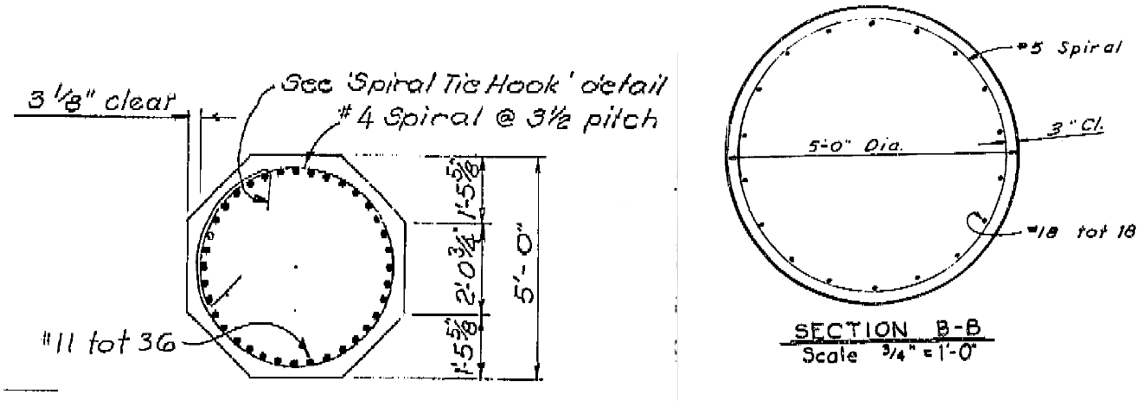


Fig. 14: Representative column details from Caltrans bridge Numbers `04-0236` (Left: Painter St. Overcrossing) and `58-0215` (Right: Meloland Road Overpass).

4.2.2.2 Reinforced Concrete

Inelasticity in flexural members is modeled through the use of nonlinear uniaxial constitutive relations, which are integrated over cross-sectional geometries. Three material behaviors are typically considered in reinforced concrete structures:

1. Core, or "Confined" concrete,
2. Cover, or "Unconfined" concrete, and
3. Reinforcing steel.

Most inelastic models developed for this *BRACE*^{2} project adopt models based on the so-called Algebraic Evolution Equations (AEEs), described by Yassin [101], which employ the Hognestad envelope of Eq. (4.28), discussed in Section 4.5.1.1, for both confined and unconfined concrete, Fig. 15. For reinforcing bars, the steel material typically follows the constitutive relationship of Giuffrè-Menegotto-Pinto (GMP) model, discussed in detail in Chapter 8. The GMP model is based on the basic models Goldberg-Richard, discussed in Section 4.5.1.2. Refer to Section 4.5 for theoretical details for the inelastic modeling of concrete, steel, and elastomers.

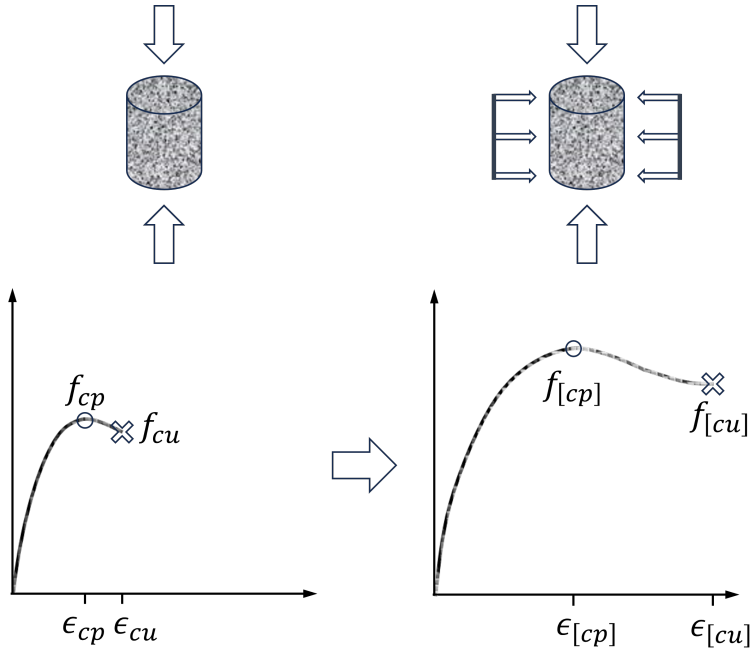


Fig. 15: Modeling confined concrete often requires an empirical transformation of the unconfined concrete behavior.

In-span Hinges

In-span hinges are modeled as rotational releases in the superstructure. The translational behavior of in-span hinges are modeled as nonlinear force-displacement relationships, incorporating the effects of pounding, restrainers, bearings, and shear keys. Varying levels of simplification for the hinge models can be employed to linearize or remove discontinuities in the force-displacement relationships. A detailed discussion of modeling the in-span hinges of the bridge superstructure can be found in Chapter 8.

Abutments

Abutments are modeled as rigid beam elements with ends that are free to rotate, refer to Chapter 8 for details about modeling of the abutments in the transverse, longitudinal, and vertical directions. The translational behaviors of the abutment ends are modeled as nonlinear force-displacement relationships, incorporating the effects of backwalls, wingwalls, soil backfill, and elastomeric bearings. Varying levels of simplification for the abutment models can be employed to linearize or remove discontinuities in the force-displacement relationships.

4.2.2.3 Soil-Shaft Interaction[4]

The finite element models that consider the interaction between soil and shafts of monitored bridges are developed and implemented, aiming to provide one using the platform with detailed understanding of the dynamics between soil, ground movements, and structural response. For each bridge of

[4]For more details, the readers are referred to [59, 60].

interest, before modeling, the geotechnical data at the site are gathered through the instrumentation of boreholes available near the structure and relevant documents. The data includes the distribution of different soil and rock layers. Moreover, some historical data such as the range of groundwater levels are provided by the bridge owner. With the geotechnical data of the site from boreholes and the historical records, the soil-shaft interaction is analyzed with a two-step finite element modeling approach: (1) the site-specific free-field model, and (2) the drilled shaft model.

Free-field modeling The three-dimensional free-field nonlinear soil model calculates the displacements as the applied excitation to the inelastic section models for drilled shafts on the bridge. This is modeled as a column segmented with multiple elements. The constitutive behavior of soil is described by a pressure-independent, multi-yield model with the integration of a range of pertinent soil properties. Then, considering the elastoplastic behavior and the hysteretic damping, the shear velocity (V_s) measurements of soil layers are utilized for deriving the strength parameters of the model. Furthermore, the implementation of the Lysmer-Kuhlemeyer dashpot [49] is achieved by a viscous uniaxial material, where the designed dashpot nodes are connected by zero-length elements. For the models with bidirectional loading (2-D), the soil column is modeled with standard eight-node brick elements. Finally, subjected to the input motion of the earthquake event of interest, the site responses of the models are obtained, where the dynamic displacement histories are specifically utilized as the excitation of the model of the drilled shafts in the second step.

Soil-shaft interaction modeling To accurately capture the shafts' response during seismic events, their sections are modeled with the consideration of inelastic behavior. The nodes of the shafts are established with three dimensions and six degrees of freedom. The interaction between soil and the shaft is modeled using an array of springs arranged in two distinct directions (the vertical excitation is not considered), where the spring nodes are connected by zero-length elements. The constitutive behavior of these springs is specified such that the lateral springs (in the N-S and W-E directions) function as *p-y* springs. The p-y curves reflect the constitutive model of the corresponding soil type, e.g., sandy soil, clay, and weak rock. The key parameters for the hysteretic model of the soil are determined based on the *p-y* curves, obtained following the methodology proposed by Rees et al. [72]. For modeling the inelastic shaft, the fiber section is selected [53], where the Hognestad concrete model of [101] (see Section 4.5.1.1) and a bilinear model are employed for the concrete and reinforcing bars, respectively. Furthermore, a force-interpolated element, which accounts for nonlinear curvature distribution along the element's length [86], is adopted in the modeling. Finally, the displacement at the soil end of each spring, computed from the free-field model at the first step, is applied as a multiple-support excitation.

Integrating soil-shaft interaction models with bridge models Given that the bridge and the corresponding foundation models are decoupled and developed separately by the structural engineering and geotechnical engineering teams, respectively, the final step is to integrate them into one holistic model. In the current implementation of the platform, the soil-shaft interaction models are integrated with the structural model of the bridge itself by connecting the head node of each shaft pile to the bottom node of the corresponding column by enforcing an equal-degree-of-freedom constraint. Such implementation is demonstrated in Chapter 8 with example results shown in Chapter 9.

4.2.3 Pre-processing

Because the sensors are aligned with the structure, but the model degrees of freedom are aligned with global north, south, east, and west directions, the recorded motions and/or the simulated motions must be rotated to align the two. This is performed using the `quakeio` package which is documented in Chapter 14 of Part IV. After performing this pre-processing, the motions at the ground level are used as input to the model.

4.2.4 Limit States

In the limit state analysis, columns are modeled by a single distributed-inelasticity frame element with fiber discretized sections. The node at the base is restrained in all degrees of freedom with the other (top) node connected to the superstructure being free from any constraints. Column components have 4 integration points using the Gauss-Lobatto rule, refer to the schematic on the left part of Fig. 16, while the bent components have 3 integration points in the columns to better match the plastic hinge length in double curvature. Cap beams are modeled as elastic beam elements with cross-section properties computed from the geometry. A rigid joint offset is inserted at the top of each column to the center of the cap beam.

The cross-sections are generally discretized such that the number of circumferential fibers match the number of longitudinal reinforcing bars. In the radial direction, the cover has 2 to 4 fibers and the core has 6 to 10 fibers, depending on the diameter, refer to the cross-section on the right part of Fig. 16.

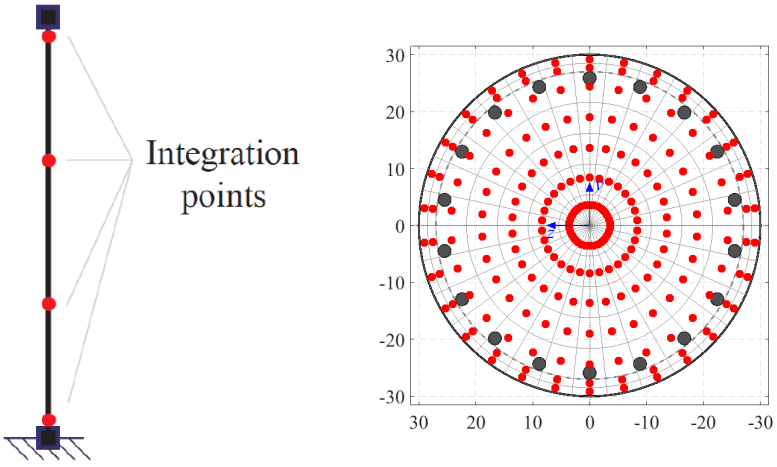


Fig. 16: Strategy of fiber modeling of column members.

To determine the metric of *damage state* (DS) for a column component given an imposed dynamic loading, maximum strain and maximum displacement are computed from NRHA. There are multiple proposed schemes for defining DS, including strain-based DS (Table 2) that models the relationship between fiber strains and component capacity based on previous Caltrans-PEER studies, and ductility-based DS [75] that models the relationship between a displacement-based damage index and component capacity. The health monitoring platform employs strain-based DS.

Table 2: Damage state criteria for strain-based component capacity model.

Damage State	Caltrans-PEER Description	Criteria Used to Define Damage State	Damage State		
			Fiber Location	Compression or Tension	Strain
DS0	No damage	–	–	–	–
DS1	EQ-related tight cracking of cover	Cover cracking: the cover surface reaches tensile strength	Any outermost cover fiber	Tension	ϵ_t
DS2	Moderate cracking (mixed orientations) & minor spalling/flaking	Minor spalling: the cover surface reaches compressive strength	Any outermost cover fiber	Compression	ϵ_{sp}
DS3	Open cracking or major spalling (exterior to confinement)	Major spalling: a significant depth of the cover reaches compressive strength	Any cover fiber at 1/2-3/4 of the cover depth	Compression	ϵ_{sp}
DS4	Exposed core (interior of confinement)	Exposed core: the entire depth of the cover reaches compressive strength	Any innermost cover fiber	Compression	ϵ_{sp}
DS5	Visible bar buckling; confinement loss or core shedding	Core shedding: the outer surface of the core begins to fail	Any outermost core fiber	Compression	ϵ_{cu}
	Multi-bar rupture or buckling; large drift; or core crushing	Bar rupture: longitudinal bars reach ultimate tensile strength	Any longitudinal bar	Tension	ϵ_{su}
DS6	Column collapse (Near-total loss of axial capacity)	Loss of axial capacity: approximately 1/2 of the core fails	All core fibers at 1/4 of the core depth	Compression	ϵ_{cu}

The damage state metric can be used to further extend the decision-making process for operational actions with standard practices by Caltrans' Structures Maintenance and Investigations group, such as those outlined in the "Caltrans First Responder Bridge Assessment Guide". For example, DS1-2 corresponds to allowing the structure to remain open; DS3-4 corresponds to inspecting the structure while allowing it to remain open; DS5 corresponds to closing the bridge for further inspection, and DS6 corresponds to closing the bridge immediately. The *decision boundaries* are to be calibrated in the future as more events occur and are analyzed by the health monitoring platform.

4.3 Interpretation

Following their execution on the platform, Type I predictors results are processed into metrics. The processed metrics are displayed on the metric cards (Fig. 17). As they are results of Type I predictors, they correspond to *simulated* response values. When there are multiple predictors, the desired predictor can be selected from the predictor selection region at the bottom of each metric card, as discussed in Section 2.5.

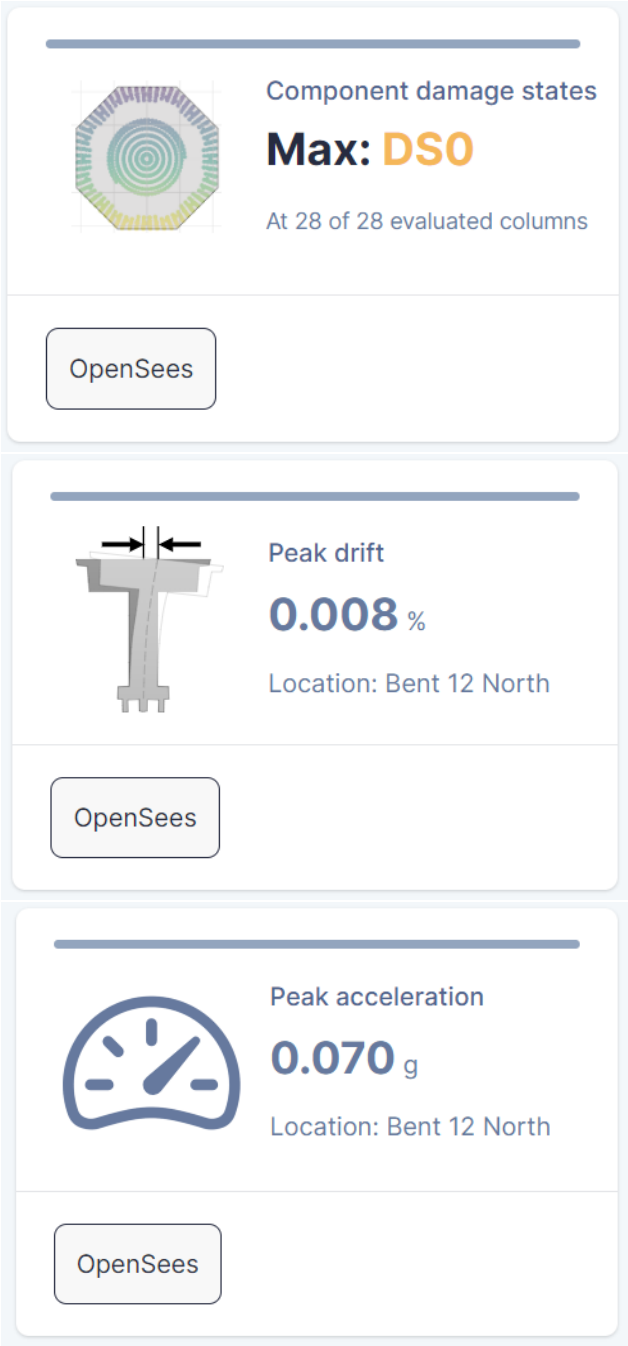


Fig. 17: Type I predictor metrics.

4.4 Appendix 4.A: Discretization

This appendix presents a general approach for discretizing a large class of continuum equilibrium problems that are governed by partial differential equations to formulate a representative discrete internal force vector $\mathbf{f}^{\text{int}}$, as required by the Type I predictors of this chapter. The procedure generalizes the classical finite element method which is typically formulated on Sobolev vector spaces [32] to the setting of nonlinear manifolds[5], which are encountered in the geometrically nonlinear analysis of plates, rods, and shells. In all such problems, equilibrium can be stated in terms of nonlinear partial differential equations of the form:

$$\rho \ddot{\boldsymbol{u}} = \boldsymbol{f} - \mathbb{B}_{\chi} \boldsymbol{\sigma}, \tag{4.2}$$

where $\mathbb{B}_{\chi}$ is a nonlinear differential operator and both the generalized stress $\boldsymbol{\sigma}$ and the body force $\boldsymbol{f}$ are *continuous* functions defined over a body $\mathcal{B}$ placed in space by a map χ. In general $\mathbb{B}$ may depend nonlinearly on χ, but always acts linearly on $\boldsymbol{\sigma}$.

4.4.1 Approximation

The structure outlined in Box 1 implies a weak statement of Eq. (4.2) that is centered around the Galerkin *residual functional*:

$$\mathcal{G}\left[\boldsymbol{\eta}\right] \triangleq \left\langle \boldsymbol{\eta},\, \boldsymbol{f} \right\rangle_{\chi} - \left\langle \boldsymbol{\eta},\, \mathbb{B}_{\chi} \boldsymbol{\sigma} \right\rangle_{\chi}. \tag{4.3}$$

When specialized to classical continuum mechanics, this is equivalent to the classical virtual work functional (before integration by parts):

$$\int \delta \boldsymbol{u} \cdot \operatorname{div} \boldsymbol{\sigma}\, d\Omega + \int \delta \boldsymbol{u} \cdot \boldsymbol{f}\, d\Omega, \tag{4.4}$$

where the virtual displacements $\delta \boldsymbol{u}$ represent the test basis $\boldsymbol{\eta}$ and the equilibrium operator $\mathbb{B}$ is represented by the (negative) divergence $-\operatorname{div}$. Note that the divergence always acts linearly on $\boldsymbol{\sigma}$, but $\boldsymbol{\sigma}$ may depend nonlinearly on the displacement $\boldsymbol{u}$, resulting in a nonlinear problem. The developments that follow are kept applicable to a much wider class of mechanics models by retaining the inner product $\langle \cdot, \cdot \rangle$ and operator $\mathbb{B}$ notations. Note that in the following, the subscript χ on $\mathbb{B}_{\chi}$ and $\langle \cdot, \cdot \rangle_{\chi}$ are dropped for brevity and the body force $\boldsymbol{f}$ is not considered.

[5] A manifold is a mathematical representation of a curved surface. Manifolds naturally arise as solution sets of constrained systems of equations and as graphs of functions.

BOX 1 Continuous Mechanics

Body A body $\mathcal{B}$ is a set of points $\boldsymbol{\xi}$, and a placement $\mathcal{B}_\chi$ of $\mathcal{B}$ is a Riemannian manifold $\mathcal{B}_\chi$ with boundary $\partial\mathcal{B}$ and tangent bundle $T\mathcal{B}_\chi$ embedded in an ambient Euclidean space $\mathcal{E}$. The map $\chi\colon \mathcal{B} \to \mathcal{B}_\chi$ associates the state of a material point $\boldsymbol{\xi}$ to the placement $\mathcal{B}_\chi$. A reference configuration $\chi_0\colon \mathcal{B} \to \mathcal{B}_{\chi_0}$ is often implied and the pull-back by the composition $\chi \circ \chi_0^{-1}$ is denoted by $\boldsymbol{A}$.

Energy The set $\mathscr{C}$ of configurations χ forms a smooth manifold. A configuration χ is identified by a couple $(\boldsymbol{x}, \boldsymbol{\Lambda})$ where $\boldsymbol{x} \in \mathscr{C}_x$ is a vectorial configuration variable and $\boldsymbol{\Lambda} \in \mathscr{C}_\Lambda$ a non-vectorial variable. These enter independently in a Riemannian metric on the tangent space $T_\chi\mathscr{C}$ to $\mathscr{C}$ at χ, which takes the assumed form:

$$\langle \cdot, \cdot \rangle_\chi = \langle \cdot, \cdot \rangle_x + \langle \cdot, \cdot \rangle_\Lambda . \tag{4.5}$$

Stress The spatial stress $\boldsymbol{\sigma}$ in Eq. (4.2) is related to a material stress $\boldsymbol{s}$ which is defined with respect to inner products in $\mathscr{C}$ at the current χ and reference χ_0 configurations:

$$\langle \boldsymbol{c}, \boldsymbol{\sigma} \rangle_\chi = \langle \boldsymbol{A}\boldsymbol{c}, \boldsymbol{s} \rangle_{\chi_0} = \langle \boldsymbol{c}, \boldsymbol{A}_* \boldsymbol{s} \rangle_\chi \tag{4.6}$$

for any $\boldsymbol{c}$. Similarly, a dual $\mathbb{B}^*$ to the equilibrium operator $\mathbb{B}$ in Eq. (4.2) is defined with respect to Eq. (4.5):

$$\left\langle \boldsymbol{\eta}, \mathbb{B}_\chi \boldsymbol{\sigma} \right\rangle_\chi = \left\langle \mathbb{B}^*_\chi \boldsymbol{\eta}, \boldsymbol{\sigma} \right\rangle_\chi . \tag{4.7}$$

Strain The material strain $\boldsymbol{e}$ is compatible with $\mathbb{B}^*_\chi$ through the differential relation:

$$D\boldsymbol{e} \cdot \boldsymbol{u} \equiv \boldsymbol{A}\mathbb{B}^*_\chi \boldsymbol{u}. \tag{4.8}$$

Constitution The constitutive response is defined between the material stress $\boldsymbol{s}$ and the material strain $\boldsymbol{e}$ with material $\boldsymbol{C}_s$ and spatial $\boldsymbol{C}_\sigma$ moduli defined by:

$$D\boldsymbol{s} \cdot \boldsymbol{u} = \boldsymbol{C}_s \, D\boldsymbol{e} \cdot \boldsymbol{u} \qquad \text{and} \qquad \boldsymbol{C}_\sigma \triangleq \boldsymbol{A}_* \boldsymbol{C}_s \boldsymbol{A}. \tag{4.9}$$

In nonlinear finite element analysis, the solution algorithm searches for a configuration χ in the space $\mathscr{C}$ that satisfies the equilibrium relations in Eq. (4.2) by applying a sequence of small perturbations $\{\boldsymbol{u}^{(i)}\}$ to an initial guess $\chi^{(0)}$. Each perturbation $\boldsymbol{u}^{(i)}$ is determined by the linearization of Eq. (4.3) at the last trial configuration $\chi^{(i)}$ of the iterative process. At each iteration, the tangent space $T_{\chi^{(i)}}\mathscr{C}$ contains infinitely many modes for representing $\boldsymbol{u}^{(i)}$ in updating the configuration $\chi^{(i)}$. This appendix develops the general equations for the procedure of updating the trial configuration $\chi^{(i)}$ by first linearizing Eq. (4.3) in Section 4.4.2 and then developing a finite-dimensional representation of the linearized residual in Section 4.4.3.

4.4.2 Linearization

The linearization of the residual functional $\mathcal{G}$ in Eq. (4.3) is akin to a first-order Taylor series expansion given by:

$$\mathscr{L}\mathcal{G}\left[\boldsymbol{\eta}\right] = \mathcal{G}\left[\boldsymbol{\eta}\right] + \nabla_{\boldsymbol{u}}\mathcal{G}\left[\boldsymbol{\eta}\right], \tag{4.10}$$

where $\nabla_{\boldsymbol{u}}\mathcal{G}[\boldsymbol{\eta}]$ is the covariant derivative of the functional $\mathcal{G}$. This is given by the formula for the covariant differentiation of co-vectors:

$$\nabla_{\boldsymbol{u}}\mathcal{G}\left[\boldsymbol{\eta}\right] = D\mathcal{G}\left[\boldsymbol{\eta}\right]\cdot\boldsymbol{u} - \mathcal{G}\left[\nabla_{\boldsymbol{u}}\boldsymbol{\eta}\right], \tag{4.11}$$

where $D\mathcal{G}\left[\boldsymbol{\eta}\right]\cdot\boldsymbol{u}$ is the Gateaux differential of the residual component $\mathcal{G}\left[\boldsymbol{\eta}\right]$ in the direction $\boldsymbol{u} \in T_{\chi}\mathscr{C}$ and $\mathcal{G}\left[\nabla_{\boldsymbol{u}}\boldsymbol{\eta}\right]$ accounts for the nonlinearity of the manifold. The covariant term $\mathcal{G}\left[\nabla_{\boldsymbol{u}}\boldsymbol{\eta}\right]$ is commonly neglected in the literature, which amounts to selecting an implicit connection, and often produces an asymmetric tangent stiffness matrix.

At the i^{th} iteration, the differential is defined in terms of an integral curve $\tau \mapsto \chi_{\tau}$ through $\mathscr{C}$ with the properties:

$$\chi_{\tau}(0) = \chi^{(i)} \in \mathscr{C} \qquad \text{and} \qquad \dot{\chi}_{\tau}(0) = \boldsymbol{u}^{(i)} \in T_{\chi}\mathscr{C}. \tag{4.12}$$

To produce distinct material and geometric tangents, Eq. (4.3) is separated into terms with purely material and geometric quantities. This is achieved by transposing the operators $\mathbb{B}$ and $\boldsymbol{A}_{*}$ with the help of Eq. (4.6) and Eq. (4.7) to shift the non-material terms in Eq. (4.3) to the first argument of the inner product, leaving only the material stress $\boldsymbol{s}$ as the second argument:

$$\mathcal{G}[\boldsymbol{\eta}] = \langle \boldsymbol{\eta},\, \mathbb{B}\boldsymbol{\sigma}\rangle = \left\langle \mathbb{B}^{*}\boldsymbol{\eta},\, \boldsymbol{\sigma}\right\rangle = \left\langle \boldsymbol{A}\mathbb{B}^{*}\boldsymbol{\eta},\, \boldsymbol{s}\right\rangle. \tag{4.13}$$

Linearization now proceeds in the standard fashion with the application of the product rule to the last expression of Eq. (4.13):

$$\begin{aligned} D\mathcal{G}[\boldsymbol{\eta}]\cdot\boldsymbol{u} = &\quad \tfrac{\mathrm{d}}{\mathrm{d}\tau}\mathcal{G}_{\tau}[\boldsymbol{\eta}]\Big|_{\tau=0} \\ = &\quad \left\langle \boldsymbol{A}\mathbb{B}^{*}\boldsymbol{\eta},\, D\boldsymbol{s}\cdot\boldsymbol{u}\right\rangle \;+\; \left\langle D\left[\boldsymbol{A}\mathbb{B}^{*}\boldsymbol{\eta}\right]\cdot\boldsymbol{u},\, \boldsymbol{s}\right\rangle. \\ \triangleq &\quad \mathcal{K}_{M}\left[\boldsymbol{\eta},\boldsymbol{u}\right] \;+\; \mathcal{K}_{G}\left[\boldsymbol{\eta},\boldsymbol{u}\right], \end{aligned} \tag{4.14}$$

where $\mathcal{K}_{M}$ and $\mathcal{K}_{G}$ are referred to as the *material* and *geometric* stiffness *operators* of the problem. The stress derivative $D\boldsymbol{s}\cdot\boldsymbol{u}$ that appears in $\mathcal{K}_{M}$ above is taken fixed in the reference configuration and expanded by the chain rule to give:

$$\begin{aligned} D\boldsymbol{s}(\boldsymbol{e})\cdot\boldsymbol{u} &= \frac{\mathrm{d}}{\mathrm{d}\tau}\,\boldsymbol{s}\circ\boldsymbol{e}\,\Big|_{\tau=0} \\ &= \boldsymbol{C}_{s}\,D\boldsymbol{e}\cdot\boldsymbol{u} \\ &= \boldsymbol{C}_{s}\,\boldsymbol{A}\mathbb{B}^{*}\boldsymbol{u}, \end{aligned} \tag{4.15}$$

where Eq. (4.9) supplies the modulus $\boldsymbol{C}_{s}$ and Eq. (4.8) the strain linearization $\boldsymbol{A}\mathbb{B}^{*}\boldsymbol{u}$. Substitution back into Eq. (4.14) furnishes the final expression for the material stiffness $\mathcal{K}_{M}$ operator:

$$\mathcal{K}_{M}\left[\boldsymbol{\eta},\boldsymbol{u}\right] = \left\langle \boldsymbol{A}\mathbb{B}^{*}\boldsymbol{\eta},\, \boldsymbol{C}_{s}\,\boldsymbol{A}\mathbb{B}^{*}\boldsymbol{u}\right\rangle = \left\langle \mathbb{B}^{*}\boldsymbol{\eta},\, \boldsymbol{C}_{\sigma}\,\mathbb{B}^{*}\boldsymbol{u}\right\rangle, \tag{4.16}$$

where Eq. (4.9) is used to replace $\boldsymbol{C}_s$ by $\boldsymbol{C}_\sigma$ after transferring $\boldsymbol{A}$ in the first inner product argument to the second argument with Eq. (4.6).

The geometric stiffness $\mathcal{K}_G$ is characterized by the directional derivative $D[\boldsymbol{A}\mathbb{B}^*\boldsymbol{\eta}]\cdot\boldsymbol{u}$. Application of the product rule furnishes:

$$\mathcal{K}_G\left[\boldsymbol{\eta},\boldsymbol{u}\right] = \left\langle \partial_{\boldsymbol{u}}\left[\boldsymbol{A}\right]\mathbb{B}^*\boldsymbol{\eta},\, \boldsymbol{s}\right\rangle + \left\langle \boldsymbol{A}\partial_{\boldsymbol{u}}\left[\mathbb{B}^*\boldsymbol{\eta}\right],\, \boldsymbol{s}\right\rangle. \tag{4.17}$$

The first term on the right arises from the push-forward between the material and the spatial representation while the second term captures the dependence of the operator $\mathbb{B}^*$ on the configuration χ. In a small deformation setting where the material and spatial representations are assumed to coincide, both terms will naturally vanish.

4.4.3 Discretization

Because Eq. (4.10) and Eq. (4.14) are linear in both $\boldsymbol{\eta}$ and $\boldsymbol{u}$, a discrete algebraic problem arises directly when the discretized vectors $\boldsymbol{\eta}^h$ and $\boldsymbol{u}^h$ are represented as a linear combination of *simple* basis functions that only depend on the coordinate $\boldsymbol{\xi}$. However, for problems formulated on manifolds, a simple interpolation is often inappropriate. Instead it is often desirable to apply a *configuration-dependent* interpolation, where $\boldsymbol{\eta}^h$ and $\boldsymbol{u}^h$ depend nonlinearly on the nodal values [73, 16, 81]. In this case, discrete expressions for Eq. (4.10) and Eq. (4.14) follow by constructing the curve χ_τ in Eq. (4.12) in terms of discrete nodal parameters $\{\boldsymbol{u}_b\}$ furnishing the discrete tangent space $T_\chi\mathscr{C}^h$ where elements take the form:

$$\boldsymbol{u}^\ell \triangleq \frac{\mathrm{d}}{\mathrm{d}\tau}\chi_\tau\bigg|_{\tau=0} = \sum_b \frac{\mathrm{d}}{\mathrm{d}\tau}\boldsymbol{u}^h(\tau\boldsymbol{u}_b)\bigg|_{\tau=0}.$$

This motivates the definition of *effective interpolation bases* $\{\boldsymbol{\varphi}_{\eta a}\}$ and $\{\boldsymbol{\varphi}_{ub}\}$:

$$\boldsymbol{\varphi}_{\eta a}(\boldsymbol{\xi}) \triangleq \frac{\mathrm{d}}{\mathrm{d}\tau}\boldsymbol{\eta}^h(\boldsymbol{\xi};\tau\boldsymbol{u}_a)\bigg|_{\tau=0} \qquad \text{and} \qquad \boldsymbol{\varphi}_{ub}(\boldsymbol{\xi}) \triangleq \frac{\mathrm{d}}{\mathrm{d}\tau}\boldsymbol{u}^h(\boldsymbol{\xi};\tau\boldsymbol{u}_b)\bigg|_{\tau=0}. \tag{4.18}$$

4.4.3.1 Resisting forces

The evaluation of Eq. (4.3) on the linearized variation $\boldsymbol{\eta}^\ell$ furnishes a representation on the basis $\boldsymbol{\varphi}_{\eta a}$ which follows from the linearity of the inner product:

$$\mathcal{G}\left[\boldsymbol{\eta}^\ell\right] = \left\langle \boldsymbol{\eta}^h,\, \mathbb{B}\boldsymbol{\sigma}\right\rangle = \sum_a \left\langle \boldsymbol{\varphi}_{\eta a}\boldsymbol{\eta}_a,\, \mathbb{B}\boldsymbol{\sigma}\right\rangle. \tag{4.19}$$

Integration by parts is once again expressed by shifting from the operator $\mathbb{B}$ in the second argument to $\mathbb{B}^*$ in the first:

$$\begin{aligned}
\mathcal{G}\left[\boldsymbol{\eta}^\ell\right] &= \sum_a \left\langle \mathbb{B}^*\left[\boldsymbol{\varphi}_{\eta a}\boldsymbol{\eta}_a\right],\, \boldsymbol{\sigma}\right\rangle \\
&= \sum_a \left\langle \boldsymbol{B}_a^{\mathrm{t}}\boldsymbol{\eta}_a,\, \boldsymbol{\sigma}\right\rangle \qquad \text{with} \quad \boldsymbol{B}_a^{\mathrm{t}} \triangleq \mathbb{B}^*\boldsymbol{\varphi}_{\eta a} \\
&= \sum_a \left\langle \boldsymbol{\eta}_a,\, \boldsymbol{B}_a\boldsymbol{\sigma}\right\rangle,
\end{aligned} \tag{4.20}$$

where $\boldsymbol{B}_a^{\mathrm{t}}$ is the linear operator that results from the application of $\mathbb{B}^*$ to $\boldsymbol{\varphi}_{\eta a}$. Finally, recall that for any orthonormal basis $\{\mathbf{e}_k\}$ spanning the degrees of freedom in $T\mathcal{B}_\chi$, an identity is furnished by the summation $\mathbf{e}_k \otimes \mathbf{e}_k$. This is used in Eq. (4.20) to give:

$$\begin{aligned} \mathcal{G}\left[\boldsymbol{\eta}^\ell\right] &= \sum_{a,k} \left\langle (\mathbf{e}_k \otimes \mathbf{e}_k)\, \boldsymbol{\eta}_a,\ \boldsymbol{B}_a \boldsymbol{\sigma} \right\rangle \\ &= \sum_{a,k} \boldsymbol{\eta}_a \cdot \mathbf{e}_k \left\langle \mathbf{e}_k,\ \boldsymbol{B}_a \boldsymbol{\sigma} \right\rangle, \end{aligned} \tag{4.21}$$

where the summation $\langle \mathbf{e}_k,\ \boldsymbol{B}_a \boldsymbol{\sigma} \rangle\, \mathbf{e}_k$ represents the residual element nodal forces $\{\mathbf{f}_a^{\mathrm{int}}\}$.

4.4.3.2 Tangent Stiffness

The material tangent $\mathcal{K}_M\left[\boldsymbol{\eta}^\ell, \boldsymbol{u}^\ell\right]$ depends on $\boldsymbol{u}^\ell$ via the operator $\mathbb{B}^*$. On account of linearity, the operation has the following representation:

$$\mathbb{B}^* \boldsymbol{u}^\ell = \sum_b \boldsymbol{B}_b^\star \boldsymbol{u}_b \qquad \text{with} \qquad \boldsymbol{B}_b^\star \triangleq \mathbb{B}^* \boldsymbol{\varphi}_{ub}. \tag{4.22}$$

Similarly, a discrete geometric operator $\boldsymbol{G}_{ab}[\boldsymbol{s}]$ is defined as the partial derivatives of the product $\boldsymbol{B}_a \boldsymbol{A}_* \boldsymbol{s}$ with respect to the nodal parameters $\boldsymbol{u}_b$, with $\boldsymbol{s}$ held constant:

$$\boldsymbol{G}_{ab}[\boldsymbol{s}] = \partial_b [\boldsymbol{B}_a \boldsymbol{A}_* \mid \boldsymbol{s}], \tag{4.23}$$

where the vertical bar indicates that the terms that follow are held constant under differentiation. In terms of $\boldsymbol{B}_a$, $\boldsymbol{B}_b^\star$, and $\boldsymbol{G}_{ab}$, the Galerkin differential $D\mathcal{G}\left[\boldsymbol{u}^\ell\right] \cdot \boldsymbol{\eta}^\ell$ of Equation (4.14) then has the representation:

$$\mathcal{K}_M\left[\boldsymbol{\eta}^\ell, \boldsymbol{u}^\ell\right] + \mathcal{K}_G\left[\boldsymbol{\eta}^\ell, \boldsymbol{u}^\ell\right] = \sum_{a,b} \boldsymbol{\eta}_a \cdot \mathbf{K}_{ab}\, \boldsymbol{u}_b, \tag{4.24}$$

where the nodal stiffness matrices $\{\mathbf{K}_{ab}\}$ are furnished by summation on i and j:

$$\mathbf{K}_{ab} \triangleq \left\langle \mathbf{e}_i,\ \boldsymbol{B}_a \boldsymbol{C}_\sigma \boldsymbol{B}_b^\star \mathbf{e}_j \right\rangle \mathbf{e}_i \otimes \mathbf{e}_j + \left\langle \mathbf{e}_i,\ \boldsymbol{G}_{ab}[\boldsymbol{s}] \mathbf{e}_j \right\rangle \mathbf{e}_i \otimes \mathbf{e}_j. \tag{4.25}$$

4.4.4 Summary

The results of this appendix are collected in Box 2. The operators $\boldsymbol{B}_a$, $\boldsymbol{B}_b^\star$, and $\boldsymbol{G}_{ab}$ in Box 2 are the essential components that define a finite element and furnish the element internal forces $\mathbf{f}^{\mathrm{int}}$ and stiffness $\mathbf{K}$. If $\boldsymbol{\varphi}_{\eta a}$ and $\boldsymbol{\varphi}_{ub}$ are equivalent, the discrete equilibrium operator $\boldsymbol{B}_a$ will be the transpose of the discrete strain operator $\boldsymbol{B}_a^\star$ and the discrete material stiffness will be symmetric.

In a *Bubnov-Galerkin* formulation, the iterative solution procedure converges to a configuration where the residual of Eq. (4.2) is orthogonal to the chosen modes of perturbation, which may be represented in $\boldsymbol{u}$. This variant is commonly referred to simply as *the Galerkin* method and shares an intimate link with problems that can be derived as energy minimizers.

Alternatively, a *Petrov-Galerkin* formulation arises when variations $\boldsymbol{u}^h$ and $\boldsymbol{\eta}^h$ are permitted distinct representations. This can become preferable when a complicated basis is chosen for $\boldsymbol{u}$, as it can greatly simplify the derivation of a consistent tangent stiffness.

BOX 2 Finite Element Operators

- A Galerkin approximation of Eq. (4.10) is defined by effective interpolation bases $\{\boldsymbol{\varphi}_{\eta a}\}$ and $\{\boldsymbol{\varphi}_{ub}\}$ where:

$$\boldsymbol{\eta}^{\ell}(\boldsymbol{\xi}) = \sum_{a} \boldsymbol{\varphi}_{\eta a}(\boldsymbol{\xi})\boldsymbol{\eta}_a \qquad \text{and} \qquad \boldsymbol{u}^{\ell}(\boldsymbol{\xi}) = \sum_{b} \boldsymbol{\varphi}_{ub}(\boldsymbol{\xi})\boldsymbol{u}_b.$$

- Application to the linearized residual of Eq. (4.10) furnishes *three* characteristic operators:

(i)	$\boldsymbol{B}^{\mathrm{t}}_a$	$= \mathbb{B}^{*}\boldsymbol{\varphi}_{\eta a}$	Discrete equilibrium operator
(ii)	$\boldsymbol{B}^{\star}_b$	$= \mathbb{B}^{*}\boldsymbol{\varphi}_{ub}$	Discrete strain operator
(iii)	$\boldsymbol{G}_{ab}[\boldsymbol{s}]$	$= \partial_b\,[\boldsymbol{B}_a\boldsymbol{A}_{*} \mid \boldsymbol{s}]$	Discrete geometric operator

- These operators determine the *nodal residual forces* and the *tangent stiffness*, which in terms of an orthonormal basis $\{\mathbf{e}_k\}$ spanning the degrees of freedom in $T\mathcal{B}_{\chi}$ are furnished by:

$$\begin{aligned} \mathbf{f}^{\mathrm{int}}_a &= \langle \mathbf{e}_i,\ \boldsymbol{B}_a\boldsymbol{\sigma}\rangle\,\mathbf{e}_i, \\ \mathbf{f}^{\mathrm{ext}}_a &= \big\langle \boldsymbol{\varphi}_{\eta a}\mathbf{e}_k,\ \boldsymbol{f}\big\rangle\,\mathbf{e}_k, \end{aligned} \tag{4.26}$$

and

$$\mathbf{K}_{ab} = \big\langle \mathbf{e}_i,\ \boldsymbol{B}_a\boldsymbol{C}_{\sigma}\boldsymbol{B}^{\star}_b\mathbf{e}_j\big\rangle\,\mathbf{e}_i \otimes \mathbf{e}_j + \big\langle \mathbf{e}_i,\ \boldsymbol{G}_{ab}[\boldsymbol{s}]\mathbf{e}_j\big\rangle\,\mathbf{e}_i \otimes \mathbf{e}_j.$$

4.5 Appendix 4.B: Inelasticity

In order to simulate hysteresis, one must consider a stress response $\boldsymbol{\sigma}(\boldsymbol{\varepsilon})$[6] that is not just a function of $\boldsymbol{\varepsilon}$ and $\dot{\boldsymbol{\varepsilon}}$ at a fixed point in time[7], but additionally depends on the *history* of these quantities. Such a functional relationship is expressed with the notation $\boldsymbol{\sigma}\{\boldsymbol{\varepsilon}, \dot{\boldsymbol{\varepsilon}}\}$ and extends to the nodal internal force vector $\mathbf{f}^{\text{int}}$ as follows:

$$\mathbf{f}_a^{\text{int}}\{u, \dot{u}\} = \langle \mathbf{e}_k,\ \boldsymbol{B}_a \boldsymbol{\sigma}\{\varepsilon, \dot{\varepsilon}\} \rangle\, \mathbf{e}_k. \tag{4.27}$$

The objective of this appendix is to develop the most common approaches that are used to formulate such hysteretic functional relationships for 1D response. These can generally be divided into the following three categories:

Algebraic evolution equations (AEEs) dictate an evolving response in terms of algebraic relations and rules for switching between them. These models can be very intuitive to formulate and implement but do not generalize well to multiaxial response. AEEs are particularly useful for modeling the uniaxial response of steel and concrete.

Ordinary differential equations (ODEs) specify the evolution of a response in terms of a function of the response and its rate. It is often more difficult to incorporate nuanced physical behaviors into these models, such as the Lüders plateau of steel[8], and they are also known to violate certain ideal physical principles. Despite these drawbacks, these models still tend to perform reasonably well. ODE models like those in Section 4.5.2 are particularly useful for modeling elastomeric elements like bearings. ODEs have had more success than AEEs in modeling multiaxial responses.

Differential algebraic equations (DAEs) are formulated in terms of *both* differential and algebraic relationships. These tend to be the most difficult formulations to solve. However, DAEs generalize elegantly to multiaxial material response and generally form the basis for the sophisticated theories of plasticity. DAEs are particularly useful for modeling multiaxial plasticity. DAEs are not discussed further in this report since they are not currently used in the *BRACE*^{2} project.

4.5.1 Algebraic Evolution Equations

For simple models, AEEs tend to be more computationally efficient and conceptually easier to implement than models formulated in terms of ODEs or DAEs. AEEs that aim to capture complex behaviors can quickly become cumbersome to formulate and implement, but generally they do not incur a truncation error like ODEs and DAEs and at their worst only may require an iterative root-finding procedure. Because of the empirical nature of most AEEs, they tend to offer parameters that

[6] In this appendix, constitutive response is discussed in terms of the more familiar spatial notation $(\boldsymbol{\sigma}, \boldsymbol{\varepsilon})$ as opposed to the material notation $(\boldsymbol{s}, \boldsymbol{e})$ from Section 4.4. Note that in the context of small deformations the material and spatial representations approximately coincide and one has $\boldsymbol{\varepsilon} \approx \boldsymbol{e}$ and $\boldsymbol{\sigma}(\boldsymbol{\varepsilon}) \approx \boldsymbol{s}(\boldsymbol{\varepsilon})$.

[7] A superposed dot indicates a time derivative.

[8] Lüders plateau is observed on the stress-strain curve just after the elastic regime. Moreover, Lüders banding, a material instability, is reported to be associated with unpinning of dislocations from nitrogen and carbon atmospheres. The existence of Lüders plateau on the stress-strain curves may influence the bending behavior of steel structures and their ductile fracture response.

are more intuitive to practitioners and easier to identify from experimental data. An AEE model is typically comprised of three components:

Envelope is specified as a functional relation between σ and ε which describes the monotonic behavior of the model.

Unloading is specified as a set of rules which describe the hysteretic behavior of the model.

Degradation is a rule for changing the shape of the envelope as energy is dissipated in the response.

An advantage of the AEE models is that they are generally parameterized by characteristic points on an envelope curve, which is associated with simple monotonic tension and compression tests. Some standard envelopes are presented in the remainder of this section.

4.5.1.1 Concrete

Concrete envelope curves are formulated in terms of the following parameters (see also Fig. 15):

f_{cp}	peak compressive stress
ε_{cp}	peak compressive strain
f_{cu}	compressive crushing strength
ε_{cu}	strain at crushing strength
λ	ratio between unloading slope at ε_{cu} and initial slope

Hognestad Hognestad's (Hognestad [30], Kent and Park [44], Scott, Park, and Priestley [77]) monotonic compression envelope is simply the piecewise union of an initially parabolic response, followed by a linear loss of strength:

$$\sigma(\varepsilon) = \begin{cases} Kf_{cp}\left[2\left(\dfrac{\varepsilon}{\varepsilon_{cp}}\right) - \left(\dfrac{\varepsilon}{\varepsilon_{cp}}\right)^2\right], & \varepsilon \le \varepsilon_{cp} \\ Kf_{cp}\left[1 - Z\left(\varepsilon - \varepsilon_{cp}\right)\right], & \varepsilon_{cp} < \varepsilon \le \varepsilon_{cu} \\ 0.2Kf_{cu}, & \varepsilon > \varepsilon_{cu} \end{cases} \tag{4.28}$$

where K (a scaling for strength) and Z (strain softening slope) are model parameters. A hysteretic loading-reloading algorithm for this curve was proposed by Yassin [101], which has been implemented in the widely used `Concrete02` material model of OpenSees [56].

Popovics The curve used by [70] is commonly known as the Mander curve and is given by the nonlinear expression:

$$\sigma_c(\varepsilon) = f_{cp}\frac{(\varepsilon/\varepsilon_{cp})r}{r - 1 + (\varepsilon/\varepsilon_{cp})^r}, \tag{4.29}$$

where r is a parameter often computed from:

$$r = \frac{E_c}{E_c - \left(f_{cp}/\epsilon_{cp}\right)}, \tag{4.30}$$

where E_c is the Young's modulus of the concrete material.

4.5.1.2 Steel

Ramberg-Osgood A popular envelope model for metalic materials was developed by [71] who proposed the following equation:

$$\varepsilon = \frac{\sigma}{E} + a\left(\frac{\sigma}{\sigma_y}\right)^n, \tag{4.31}$$

where E is the elastic modulus, σ_y is the yield stress, assumed to be at the 0.2% offset strain, n is a parameter controlling the transition to yield, and a is a yield offset parameter, typically taken as 0.002. Note that the stress σ is defined implicitly in terms of strain ε in this model and must be solved for numerically.

Goldberg-Richard [27] proposed a curve which furnishes the stress explicitly in terms of strain, as expressed below:

$$\bar{\sigma}(\bar{\varepsilon}) = \left[b\bar{\varepsilon} + \frac{(1-b)\bar{\varepsilon}}{(1+|\bar{\varepsilon}|^r)^{\frac{1}{r}}} \right], \tag{4.32}$$

where $\bar{\sigma} = \sigma/\sigma_y$, $\bar{\varepsilon} = \varepsilon/\varepsilon_y$, $(\sigma_y, \varepsilon_y)$ is the yield point on the curve, b is the strain hardening parameter, and the parameter r influences the shape of the transition curve and takes account of the Bauschinger effect[9]. A hysteretic loading-reloading algorithm for this curve was proposed by [25], which was extended by [22] to include isotropic hardening.

4.5.2 Ordinary Differential Equations

Ordinary differential equations offer a powerful framework for formulating hysteretic models. For example, one-dimensional perfect plasticity can be formulated as the following ODE:

$$\begin{aligned} \dot{\bar{\sigma}} &= \phi(\bar{\sigma}, \dot{\varepsilon})\, \dot{\varepsilon} \\ \phi(\bar{\sigma}, \dot{\varepsilon}) &= \begin{cases} E & \text{if } |\bar{\sigma}| < 1 \quad \text{or} \quad \bar{\sigma}\,\mathrm{sgn}(\dot{\varepsilon}) \le 0 \\ bE & \text{if } |\bar{\sigma}| \ge 1 \quad \text{and} \quad \bar{\sigma}\,\mathrm{sgn}(\dot{\varepsilon}) > 0 \end{cases} \end{aligned} \tag{4.33}$$

where $\bar{\sigma} = \sigma/\sigma_y$, σ_y is the yield stress, E is the elastic modulus, b is the strain hardening parameter, sgn is the signum function[10]. An important family of hysteretic models (with various forms of the function ϕ) that generalizes this formulation are known as Bouc-Wen models and are generally given by a relation of the following form:

$$\begin{aligned} \sigma(\varepsilon) &= bE\varepsilon + (1-b)Ez(\varepsilon), \\ \dot{z} &= \frac{\partial z}{\partial \varepsilon}\frac{\partial \varepsilon}{\partial t} = \phi(z, \varepsilon, \dot{\varepsilon})\, \dot{\varepsilon}. \end{aligned} \tag{4.34}$$

[9] A property of materials where the material's stress/strain characteristics change as a result of the microscopic stress distribution of the material. For example, an increase in tensile yield strength occurs at the expense of the compressive yield strength in steel material.

[10] The signum function of a real number x is a piecewise function which is defined as follows:

$$\mathrm{sgn}(x) \triangleq \begin{cases} -1 & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ 1 & \text{if } x > 0. \end{cases}$$

Bouc's Hysteresis (1967) The original model presented by [10] can be given in the form of Eq. (4.34) with ϕ given by:

$$\phi(z, \dot{\varepsilon}) = A - \alpha \operatorname{sgn}(\dot{\varepsilon})\, z, \tag{4.35}$$

where A scales the hysteretic loop and α controls the shape of the hysteresis loop. Bouc's model yields a separable initial-value problem for $\alpha \neq 0$ with solutions:

$$z(\varepsilon) = \begin{cases} -\frac{1}{\alpha}\left(C_1 e^{-\alpha\varepsilon} - A\right), & \text{if } \dot{\varepsilon} > 0 \\ \frac{1}{\alpha}\left(C_2 e^{\alpha\varepsilon} - A\right), & \text{if } \dot{\varepsilon} < 0 \end{cases} \tag{4.36}$$

where C_1 and C_2 must be calculated piecewise by using the final state of the previous branch as the initial condition.

Wen's Hysteresis (1976) Wen [97] extended the original model by Bouc to control the sharpness of the hysteresis in transition from elastic to inelastic region:

$$\phi(z, \dot{\varepsilon}) = A - (\alpha \operatorname{sgn}(z\dot{\varepsilon}) - \beta)\, |z|^n, \tag{4.37}$$

where n is a parameter that controls the sharpness.

Baber-Wen Degradation (1981) Baber and Wen (1981) [7] further extended the model to include strength and stiffness deterioration:

$$\begin{aligned} \phi(z, \dot{\varepsilon}) &= \frac{1}{\eta}\left(A - \nu|z|^n \left(\alpha \operatorname{sgn}(\dot{\varepsilon} z) + \beta\right)\right), \\ A &= A_o - \delta_A \psi, \\ \nu &= 1 + \delta_\nu \psi, \\ \eta &= 1 + \delta_\eta \psi, \\ \dot{\psi} &= (1 - b) E \dot{\varepsilon} z, \end{aligned} \tag{4.38}$$

where A_o, δ_A, δ_ν, and δ_η are degradation parameters and ψ is the absorbed hysteretic energy, defined as follows:

$$\psi = (1 - b)\omega^2 \int_0^t \dot{\varepsilon}(\tau) z(\tau) \mathrm{d}\tau, \tag{4.39}$$

where ρ is the material density and $\omega^2 := E/\rho$ is the squared pseudo-natural frequency of the nonlinear system.

4.5.3 Implementations

When such models are employed within a finite element simulation, the governing ODE for the internal variables must be integrated numerically. For example, applying the implicit *Backward-Euler* discretization to the variables $\dot{\psi}$ and $\dot{z}$ from Bouc-Wen-Baber (1981) [7] furnishes the following implicit problem for z_{n+1} and ψ_{n+1}:

$$
\begin{aligned}
z_{n+1} &= z_n + \Delta t \frac{A_{n+1} - |z_{n+1}|^n \left\{\gamma + \beta \operatorname{sgn}\left(\frac{\epsilon_{n+1}-\epsilon_n}{\Delta t} z_{n+1}\right)\right\} \nu_{n+1}}{\eta_{n+1}} \frac{\epsilon_{n+1} - \epsilon_n}{\Delta t}, \\
\psi_{n+1} &= \psi_n + \Delta t (1-\alpha) k_o \frac{\epsilon_{n+1} - \epsilon_n}{\Delta t} z_{n+1}, \\
A_{n+1} &= A_o - \delta_A \psi_{n+1}, \\
\nu_{n+1} &= 1 + \delta_\nu \psi_{n+1}, \\
\eta_{n+1} &= 1 + \delta_\eta \psi_{n+1},
\end{aligned}
$$

which is generally solved with Newton-Raphson iterations.

Chapter 5

Type II Prediction: System Identification

5.1 Premise

The state of a structure as it experiences an event is assumed to be characterized by its displacement $\mathbf{u}$ and velocity $\dot{\mathbf{u}}$. It is governed by Newton's laws of motion (same as Eq. (4.1) but repeated here for convenience):

$$\mathbf{m}\ddot{\mathbf{u}} = \mathbf{f}^{\text{ext}} - \mathbf{f}^{\text{int}}(\mathbf{u}, \dot{\mathbf{u}}),$$

where $\ddot{\mathbf{u}}$ is the acceleration associated with the state of the structure, $\mathbf{f}^{\text{int}}$ characterizes the internal resisting force produced by the structure in response to its configuration, and $\mathbf{f}^{\text{ext}}$ is the external loading that is imposed by the event. Type II predictors are premised on the idea that if general constraints on $\mathbf{m}$ and $\mathbf{f}^{\text{int}}$ are *assumed* and $\mathbf{f}^{\text{ext}}$ and $\ddot{\mathbf{u}}$ are *given*, then the function $\mathbf{f}^{\text{int}}$ can be approximately *determined* by algebra. On the health monitoring platform, this is realized by prescribing the domain and range of $\mathbf{f}^{\text{int}}$ and then using *system identification* methods to fit $\mathbf{f}^{\text{int}}$ to directly observed loading and acceleration data.

5.2 Modeling

Table 3 presents the methods available to use as system identification predictors. There are two main categories of methods: state-space models and input/output models. State-space models (see Section 13.1) use matrix methods to estimate a set of differential equations relating input, state, and output. Often, a stabilization diagram (Fig. 18) is used to interpret the results of state-space models. Identified periods and damping ratios are those that are consistent across model orders. Input/output models (see Section 13.6), such as transfer functions, use complex analysis to estimate a direct relationship between the input and output. Often, a frequency response function (Fig. 19) is used to interpret the results of input/output models. Identified periods are those that are associated with peaks in the amplitude of the transfer function.

Generally, state-space models are more suitable for systems with multiple inputs and outputs, while input/output models are more suitable for single-input, single-output systems. Out of the state-space models, SRIM is generally preferable when little to no correlation between future inputs and the current state is expected. OKID-ERA is generally preferable when it is expected that the system's

Table 3: Methods of system IDentification (ID).

Name	Abbreviation	Classification
Observer Kalman ID-Eigensystem Realization Algorithm	OKID-ERA	State-Space
System Realization using Information Matrix	SRIM	State-Space
Response Spectrum Transfer Function	RSTF	Input/Output
Fourier amplitude Spectrum Transfer Function	FSTF	Input/Output
Power Spectrum Transfer Function	PSTF	Input/Output

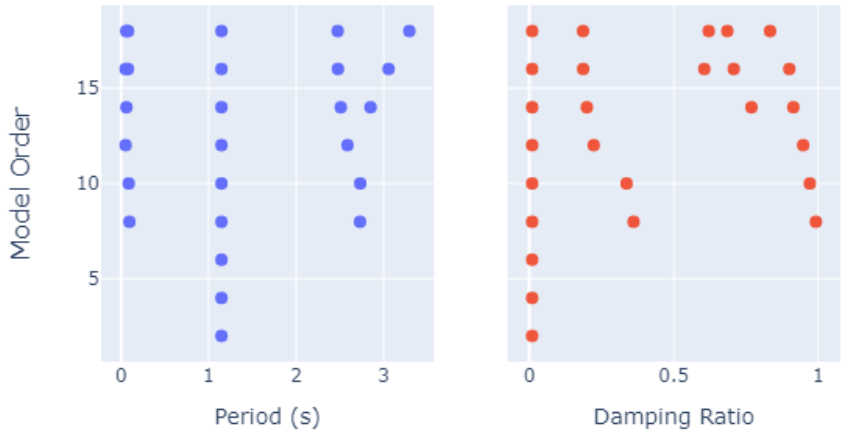


Fig. 18: Stabilization diagram, commonly used to interpret state-space model results.

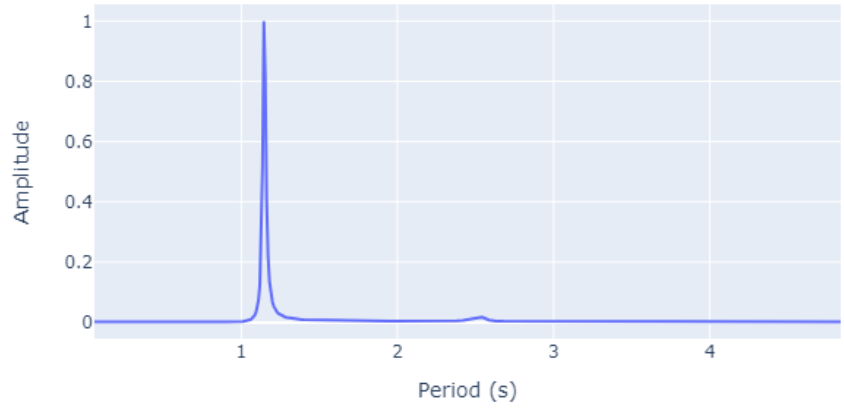


Fig. 19: Frequency response function, commonly used to interpret input/output model results.

impulse response can be estimated from the forced response. Although all input/output models give similar results, the response spectrum transfer function is the most familiar to practicing engineers.

5.2.1 Configuration Interface

The following notes and figures describe the workflow for interested users to build a system identification predictor on the *BRACE*^{2} platform.

- From an asset's Structure Profile, select the Configure Predictors button (1, Fig. 20).
- Select, New System ID (2, Fig. 21).
- Complete the Predictor Builder form to configure a system identification predictor (Fig. 22):
  - Provide a name for the predictor (3).
  - Select Method (methods are discussed broadly in Section 5.2.2 and in depth in Section 13.1) (4).
  - Specify decimation (downsampling) factor (5).
  - Specify method-specific model parameters (6).
  - Select Channels. Define input and output signal identification numbers, as defined in the Structure Profile, e.g., Fig. 20, (7).
  - Select Submit (8).
- Once the form is completed, the system identification predictor is populated on the Predictors page (9, Fig. 21).

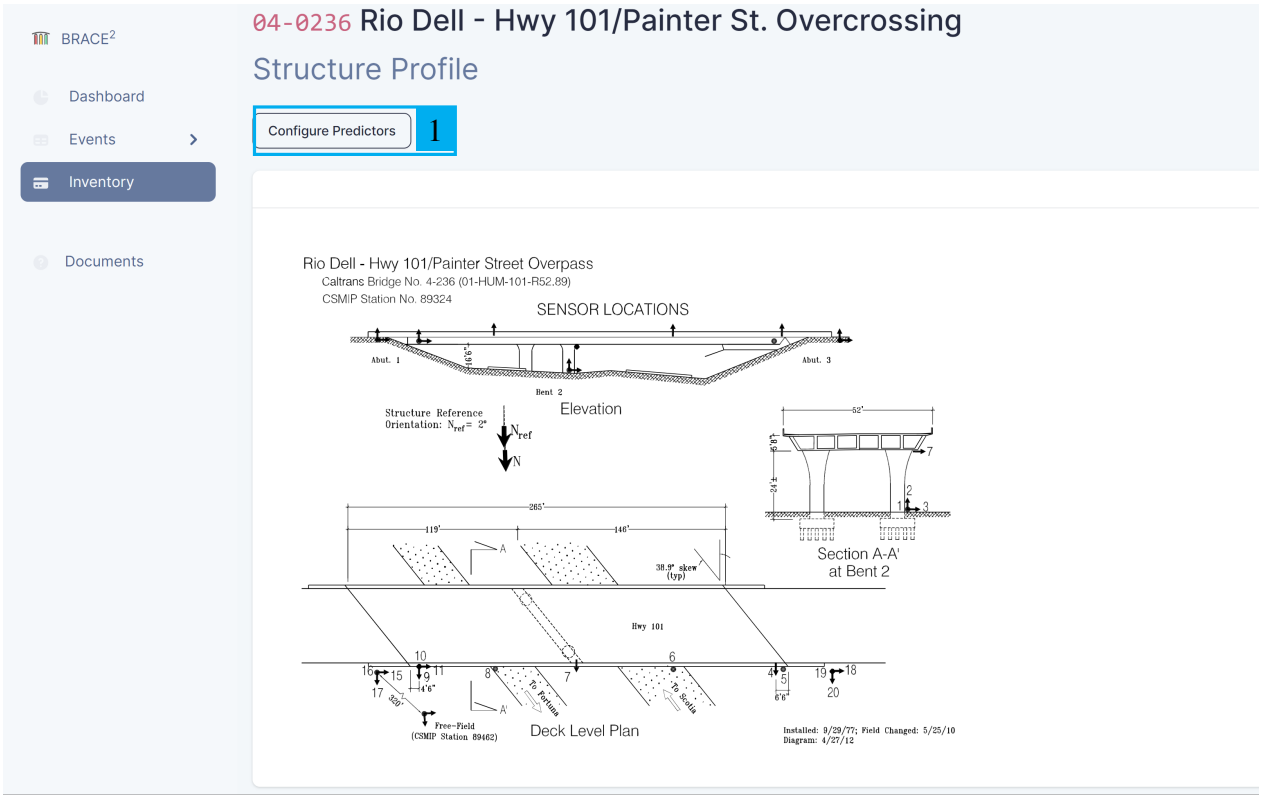


Fig. 20: An example asset's Structure Profile.

For the structure in Fig. 20, an example system identification predictor is built in Fig. 22. The method is SRIM with a decimation factor of 1, a model order of 8, a prediction horizon of 100, input channel 3 (aligned with the North-South axis and located at the base of the North column of the shown skewed bent), and output channels 9, 7, and 4 (in the transverse direction at the superstructure from East to middle to West).

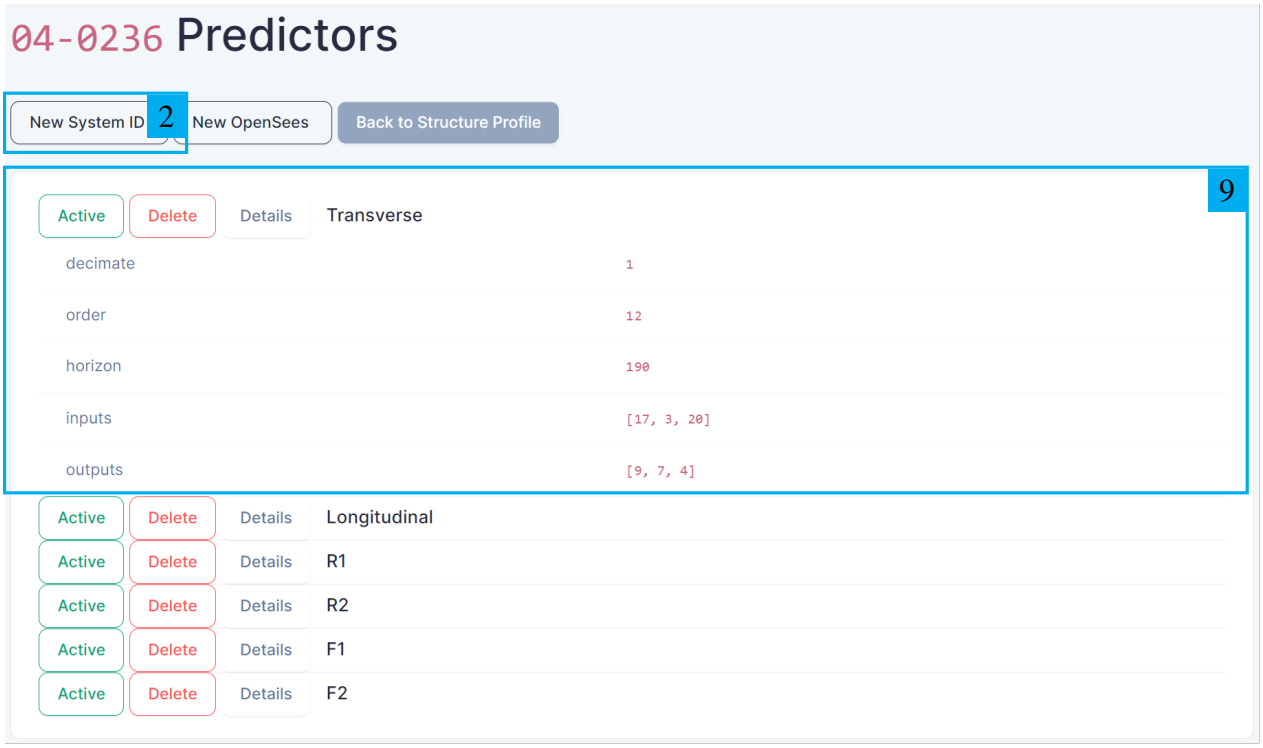


Fig. 21: An example asset's Predictors page.

5.2.2 Configuration Options

Table 4 presents the options available for each system identification predictor, including a top-level view of their interpretation. For most options, the default value provided by the configuration form is generally acceptable, but the input and output channels must be selected based on the sensor locations, identified by their identification number in the Structure Profile. In addition, adjustment of the model order should be considered. Model order is an option only applicable for state-space models. As the model order increases, the number of considered natural modes of vibration increases. Accordingly, model complexity and computational time can also increase. Therefore, it is recommended not to over-define the model by setting the model order too high. For example, for an ordinary standard reinforced concrete bridge with a two-span superstructure supported on a single column bent, a model order between 4 and 20 is generally acceptable and expected to capture the significantly contributing modes of the structural system (see Section 13.1.2 for a discussion on optimal model order).

To give an idea of the effect of system identification model parameters on computational demand, the computational complexity[11] of each method with respect to the number of the input channels, n_i, the number of the output channels, n_o, the prediction horizon, n_h, and the number of timesteps, n_t, is shown in Table 5. The table also lists the limiting computation of each algorithm, which is the step of the algorithm that requires the highest magnitude of computational demand and thus governs the

[11]Computational complexity is expressed using the Big O notation. It is a fundamental mathematical concept to describe the limiting behavior of a function when the argument tends towards a particular value or infinity. For an algorithm, it measures the time complexity and storage space complexity by measuring how its run time or space requirements grow as the input size grows.

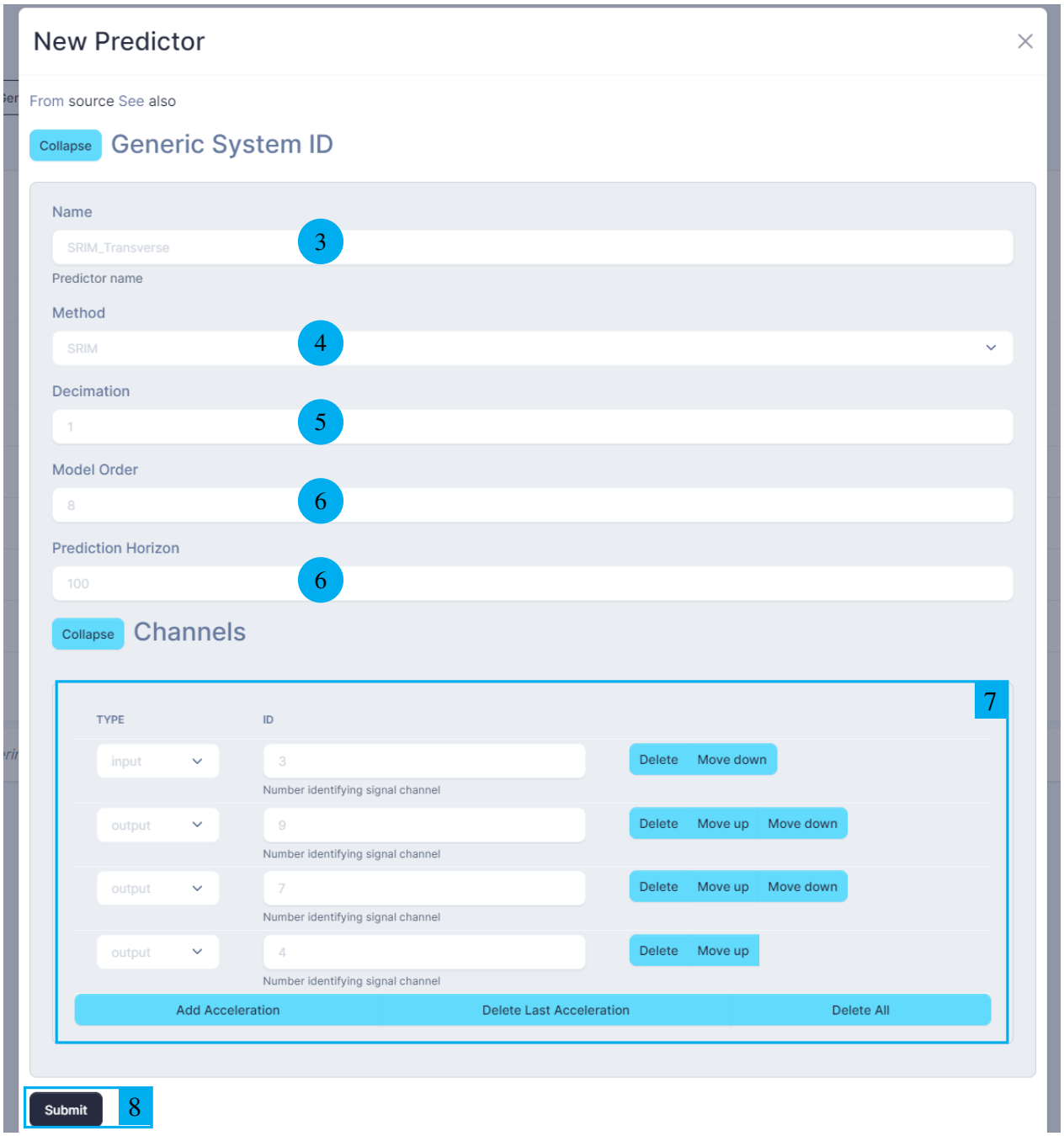


Fig. 22: Predictor Builder form for configuring a generic system identification predictor.

computational complexity. In general, state-space models (OKID-ERA & SRIM in *BRACE*^{2}) have higher computational complexities, but are useful for multi-input, multi-output systems (complexity naturally scales up with the number of inputs and outputs). Input-output models (FSTF, PSTF & RSTF in *BRACE*^{2}) are only used for single-input, single-output systems on the *BRACE*^{2} platform.

Table 4: Options available for *BRACE*^{2} platform to configure Type II predictors.

Model	Option	Symbol	Top-Level Interpretation
All	Input Channel	`inputs`	Input channel number
All	Output Channel	`outputs`	Output channel number
All	Decimation	`decimate`	Downsampling factor
OKID-ERA, SRIM	Model Order	`order`	2× # contributing modes
SRIM	Prediction Horizon	`horizon`	# considered timesteps
FSTF	Period Band	`period_band`	Considered period range
RSTF	Damping	`damping`	Damping ratio

Table 5: Computational complexities of system identification methods (n_i = # inputs, n_o = # outputs, n_h = prediction horizon, n_t = # timesteps).

Method	Complexity	Limiting computation
OKID-ERA	$O\left((n_h \times n_o) \times (n_h \times n_i)^2\right)$	pseudo-inverse
SRIM	$O\left((n_h \times n_o)^3\right)$	singular value decomposition
FSTF, PSTF	$O\left(n_t \times \log_2 n_t\right)$	Fast Fourier Transform
RSTF	$O\left(n_t^2\right)$	numerical integration

5.3 Interpretation

An example set of system identification prediction results is shown in Figs. 23 and 24, mainly for illustrations only and to demonstrate the capabilities available in the *BRACE*^{2} platform with respect to Type II predictors. Therefore, no attempt was made in this example to make the results of the different predictors match by calibrating the default parameters of the different predictors. In the frequency content plot (Fig. 23), the results are shown for all Type II predictors at once; to isolate the view of one predictor at a time, the user can select the predictor from the legend. The horizontal axis corresponds to the different periods of vibration and the vertical axis corresponds to their corresponding spectral amplitude. In this example plot, the predictors named "Transverse" and "Longitudinal" are state space, i.e., time-domain method, predictors (Section 13.1) and their configured channels correspond to transverse and longitudinal recorded motions, respectively. The spectral amplitudes are not tracked for the *BRACE*^{2} implementations of the state space models. Therefore, their plot traces are vertical (dashed) lines at the identified fundamental periods. The predictors corresponding to frequency domain methods are named "R1" and "R2" for the response spectrum predictors with two different channel configurations, and "F1" and "F2" for the Fourier spectrum, also with two different channel configurations. The full spectra showing the relationship between period and amplitude are shown for these frequency domain predictors. The tabulated results in Fig. 24 are currently shown for the predictor named "Transverse"; other predictors can be selected from the listed options at the bottom to view their results. Each row of the prediction results table represents one computed mode and includes estimated natural period, natural frequency, damping ratio, and mode shape. The prediction results also include the modal indicators, Extended Modal Amplitude Coherence (EMAC) and Modal Phase Collinearity (MPC), which are measures of trustworthiness of modal parameter prediction (low values of EMAC and MPC correspond to

identified modes that are not trustworthy) and are discussed further in Section 13.1.2. The modes are listed in order of expected contribution.

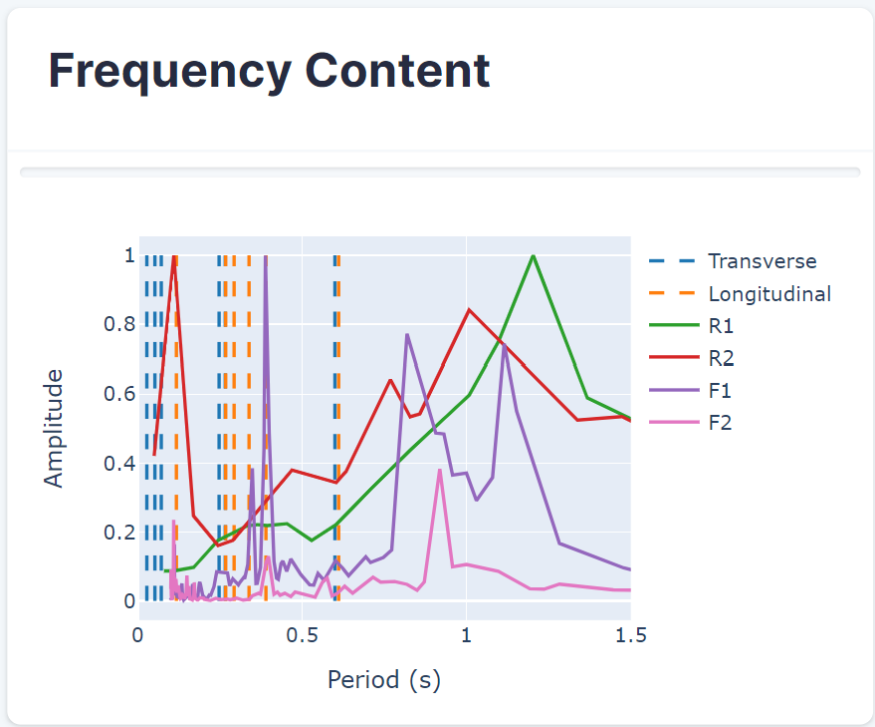


Fig. 23: Example system identification prediction frequency content plot.

System ID

Event results for configured system identification procedures.

PERIOD	FREQUENCY	DAMPING	SHAPE	EMAC	MPC
0.600	1.667	0.046	[0.039, 0.021, 0.05, 0.041]	1.000	0.985
0.266	3.752	0.053	[0.125, 0.067, 0.045, 0.022]	1.000	0.966
0.248	4.038	0.071	[0.145, 0.08, 0.055, 0.037]	1.000	0.934
0.072	13.928	0.003	[0.073, 0.004, 0.013, 0.01]	1.000	0.960
0.052	19.097	0.007	[0.036, 0.002, 0.006, 0.01]	1.000	0.743
0.028	35.899	0.001	[0.053, 0.001, 0.001, 0.001]	1.000	0.999

Transverse | Longitudinal | R1 | R2 | F1 | F2

Fig. 24: Example system identification prediction results table.

Part III

Pilot Studies

Part III documents the pilot studies that have been performed as part of Phase I of the *BRACE*^{2} project. This material will be a valuable resource for engineers that may be tasked with developing new digital twins to be added to the platform. These studies consist of 21 *partial* digital twins that are documented in Chapter 6 and one *complete* digital twin centered around Caltrans bridge No. `33-0214L` (route 580/238 separation structure, or Hayward Bridge, for short) that is documented in Chapters 7 to 9. Specifically, Chapter 7 describes in depth the physical properties of the bridge, Chapter 8 gives a detailed account of the high-fidelity pilot model that was developed for the complete digital twin, and Chapter 9 documents an example simulation.

Chapter 6

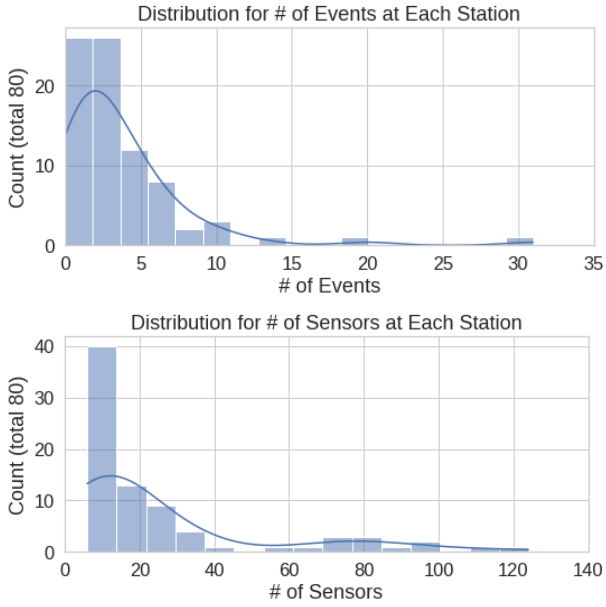
Pilot Assets

The first phase of the *BRACE*^{2} project was originally tasked to conduct a pilot study of five bridges (the last five listed in Table 6), with particular focus on the *route 580/238 separation structure* (Caltrans bridge No. 33-0214L). This study has since been extended to a total of **22** bridges (Table 6), 21 "partial digital twins" and 1 "full digital twin".

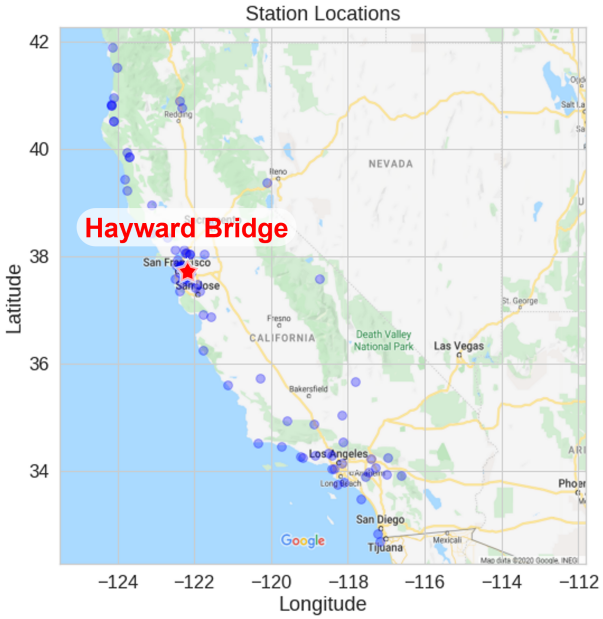
Table 6: Bridges included in Phase I of *BRACE*^{2} platform as a pilot study.

CalID	CESMD	Name	Predictors	Events
56-0586G	CE13705	Corona - I15/Hwy91 Interchange Bridge	6	3
55-0225	CE13795	Capistrano Beach - I5/Via Calif. Bridge	3	5
53-1471	CE14406	Los Angeles - Vincent Thomas Bridge	4	7
53-2795F	CE24694	Sylmar - I5/14 Interchange Bridge	2	2
53-2791	CE24704	Los Angeles - I10/La Cienega Bridge	3	5
53-1794	CE24706	Palmdale - Hwy 14/Barrel Springs Bridge	3	6
50-0271	CE24775	Grapevine - I5/Lebec Rd Bridge	3	7
50-0340	CE33742	Ridgecrest - Hwy 395/Brown Road Bridge	3	5
43-0031E	CE47315	San Juan Bautista - Hwy 101/156 Overpass	3	10
23-0015R	CE68184	Vallejo - Carquinez/I80 East Bridge	2	3
28-0352L	CE68185	Vallejo - Carquinez/I80 West Bridge	5	5
10-0299	CE79421	Leggett - Hwy 101/Confusion Hill Bridge	6	5
04-0228	CE89686	Eureka - Samoa Channel Bridge	4	18
04-0170	CE89708	Arcata - Hwy 101/Murray Road Bridge	6	7
04-0229	CE89735	Eureka - Middle Channel Bridge	3	14
04-0230	CE89736	Eureka - Eureka Channel Bridge	4	13
04-0016R	CE89973	Rio Dell - Hwy 101/Eel River Bridge	3	44
54-0823G	CE23631	San Bernardino - I10/215 Interchange	4	9
47-0048	CE54730	Lake Crowley - Hwy 395 Bridge	6	5
04-0236	CE89324	Rio Dell - Hwy 101/Painter St. Overcrossing	6	31
58-0215	CE01336	Hwy8/Meloland Overpass	7	13
33-0214L	CE58658	Hayward Hwy 580-238 Interchange	7	62
Total			93	279

With the establishment of the California Strong Motion Instrumentation Program (CSMIP) in 1972, the California Geological Survey (CGS) began to instrument state bridges with accelerometers. Prior to the 1989 Loma Prieta Earthquake, four bridges were instrumented for strong motions. Following the earthquake, under the advice of the Governors Board of Inquiry and the Caltrans Seismic Advisory Board, Caltrans increased the strong motion bridge instrumentation. Currently, 80 bridges managed by Caltrans are instrumented in partnership with CGS under CSMIP [87] (Fig. 25). Note that the CESMD strong-motion data set includes 82 bridge structures. Of this 82, 80 are instrumented under CSMIP (Code 'CE'), and 2 are instrumented under the National Strong-Motion Project (NSMP, Code 'NP'). The CSMIP network is currently undergoing upgrades with new stations. In addition, the Community Seismic Network (CSN) program has been established to supplement the sensor density of the CSMIP network with lower-cost sensors. These recent developments give promise of improved hazard response through the formation of a comprehensive monitoring system for California's highway bridge network, motivating the initiation of the *BRACE*^{2} project and the establishment of its platform. The remainder of this chapter is dedicated to summarizing the 21 partial digital twins that were investigated throughout Phase I of the *BRACE*^{2} project. A full digital twin is discussed in Chapters 7 to 9.



(a) Number of events recorded and number of sensors at each bridge



(b) Bridge (station) locations

Fig. 25: CSMIP/Caltrans bridge inventory.

6.1 Caltrans Bridge No. 56-0586G

The concrete interchange bridge from Highway Interstate 15 to Hwy 91 in Corona has a curved geometry, with 12 spans crossing over other portions of I15 and Hwy 91 over a length of 1600 feet with a height of about 70 feet. Each bent is supported by a single column with parabolic flares

along the upper portion and steel HP piles serving as their foundation. The structure is relatively new, built in 1989 and its instrumentation have captured three events since the installation in 1994.

6.2 Caltrans Bridge No. 55-0225

The Via California road in Capistrano Beach crosses over Interstate 5 on a six-span straight concrete bridge. The bridge features cantilever abutments with rocker bearings and one in-span hinge near its fourth (out of six bents) column bent. The bridge was constructed in 1960 and instrumented in 1999. Fraino et al. [23, 24] performed system identification on the Capistrano bridge in 2012 and 2014 and found significant modes of vibration with periods of 0.37 and 0.40 seconds.

6.3 Caltrans Bridge No. 53-1471

The Vincent Thomas Bridge is a landmark suspension bridge of the Greater Los Angeles area, connecting the city of San Pedro with Terminal Island. The bridge was first opened in 1963 and was instrumented with 34 accelerometers by Caltrans and CGS in 1981. Several structural analysis and system identification studies [67, 48, 33, 83, 42, 102, 41] have been performed on the bridge, which indicate that the first few vertical modes of the deck have periods of approximately 4.5, 3.0, and 2.0 seconds. The first two lateral mode periods are approximately 7.0 and 2.4 seconds.

6.4 Caltrans Bridge No. 53-2795F

The 1582-foot curved-plan interchange bridge from Hwy 14 West to Interstate 5 South in Sylmar has nine spans. The bridge was constructed and instrumented with 39 accelerometers in 1995. Because of its relatively high sensor density, it has been a subject of interest for system identification studies [5, 4, 63]. Although there are few (three or less) currently available strong motion event records, the literature indicates that these records provided enough dynamic information to identify the lateral modes of vibration. The first lateral mode has a period between 1.2-1.5 seconds and the second lateral mode has a period of approximately 1.0 second.

6.5 Caltrans Bridge No. 53-2791

This portion of Highway Interstate 10, crossing over La Cienega and Venice Blvds in Los Angeles, accommodates four lanes of traffic in each direction. Its deck is comprised of two widths, each supported by five three-column bents. Broderick and Elnashai [11] estimated with a structural analysis model that the bridge has two transverse modes with periods of 0.46 and 0.28 seconds, one longitudinal mode with period of 0.11 seconds, and three deck (vertical) modes with periods of 0.20, 0.16, and 0.16 seconds. Venkittaraman and Banerjee [91] found similar results with their structural analysis model.

6.6 Caltrans Bridge No. 53-1794

The structure allowing Barrel Springs Road to cross over Hwy 14 and the California Aqueduct in Palmdale is supported by four shear wall (infilled) bents. A system identification study by Fraino et al. [23] found that the sensor data indicates a transverse mode with a period of about 0.2 seconds.

6.7 Caltrans Bridge No. 50-0271

A portion of the small Grapevine City Road, Lebec Road, crosses over Highway Interstate 5. The bridge is supported by three shear wall (infilled) bents, and the superstructure is supported by steel plate girders.

6.8 Caltrans Bridge No. 50-0340

A four-span concrete bridge carries Hwy 395 over Brown Road in Ridgecrest. Other than its 45-degree skewed abutments and bents, the bridge has regular geometry and standard construction. The bridge is included in system identification studies of concrete highway bridges [5] and it has an identified fundamental mode period of approximately 0.3 seconds.

6.9 Caltrans Bridge No. 43-0031E

Hwy 156 passes over both directions of Hwy 101 in the city of San Juan Bautista. This overpass is supported by five shear wall (infilled) bents. Fraino et al. [23] performed system identification on this overpass.

6.10 Caltrans Bridge No. 23-0015R

The Carquinez East cantilever bridge was built in 1958 to assist the existing parallel Carquinez West cantilever bridge (built in 1927 and replaced in 2003). This was conducted to accommodate the increasing traffic on Highway Interstate 80 across the Carquinez Strait. Although the 2003 Carquinez West suspension bridge is much more popularly studied in the literature, the Carquinez East bridge is also noted to be subjected to strong motions in the area, such as the M6 South Napa earthquake of August 24, 2014 [17].

6.11 Caltrans Bridge No. 28-0352L

The Carquinez West (Alfred Zampa Memorial) suspension bridge was opened in 2003 to carry southbound traffic on Highway Interstate 80 over the Carquinez Strait from Vallejo to Crockett. Its predecessor, a cantilever bridge built in 1927, was deemed seismically vulnerable after the 1989 Loma Prieta earthquake. Several system identification studies were conducted [9, 17, 28, 105, 46, 94, 34, 106, 65], which found that the bridge's vertical mode periods are around 5.5, 5.0, 3.9, and 2.5 seconds, while the lateral mode periods are around 5.9, 2.7, and 2.6 seconds.

6.12 Caltrans Bridge No. 10-0299

The South Confusion Hill Bridge is a cast-in-place segmental concrete cantilever bridge. It was constructed in 2009 to carry two lanes of Hwy 101 over the South Fork Eel River in Mendocino County, rerouting the highway away from a landslide area. The deck varies in depth to create a slightly arched span. The bridge has two piers, approximately 167.3 feet (51 meters) in height. The bridge was instrumented in 2009 with 21 accelerometers and five strong motions have been recorded at the bridge since that time. No prior studies of this bridge have been published.

6.13 Caltrans Bridge No. 04-0228

The Samoa Channel Bridge is one of the three lengths of State Route 255 that crosses over the Eureka Channel. The bridge is long-span (2506 feet long) and was retrofitted for seismic safety in 2006. It has been studied alongside the Eureka Channel Bridge (Section 6.16) and Middle Channel Bridge (Section 6.15) for its foundation system behavior in response to seismic activities [95]. Several studies have conducted structural analysis modeling and/or system identification on the bridge. These studies [2, 88, 94, 96, 89, 79, 107] indicate a first fundamental mode period of approximately 1.0 second, and a second fundamental mode period of approximately 0.75 seconds.

6.14 Caltrans Bridge No. 04-0170

The simple four-span bridge carrying Murray Road over Hwy 101 in Arcata consists of three two-column bents and a concrete box girder. Fraino et al. [23, 24] conducted system identification studies on the bridge in 2012 and again in 2014 and found a fundamental mode with a period of about 0.15 seconds.

6.15 Caltrans Bridge No. 04-0229

The Middle Channel Bridge is one of the three lengths of State Route 255 that crosses over the Eureka Channel. It has been studied alongside the Eureka Channel Bridge (Section 6.16) and Samoa Channel Bridge (Section 6.13) for its foundation system behavior in response to seismic activities [95].

6.16 Caltrans Bridge No. 04-0230

The Eureka Channel Bridge is one of the three lengths of State Route 255 that crosses over the Eureka Channel. It has been studied alongside the Middle Channel Bridge (Section 6.15) and Samoa Channel Bridge (Section 6.13) for its foundation system behavior in response to seismic activities [95]. Structural analysis modeling was conducted by Almutairi et al. in 2016 [2] and system identification alongside structural analysis modeling was conducted by Wang et al. in 2022 [95].

6.17 Caltrans Bridge No. 04-0016R

The Eel River Bridge was originally constructed in 1940 as a truss and girder bridge. However, several log jams have caused damage to the bridge and resulted in the replacement of several of its spans and frequent repairs have been conducted to keep the bridge in service. Due to the seismic vulnerabilities of the piers, lead-rubber isolation bearings were placed between the piers and bearings in 1986. The December 20, 2022, earthquake in Rio Dell caused abutment unseating of the bridge but no apparent structural damage. No prior studies related to this bridge have been published.

6.18 Caltrans Bridge No. 54-0823G

The North-West connector bridge of the I-10/215 interchange in San Bernardino, Fig. 26, is a 2540 feet long curved bridge (from Abutment1 to Abutment 17) and consists of six sections interconnected by five in-span hinges. The sections have a cast-in-place box girder superstructure and the entire superstructure is supported on 15 single-column bents (Bents 2 to 16) with octagonal section shape. Field-welded steel jackets were used on bent columns under the Caltrans' Phase II seismic retrofit program in 1992. Two types of column retrofits were used, full-height jackets (Class F, Bents 3 to 7, 9 to 11, 13, and 15) and partial-height jackets (Class P, Bents 8, 12, and 14) as well as the combination of Class P and F column retrofit (Bents 2 and 16). In addition, seven outrigger bents with regular octagonal section shape and double-column bents constructed at the fault rupture zone (between Bents 10 and 15), under the Earthquake retrofit project No. 634 in 2005, support the box girder superstructure.

(a) Elevation view facing northwest (spans #6 through 16) (source: Caltrans)

(b) Plan view (source: Google Maps)

Fig. 26: San Bernardino bridge photographs.

System identification studies have been performed on the bridge by Huang and Shakal in 1995 [31], Desroches and Fenves in 1997 [21], and Mosquera et al. in 2009 [63]. Structural analysis modeling of the bridge was performed by Kim and Elgamal in 2014 [45]. These studies indicate that there is a fundamental vibrational mode with a period of about 1.5 seconds.

6.19 Caltrans Bridge No. 47-0048

The Crowley Lake Bridge is a reinforced concrete box girder highway bridge spanning 203 feet. A 40-foot wide roadway accommodates two lanes. The bridge is straight and supported on a double column bent as well as two diaphragm abutments. The columns have identical circular cross-sections with spiral reinforcement. The structure has been instrumented with 9 sensors, Fig. 27. In [5], a fundamental frequency of 4.8 Hz was observed using an ARX (AutoRegressive models with eXogenous inputs, refer to Section 13.5.3) method, but there was high variability in the study and this value appears inconclusive.

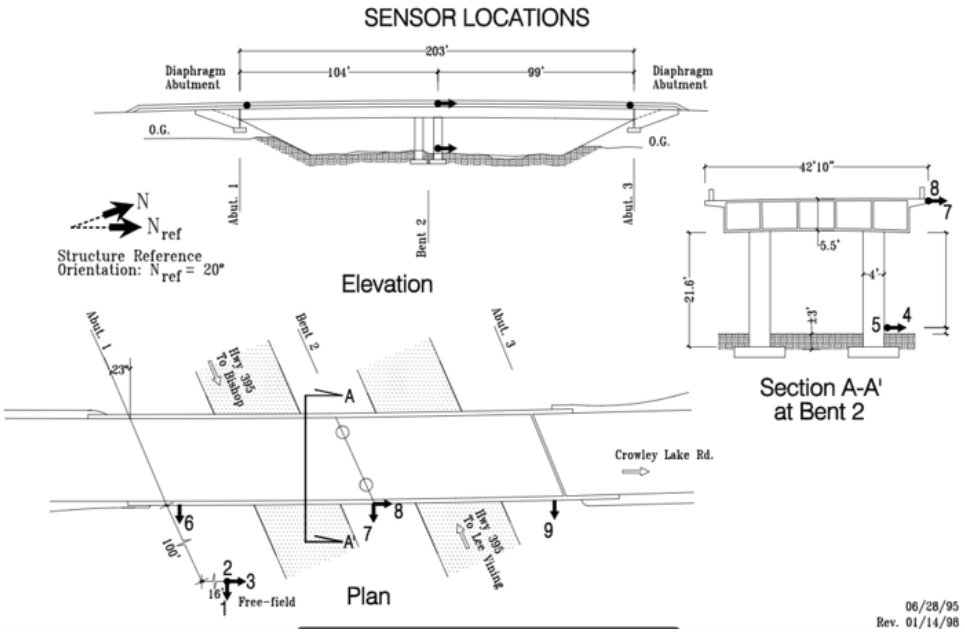


Fig. 27: Crowley Lake bridge plan and sensor locations.

Fraino et al. [23, 24] performed system identification on the Crowley Lake Bridge in 2012 and again in 2014. Moreover, Arici and Mosalam [5] studied the system identification of this bridge in 2003. These studies indicate that the bridge has two significant natural modes of vibration with close periods, namely, about 0.2 and 0.18 seconds.

6.20 Caltrans Bridge No. 04-0236

The US 101/Painter Street Overpass is a continuous reinforced concrete multicell box-girder bridge, Fig. 28. The skewed 265-foot-long deck consists of two unequal spans which meet at a two-column bent. The 40-foot-wide roadway accommodates one traffic lane in each direction, each with a comfortable shoulder. There is no horizontal or vertical curvature to the structure. The structure sits on two skewed integral abutments and two column footings supported on 20 piles each.

The east abutment is monolithic with the superstructure and is supported by 14 driven 45-ton concrete friction piles. The west abutment has a thermal expansion joint and rests on a neoprene bearing strip. The foundation of this abutment consists of 16 driven 45-ton concrete friction piles.

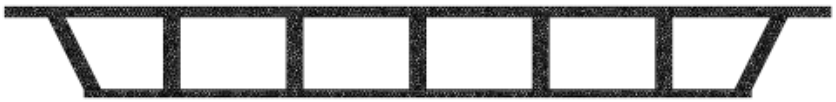


Fig. 28: Finite element model used to obtain warping constant of US 101/Painter St. Overpass.

Continuous flared columns use an obsolete detail. An experimental investigation on this matter was conducted by Sanchez et al. [76]. A numerical investigation of the continuous detail is conducted by Soleimani et al. [84]. This issue is addressed by Caltrans [13].

The Painter Street Overpass was instrumented in 1977. Channels which may be considered for transverse excitation include 7, 9, 17, and 20. Longitudinal field channels include 11, 15, and 18, Fig. 29. Various system identification studies have been conducted by the *BRACE*^{2} team and results of several historical transfer function estimates are provided in Fig. 30. More details on these procedures are provided in Part II.

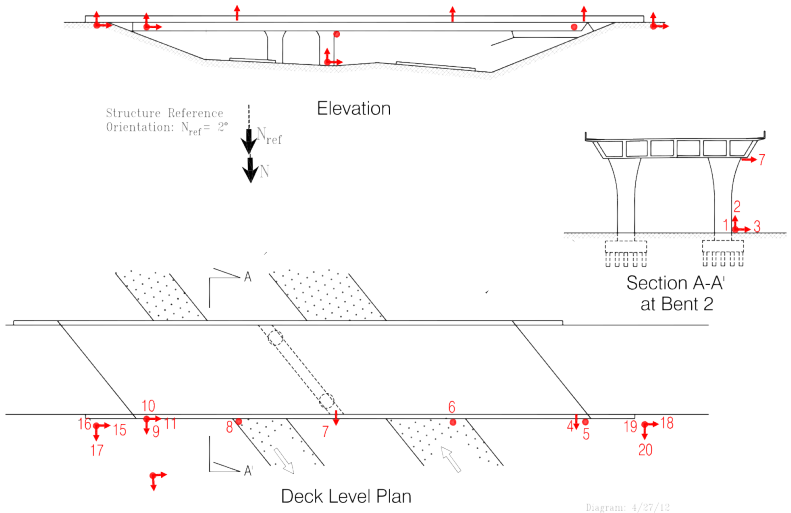


Fig. 29: Instrumentation of the Painter Street Overpass.

Maroney et al. [51] utilized records from this Painter Street Overpass in conjunction with a number of finite-element and lumped-parameter (stick) models of the entire bridge. However, none of these models accounted for soil-foundation-superstructure interaction. At each abutment, soil-wall interaction was modeled through a single real-valued transverse spring, the stiffness of which was inversely determined from the interpreted fundamental natural period, $T \approx 0.30$ seconds, in lateral vibration. Goel and Chopra [26] identified the following abutment stiffness parameters for two excitation intensities:

Excitation	**Lower bound (kips/ft)**	**Upper bound (kips/ft)**
Longitudinal (low intensity)	11,000	13,115
Transverse (high intensity)	7,500	12,000

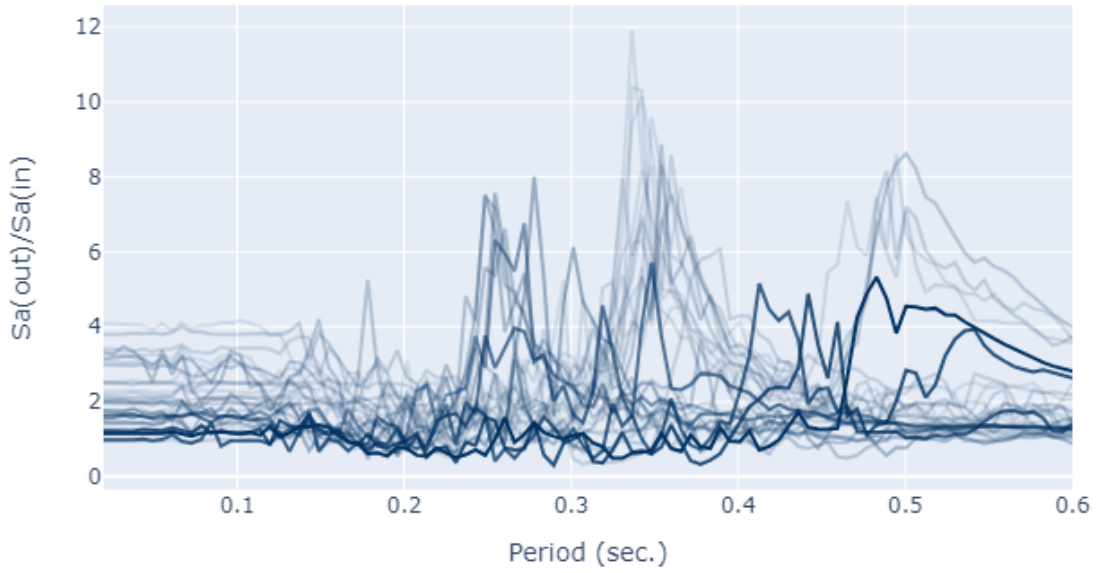


Fig. 30: Historical transfer function estimates of 29 events for the Painter St. Overpass (transverse direction using Channels 3 vs. 7). Event dates range between 1980 and 2023 with peak station accelerations ranging between $0.1g$ and $0.63g$.

Additional studies whose findings have been worked into the parameterization and calibration of the Painter Street models are those of Zhang and Makris [103] and Arici and Mosalam [5]. Other system identification and modeling studies have been performed and consistent identification of a fundamental period of about 0.29 seconds and a secondary period of about 0.23 seconds have been obtained and verified [23, 5, 104, 74, 89, 63, 93, 55, 92].

6.21 Caltrans Bridge No. 58-0215

The Meloland Overpass is a cast-in-place box girder bridge located near El Centro, California. This structure is notable for having been extensively equipped with 26 strong motion accelerometers at the time of the 1979 Imperial Valley earthquake with a surface distance of 12 miles from the epicenter. The as-built drawings supplied for the Meloland Overpass indicate that its design is in accordance with an AASHTO specification dated 1965. The following procedures were investigated for this structure:

- **Modal Analysis**: First, a modal analysis is conducted to verify the general integrity of the model. Using linear abutment stiffness values obtained from the literature, the model exhibits a transverse fundamental period roughly in the range of $0.3 - 0.4$ seconds and longitudinal period in the range of $0.2 - 0.3$ seconds.
- **Sampling Procedure**: A sampling procedure is then conducted which provides an understanding of these ranges.
- **Global Sensitivity**: Results from the sampling analysis are used to constrain further the range of parameters used and a global sensitivity analysis is conducted.
- **System Identification**: Several system identification procedures are configured for the struc-

ture which can be used to monitor the modal properties of the structure for any event that is sent to the *BRACE*^{2} platform.

The Meloland Overpass comprises two equal spans of 104 feet each. A 32-foot wide roadway accommodates one lane of traffic in each direction along a straight alignment, free of both horizontal and vertical curvatures. The abutments of the Meloland Overpass are a sub-type of integral abutments known as *diaphragm abutments* [14]. Basic properties for concrete in linear members taken according to Table 3.3.6-1 of California Department of Transportation [12] furnish values of the concrete elastic modulus, E_c, and shear modulus, G_c, as follows:

Modulus	**Value (ksi)**
E_c	$4,074$
G_c	$1,698$

Optimal elastic modulus values of $3,705$ and $3,891$ ksi were reported in Mosquera et al. [61, 62] in 2009 and 2012, respectively. These models were respectively calibrated from ranges of $3,194 - 3,889$ ksi and $3,191 - 4,061$ ksi. Assuming a unit weight of 145 pcf, the mass density of the girder is roughly 1.4 lbm/in. For reference, this results in a total super structure weight of about 112 kips (i.e., 50.8 metric tons). A finite element model of a representative cross section is presented in Fig. 12.

A series of early studies of the Meloland Overpass were conducted by Werner et al. [98], Wilson [99], Levine and Scott [47], and Wilson and Tan [100]. Werner et al. [98] conclude that the abutments and embankments are the primary contributors to the transverse response, meanwhile vertical response characteristics were dominated by properties of the deck. Levine and Scott [47] build on the study of Werner et al. [98] to produce an estimate for linear rotational stiffnesses at the abutments using a system identification procedure developed by Beck [8] (sometimes identified as the "MODE-ID" technique). The model used in this study is a simple linear 3D beam assemblage with fixed translational resistance in the abutments. Wilson and Tan [100] use a single-input/single-output (SISO) structural identification technique to further explore the abutment response. Zhang and Makris [103] builds upon the work of Wilson and Tan [100], Werner et al. [98], and Goel and Chopra [26]. Arici and Mosalam [5] conducted system identification studies of the bridge in 2003, finding that the fundamental period of vibration remained consistent within the range of 0.29 to 0.33 seconds, and that there appeared to be a secondary mode of vibration with a period within the range of 0.25 and 0.28 seconds. Fraino et al. [23, 24] conducted system identification studies of the bridge in 2012 and 2014, respectively, finding that the fundamental period of vibration remained consistent within the range of 0.26 and 0.29 seconds over five different shaking events.

Chapter 7

The Physical Hayward Bridge

Hayward Bridge (Fig. 31) is a highway interchange constructed in 1988 to connect California Interstates 580 and 238. The bridge is managed by Caltrans and is known as the Route 580/238 Separation, Bridge No. `33-0214L`. The bridge is located where the cities of Ashland, Castro Valley, and Hayward meet, with three other highways as well as the BART line crossing beneath it. Due to its proximity to the Hayward Fault and its importance in the transportation network, the Hayward Bridge requires critical decisions on its operational status when strong motion events occur. Its *digital twin* provides continuous monitoring information to aid in these decisions, as well as a recorded health history that can inform future design and monitoring activities.

Hayward Bridge is a cast-in-place (CIP) reinforced concrete bridge with 14 spans in a curved plan shape (Fig. 32). Its prestressed concrete (PC) box girder superstructure is supported on single, double, and triple column bents and seat-type abutments. The main line of the superstructure is denoted NE1, at the east end of the bridge between Bents 12 and 15. There is a secondary line denoted NR1 which branches off in the southeast direction. The bridge is separated into four frames by in-span hinges located near Bents 5, 8, and 12. Table 8 presents key data about Hayward Bridge.

Table 8: Hayward Bridge key data.

Caltrans Bridge No.	33-0214L
CGS Station No.	58658
Bridge Name	Route 580/238 Separation
Year Built	1988
Latitude, Longitude	37.6907N, 122.0993W
Structure Type	CIP/PC 5-cell box girder with 1-3 columns per bent
Structure Depth	$6' - 6''$ (Frame 1, 3 & 4); $6' - 6''$ to $8' - 0''$ and varies (Frame 2)
Bridge Length	$2,030' - 0''$
Abutments	Seat type
Bents	1, 2 & 3 column bents

The Hayward Bridge was instrumented in 1993 under the interagency agreement between Caltrans and the California Department of Conservation (DOC), with seven accelerometers on the bridge and three accelerometers at a free-field site. In 2011, the bridge was re-instrumented with 20

(a) Eastward view from west abutment (source: Caltrans)

(b) Westward view from northeast abutment (source: Caltrans)

(c) Northeastward view from Bent 4

(d) Westward view from Bent 8

(e) Southeastward view from Bent 4

(f) Plan view (source: Google Maps) with approximate superstructure extents outlined

Fig. 31: Hayward Bridge photographs.

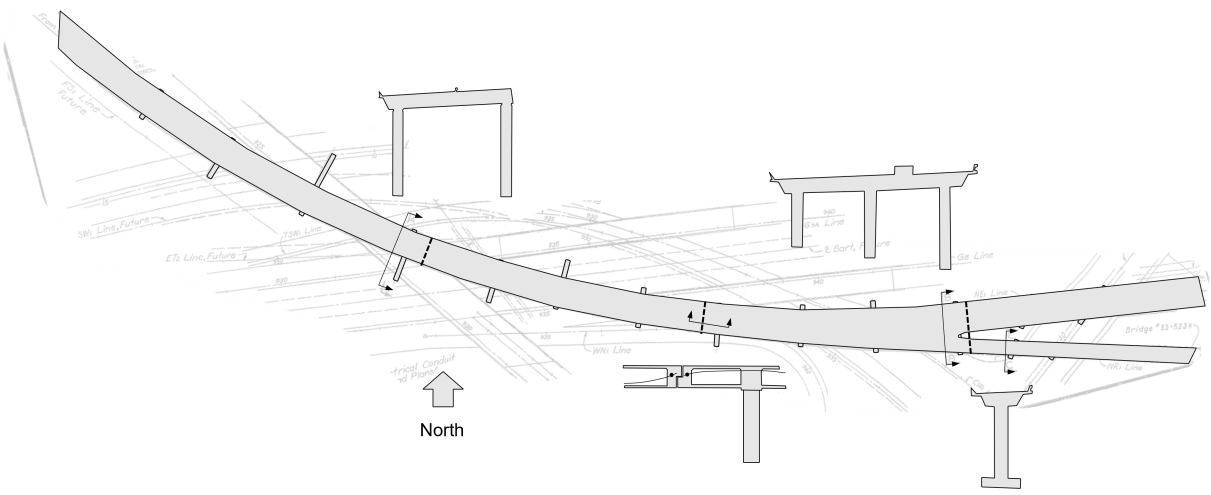
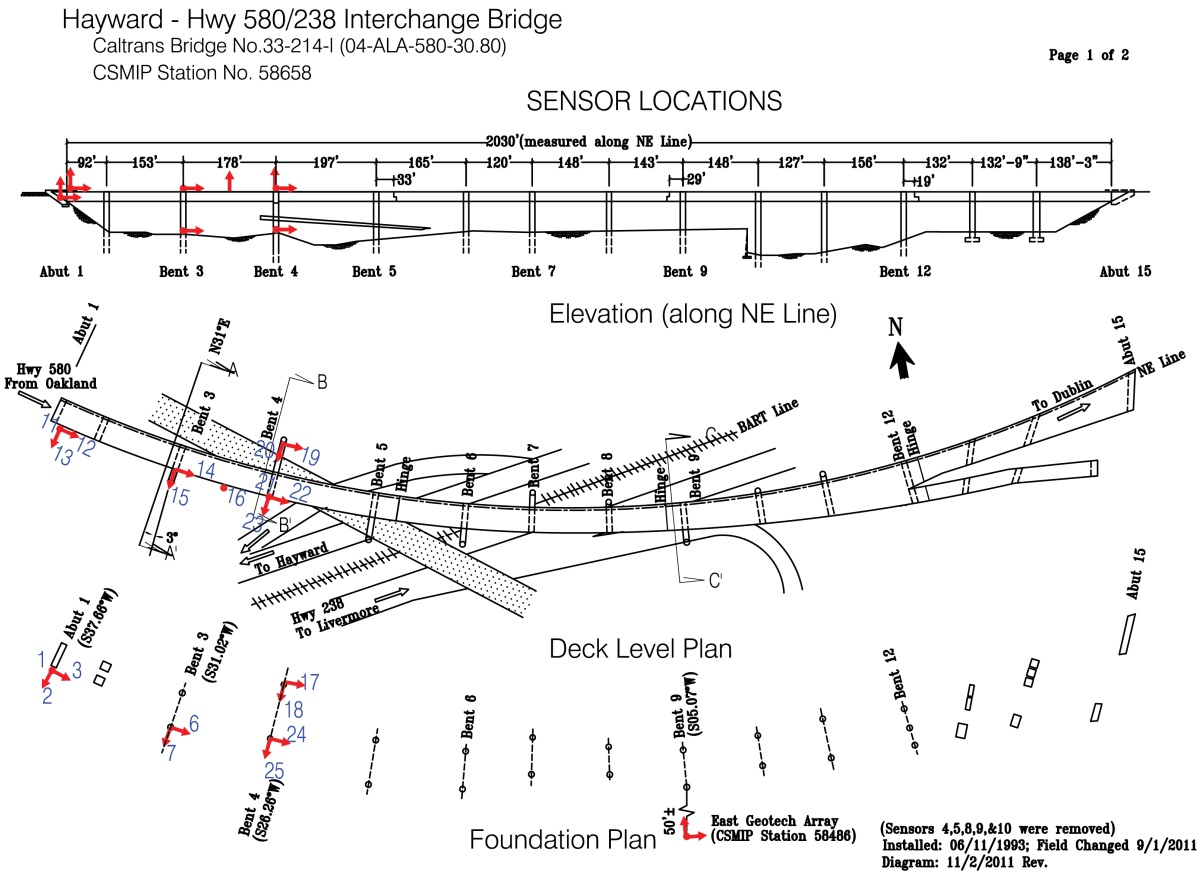


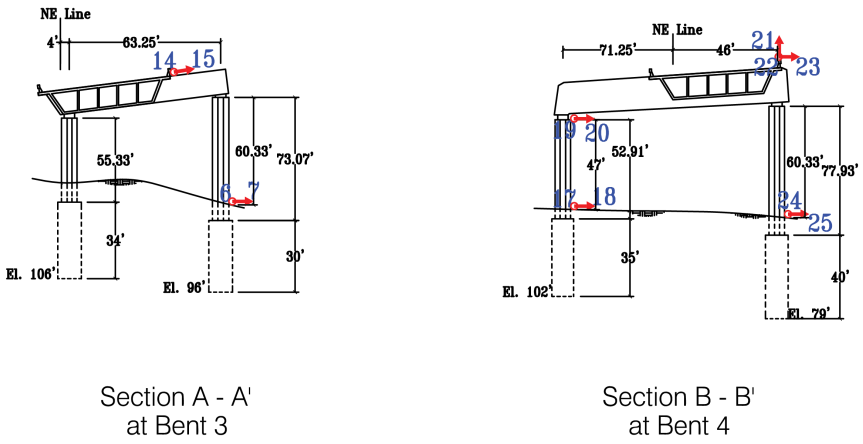
Fig. 32: Hayward Bridge plan sketch with section views of column bents and in-span hinges.

accelerometers (Fig. 33). All of the sensors on the bridge are installed at the first frame (between Abutment 1 and the in-span hinge near Bent 5), and Bents 3 and 4 include sensors at both the ground and the superstructure level.

Since 2021, 82 vibration response histories have been collected from the sensors. Out of the 82 records, 62 events are recorded at the structure during ground motion excitation (Table 9). Three events are recorded at free field locations during ground motion excitation and 17 events are function tests (Table 10). For the 62 ground motion excitation records at the structure, the distribution of peak recorded accelerations (from either the superstructure or at the base of columns) and the progression of record collection over time are shown in Fig. 34. All of the recorded motions are small excitations, causing no damage at the bridge, where the largest event has a peak recorded acceleration of 0.129g. A log-normal distribution is fit to the peak recorded accelerations of the events to visualize the skewness of the distribution [1]. There are not enough data to assign significant meaning to the fitted curve. However, the curve illustrates that potentially damaging events, e.g., those causing peak recorded acceleration larger than 1.0g, lie too far along the tail of the curve to expect their accurate representation in the collected dataset. This sparsity of data is typical of response records collected from structures, making it difficult to validate predictions for higher magnitude events. Thus, health monitoring applications often rely on characterizing the "normal" conditions such that abnormal ones can be identified or on data augmentation from *simulated events* or from an inventory



(a) Plan.



(b) Elevation.

Fig. 33: Accelerometer locations and orientations of Hayward Bridge (source: CGS).

combining many bridges to improve the prediction validation.

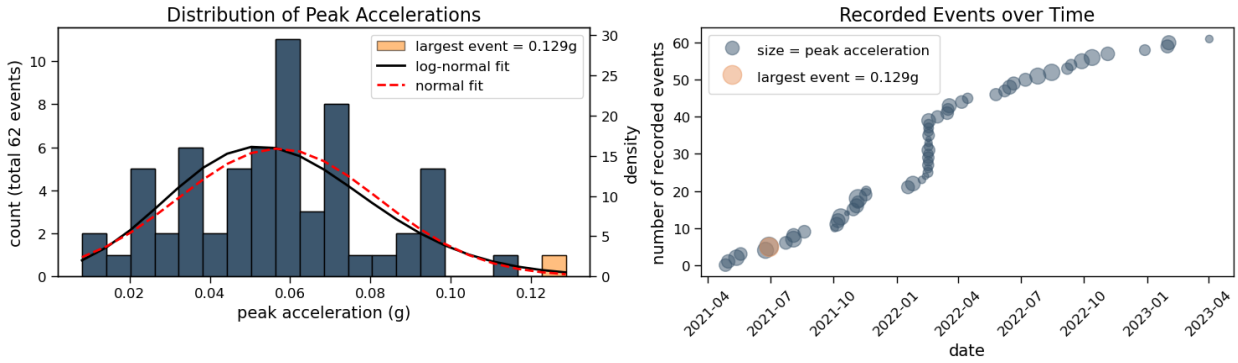


Fig. 34: Ground motion events recorded on the structure and analyzed of Hayward Bridge.

Table 9: CGS Data Summary: 62 events analyzed for Hayward Bridge.

Event File Name	Date & Time	Peak Acc. (g)	Notes
58658_000_20220215_17.46.29.P.zip	2022-02-16T01:46:00	0.046	
58658_000_20220215_21.05.13.P.zip	2022-02-16T05:05:00	0.033	
58658_000_20220216_06.20.58.P.zip	2022-02-16T14:20:00	0.010	
58658_000_20220216_07.50.13.P.zip	2022-02-16T15:50:00	0.031	
58658_001_20211011_09.22.04.P.zip	2021-10-11T16:22:00	0.097	Invalid reading at Channel 22
58658_001_20220215_16.08.15.P.zip	2022-02-16T00:08:00	0.024	
58658_001_20220215_22.09.12.P.zip	2022-02-16T06:09:00	0.049	
58658_001_20220216_00.35.36.P.zip	2022-02-16T08:35:00	0.033	
58658_001_20220216_01.36.38.P.zip	2022-02-16T09:36:00	0.061	
58658_001_20220216_04.58.04.P.zip	2022-02-16T12:58:00	0.028	
58658_001_20220216_06.05.23.P.zip	2022-02-16T14:05:00	0.021	
58658_001_20220216_07.39.12.P.zip	2022-02-16T15:39:00	0.050	
58658_001_20220216_07.56.05.P.zip	2022-02-16T15:56:00	0.035	
58658_001_20220216_09.57.45.P.zip	2022-02-16T17:57:00	0.070	
58658_001_20220301_07.29.35.P.zip	2022-03-01T15:29:00	0.056	
58658_001_20220405_16.10.04.P.zip	2022-04-05T23:10:00	0.059	
58658_001_20220607_07.22.12.P.zip	2022-06-07T14:22:00	0.052	
58658_001_20221228_11.52.14.P.zip	2022-12-28T19:52:00	0.043	
58658_001_20230130_09.55.06.P.zip	2023-01-30T17:55:00	0.062	
58658_002_20210512_09.31.03.P.zip	2021-05-12T16:31:00	0.088	Invalid reading at Channel 22
58658_002_20210623_17.15.07.P.zip	2021-06-24T00:15:00	0.094	Invalid reading at Channel 22
58658_002_20211020_12.01.50.P.zip	2021-10-20T19:01:00	0.008	Invalid reading at Channel 22
58658_002_20220216_08.03.03.P.zip	2022-02-16T16:03:00	0.036	
58658_002_20220315_11.12.29.P.zip	2022-03-15T18:12:00	0.058	
58658_002_20220414_07.26.44.P.zip	2022-04-14T14:26:00	0.039	
58658_002_20220614_11.35.52.P.zip	2022-06-14T18:35:00	0.071	
58658_002_20230201_10.22.35.P.zip	2023-02-01T18:22:00	0.073	
58658_003_20210518_13.38.19.P.zip	2021-05-18T20:38:00	0.059	Invalid reading at Channel 22
58658_003_20210628_18.29.26.P.zip	2021-06-29T01:29:00	0.129	Invalid reading at Channel 22
58658_003_20211002_12.22.00.P.zip	2021-10-02T19:22:00	0.021	Invalid reading at Channel 22
58658_003_20220316_18.13.42.P.zip	2022-03-17T01:13:00	0.051	
58658_003_20220525_12.45.41.P.zip	2022-05-25T19:45:00	0.054	
58658_003_20220620_09.12.14.P.zip	2022-06-20T16:12:00	0.063	
58658_004_20210723_08.59.25.P.zip	2021-07-23T15:59:00	0.059	Invalid reading at Channel 22
58658_004_20211005_12.19.15.P.zip	2021-10-05T19:19:00	0.070	Invalid reading at Channel 22
58658_004_20211029_15.43.45.P.zip	2021-10-29T22:43:00	0.058	
58658_004_20220318_11.46.17.P.zip	2022-03-18T18:46:00	0.073	
58658_004_20220707_13.31.02.P.zip	2022-07-07T20:31:00	0.062	
58658_004_20230401_09.43.22.P.zip	2023-04-01T16:43:00	0.021	
58658_005_20210803_12.52.01.P.zip	2021-08-03T19:52:00	0.091	Invalid reading at Channel 22
58658_005_20211007_11.51.59.P.zip	2021-10-07T18:51:00	0.071	Invalid reading at Channel 22
58658_005_20211103_09.28.30.P.zip	2021-11-03T16:28:00	0.072	
58658_005_20220725_12.09.51.P.zip	2022-07-25T19:09:00	0.094	
58658_006_20210803_16.45.39.P.zip	2021-08-03T23:45:00	0.069	Invalid reading at Channel 22
58658_006_20211105_12.07.32.P.zip	2021-11-05T19:07:00	0.056	
58658_006_20220814_18.55.32.P.zip	2022-08-15T01:55:00	0.098	
58658_007_20210426_10.09.54.P.zip	2021-04-26T17:09:00	0.058	
58658_007_20210819_08.30.48.P.zip	2021-08-19T15:30:00	0.059	Invalid reading at Channel 22
58658_007_20211105_15.20.50.P.zip	2021-11-05T22:20:00	0.114	
58658_007_20220906_12.28.54.P.zip	2022-09-06T19:28:00	0.047	
58658_008_20211116_18.00.24.P.zip	2021-11-17T02:00:00	0.056	
58658_008_20220912_03.26.48.P.zip	2022-09-12T10:26:00	0.046	
58658_009_20210430_09.40.12.P.zip	2021-04-30T16:40:00	0.066	Invalid reading at Channel 22
58658_009_20211117_11.43.20.P.zip	2021-11-17T19:43:00	0.034	
58658_009_20220927_14.49.19.P.zip	2022-09-27T21:49:00	0.082	
58658_010_20220117_07.38.48.P.zip	2022-01-17T15:38:00	0.061	
58658_010_20221012_16.02.43.P.zip	2022-10-12T23:02:00	0.093	
58658_011_20220124_11.12.50.P.zip	2022-01-24T19:12:00	0.079	
58658_011_20221104_11.31.26.P.zip	2022-11-04T18:31:00	0.067	
58658_012_20220206_16.01.26.P.zip	2022-02-07T00:01:00	0.020	
58658_013_20220211_19.13.15.P.zip	2022-02-12T03:13:00	0.016	
58658_014_20220215_12.01.04.P.zip	2022-02-15T20:01:00	0.036	

Table 10: CGS Data Summary: 20 events not analyzed for Hayward Bridge.

Event File Name	Date & Time	Peak Acc. (g)	Notes
58486_006_20210628_18.29.21.P.zip	2021-06-29T01:29:00	0.032	Free Field Record
58486_007_20211117_11.43.20.P.zip	2021-11-17T19:43:00	0.024	Free Field Record
58486_012_20220311_09.48.33.P.zip	2022-03-11T17:48:00	0.020	Free Field Record
58486_003_20210322_12.43.52.P.zip	2021-03-22T19:43:00	0.522	Function Test, Free Field
58486_005_20210527_14.24.21.P.zip	2021-05-27T21:24:00	0.518	Function Test, Free Field
58486_008_20220310_11.34.46.P.zip	2022-03-10T19:34:00	0.522	Function Test, Free Field
58486_009_20220310_11.49.29.P.zip	2022-03-10T19:49:00	0.522	Function Test, Free Field
58486_010_20220310_15.40.52.P.zip	2022-03-10T23:40:00	0.520	Function Test, Free Field
58486_011_20220311_08.08.51.P.zip	2022-03-11T16:08:00	0.523	Function Test, Free Field
58486_013_20220311_09.49.52.P.zip	2022-03-11T17:49:00	0.522	Function Test, Free Field
58486_014_20220411_12.42.43.P.zip	2022-04-11T19:42:00	0.521	Function Test, Free Field
58486_015_20220505_09.42.33.P.zip	2022-05-05T16:42:00	0.521	Function Test, Free Field
58486_016_20220505_10.42.29.P.zip	2022-05-05T17:42:00	0.520	Function Test, Free Field
58658_001_20220216_10.42.41.P.zip	2022-02-16T18:42:00	0.091	Function Test
58658_001_20220217_08.16.21.P.zip	2022-02-17T16:16:00	0.089	Function Test
58658_001_20220224_09.44.11.P.zip	2022-02-24T17:44:00	0.089	Function Test
58658_001_20221213_13.40.31.P.zip	2022-12-13T21:40:00	0.102	Function Test
58658_002_20220224_10.47.00.P.zip	2022-02-24T18:47:00	0.089	Function Test
58658_003_20211020_12.04.41.P.zip	2021-10-20T19:04:00	0.093	Function Test
58658_012_20221213_12.26.41.P.zip	2022-12-13T20:26:00	0.092	Function Test

Chapter 8

The Digital Hayward Bridge

A global numerical structural analysis model of Hayward Bridge is developed in OpenSees [56, 54] based on the as-built drawings. Simulated responses include forces, displacements, accelerations, stresses, and strains. This chapter describes the methods and assumptions adopted for the model. Note that glossaries for material and modeling properties are compiled and included in Section 8.9.

8.1 Material Properties

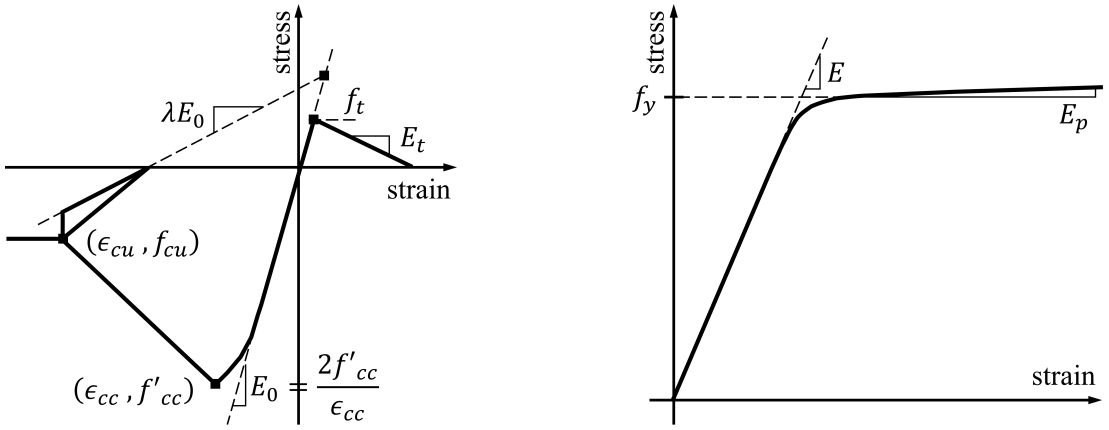
The material properties of the concrete and steel are listed in Tables 11 and 12, respectively. At components that are modeled as elastic beam-column elements, the material behavior depends on the concrete elastic and shear moduli listed in these tables and the cross-sectional area and moments of inertia determined from the as-built drawings. At components that are modeled as nonlinear beam-column elements, the material behavior depends on the hysteretic stress-strain relationships shown in Fig. 35. Concrete follows a Hognestad envelope (Eq. (4.28) of Section 4.5.1) with $E_0 = 2f'_{cc}/\epsilon_{cc}$ and degraded linear unloading/reloading stiffness according to Karsan-Jirsa [54] (Fig. 35a), with the assumption that all tensile strength assumed to have been lost herein, i.e., $f_t = 0$ and $E_t = 0$ (refer to Fig. 35a). This is especially true and conservatively assumed after the bridge's 35 years in service. The ratio between unloading and initial slope, λ, is set to 0.1. Steel reinforcing bars follow the Giuffrè-Menegotto-Pinto (GMP) model with isotropic strain hardening, $E = E_s$, and $E_p = 0.02E$ (Fig. 35b).

Table 11: Hayward Bridge numerical model concrete material properties.

Concrete Material Property		**Value**
General unconfined compressive strength (ksi)	f'_{ce}	5.0
Bent 4 cap beam unconfined compressive strength (ksi)	f'_{ce}	5.85
Bents 5 to 11 cap beam unconfined compressive strength (ksi)	f'_{ce}	5.2
Unconfined strain at maximum strength	ϵ_{c0}	0.002
Unconfined crushing (spalling) strain	ϵ_{sp}	0.005
Compressive elastic modulus (ksi)	E_c	$57\sqrt{f'_{ce}\ (\text{psi})}$
Poisson's ratio	ν_c	0.2
Shear modulus	G_c	$E_c/(2+(1+\nu_c))$

Table 12: Hayward Bridge numerical model steel material properties.

Steel Material Property		**Value**
Tensile elastic modulus (ksi)	E_s	$29,000$
Initial strain hardening tangent	E_{sh}	$0.02E_s$
Yield strength (ksi)	f_y	68
Ultimate strength (ksi)	f_u	95
Strain at initial strain hardening	ϵ_{sh}	0.0075
Strain at peak stress	ϵ_{su}	0.090



(a) Concrete (b) Steel

Fig. 35: Hayward Bridge nonlinear material models.

The model for Hayward Bridge considers properties of the unconfined concrete, $f'_{cc} = f'_{ce}$, $\epsilon_{cc} = \epsilon_{c0}$, $f_{cu} = 0.1f'_{ce}$, and $\epsilon_{cu} = \epsilon_{sp}$. For the confined concrete, the considered properties f'_{cc}, ϵ_{cc}, f_{cu}, and ϵ_{cu} are calculated according Mander's procedure [50] with an adjustment to the crushing stress-strain pair (ϵ_{cu}, f_{cu}), refer to Equation Eq. (8.1). The adjustment is calculated by extrapolating the line between (ϵ_{cc}, f'_{cc}) and $(\epsilon_{cu0}, f_{cu0})$ to $0.15f'_{cc}$. This extends the compressive descending branch of the curve, which improves numerical stability during the NRHA.

$$
\begin{aligned}
f'_{cc} &= f'_{ce}\left(-1.254 + 2.254\sqrt{1 + \frac{7.94f_{2e}}{f'_{ce}}} - 2\frac{f_{2e}}{f'_{ce}}\right),\\
\epsilon_{cc} &= \epsilon_{c0}\left(1 + 5\left(\frac{f'_{cc}}{f'_{ce}} - 1\right)\right),\\
f_{cu} &= 0.15f'_{cc},\\
\epsilon_{cu} &= \epsilon_{cu0} + (\epsilon_{cu0} - \epsilon_{cc}) \times \frac{(f_{cu} - f_{cu0})}{(f_{cu0} - f'_{cc})}.
\end{aligned}
\tag{8.1}
$$

with

$$
\begin{aligned}
f_{cu0} &= \frac{r f'_{cc} \xi}{r - 1 + (\xi)^r}, \\
\epsilon_{cu0} &= 0.004 + \frac{f_{2e}}{4 f'_{ce}}, \\
\rho_{tr} &= \frac{4 A_{sp}}{s_{tr} D_{tr}}, \\
k_e &= 1 - \frac{s_{tr}}{D_{tr}}, \\
f_{2e} &= \frac{k_e \rho_{tr} f_{ye}}{2}, \\
\xi &= \frac{\epsilon_{cu0}}{\epsilon_{cc}}, \\
r &= \frac{E_c}{E_c - (f'_{cc}/\epsilon_{cc})},
\end{aligned}
$$

where the following detailing parameters have been used:

A_{sp} : Area of transverse spiral reinforcing bar,
s_{tr} : Vertical spacing of transverse reinforcing spiral,
D_{tr} : Diameter of transverse reinforcing spiral, and
ρ_{tr} : Transverse reinforcement ratio.

8.2 Columns

The columns (Fig. 36) are modeled as nonlinear beam-column elements with $n_p = 4$ integration points and fiber discretized sections. The column elements are massless because they have relatively small mass compared to the superstructure. The typical column is octagonal in cross section (Fig. 37a) and confined by spiral transverse reinforcement. At the NR1 line, the columns of Bents 13 and 14 have a wide interlocking octagonal cross section (Fig. 37b). Generally, column bases are fixed in all Degree Of Freedom (DOF), three translations and three rotations, and the top node of each column is rigidly connected to the center of the cap beam. However, the boundary conditions vary for columns with pinned ends, as described below. In the fiber section, there are 10 fibers in the radial direction of the core and four fibers in the radial direction of the cover. Moreover, there are 64 fibers in the circumferential direction. Optionally, columns can be modeled as elastic beam-column elements to decrease computational demands during the NRHA.

Hayward Bridge columns vary in their fixity conditions. The typical column is designed to behave as pin-fixed (pinned at the top and fixed at the base) (Fig. 38a). At Bents 13 and 14 on the NR1 line, the columns are designed as fixed-fixed (Fig. 38b). At Bents 2, 12, 13, and 14 on the NE1 line, the columns are designed as fixed-pin (Fig. 38c and Fig. 38d). To represent the behavior of the different column boundary conditions, five modeling options with varying degrees of fixity and complexity are considered, as described in Table Table 13 and Fig. 39.

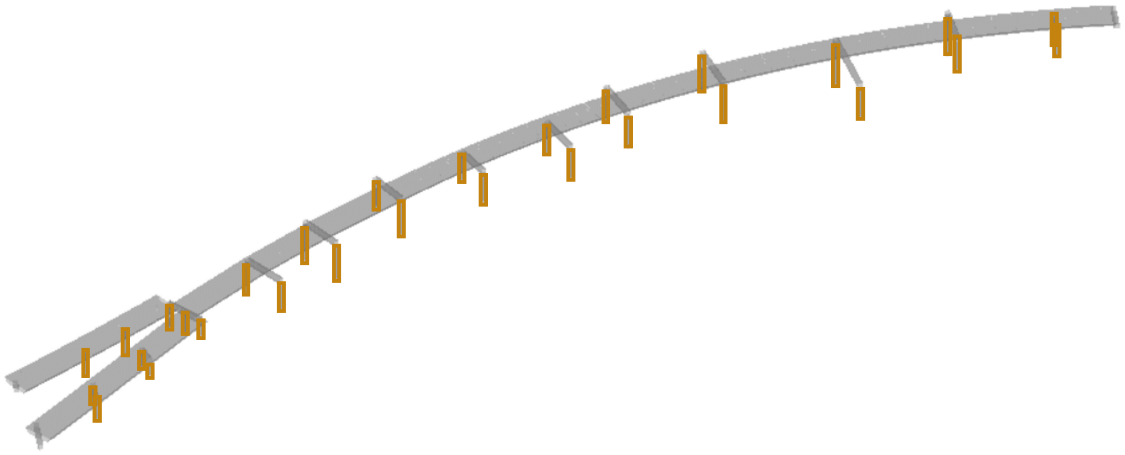
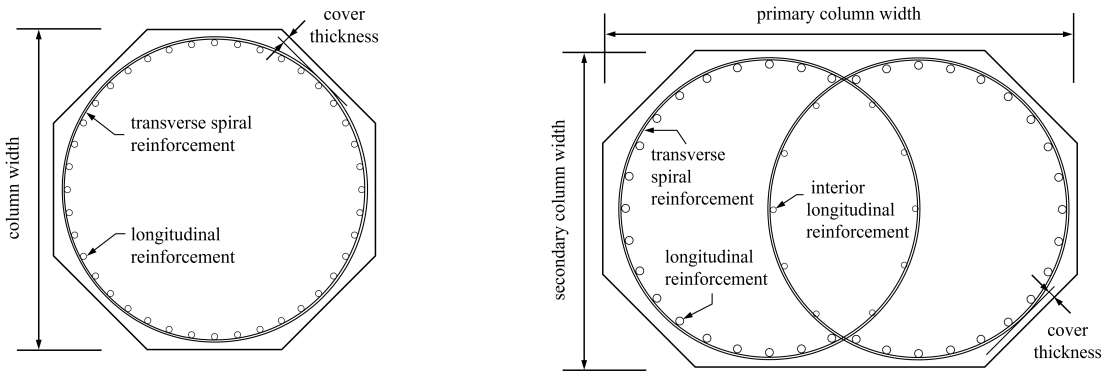


Fig. 36: Hayward Bridge numerical model: Columns.



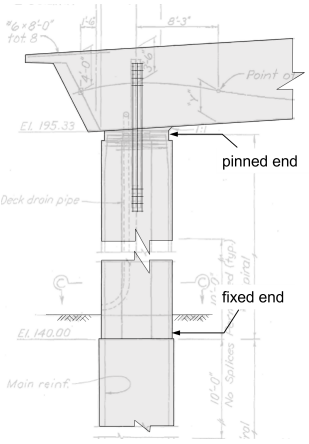
(a) Typical column

(b) Interlocking wide column

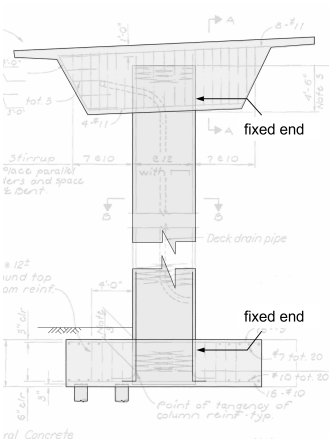
Fig. 37: Hayward Bridge column cross sections.

Table 13: Hayward Bridge numerical model column fixity options.

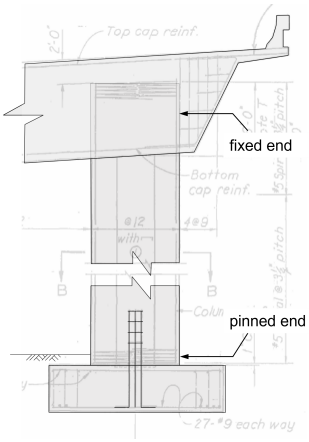
Pin Model	Name	Description
1	All Rigid	Omit pins; treat all columns as fixed-fixed
2	All Pin	Zero rotational stiffness about two horizontal axes at all pins.
3	Mixed Rigid/Pin	Where gravitational moment demand is expected to exceed the pin's cross-sectional moment capacity, apply zero rotational stiffness about two horizontal axes.
4	Fiber Sections	Model the pins explicitly as doweled, unreinforced concrete fiber sections.
5	Integrated Fiber Sections	At the corresponding integration point of each column element, model the pins explicitly as doweled, unreinforced concrete fiber sections.



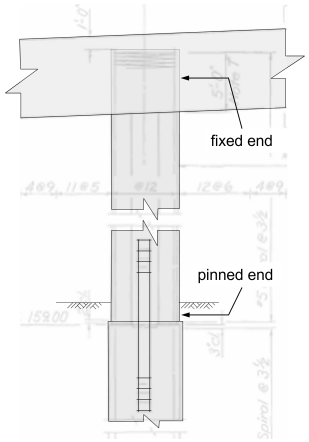
(a) Pin-fixed typical column with shaft foundation



(b) Fixed-fixed column with pile group foundation

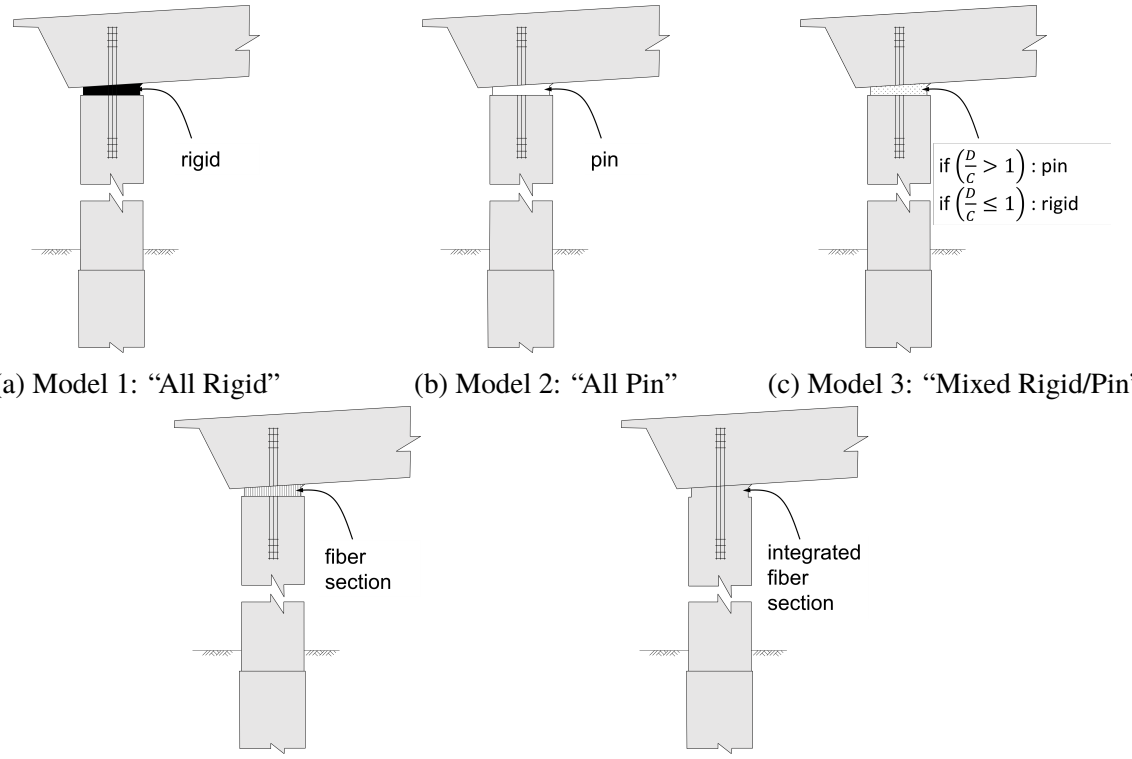


(c) Fixed-pin column with spread footing foundation



(d) Fixed-pin column with shaft foundation

Fig. 38: Hayward Bridge column fixity configurations.



(a) Model 1: "All Rigid" (b) Model 2: "All Pin" (c) Model 3: "Mixed Rigid/Pin"

(d) Model 4: "Fiber Sections" (e) Model 5: "Integrated Fiber Sections"

Fig. 39: Hayward Bridge column pin models (D: Demand; C: Capacity).

8.3 Cap Beams

The cap beams (Fig. 40) are modeled as elastic beam-column elements with a consistent mass matrix and cross-sectional area and moment of inertia calculated from the as-built drawings. Where the superstructure coincides with the cap beam, the calculation of the moment of inertia includes a "flange" (Fig. 41) representing the additional rotational inertia from the superstructure.

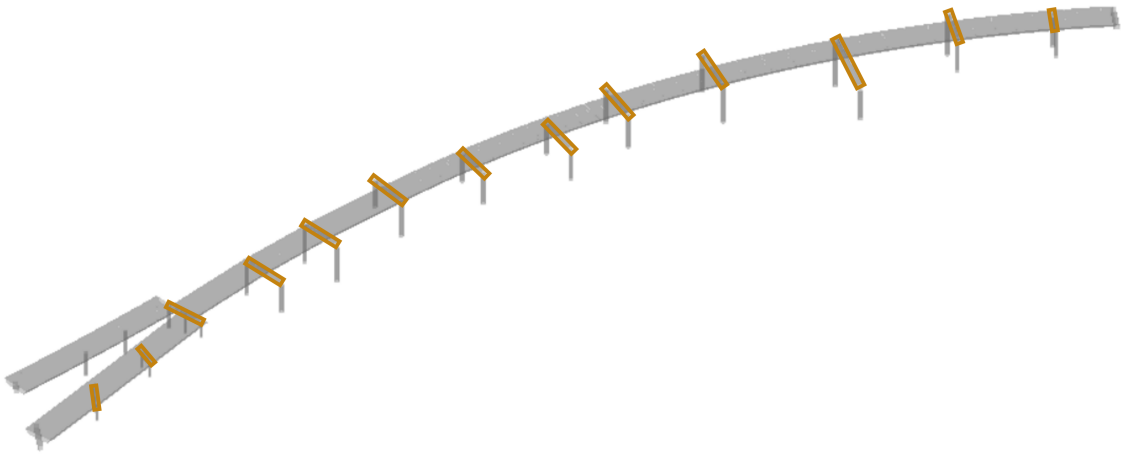


Fig. 40: Hayward Bridge numerical model: Cap beams.

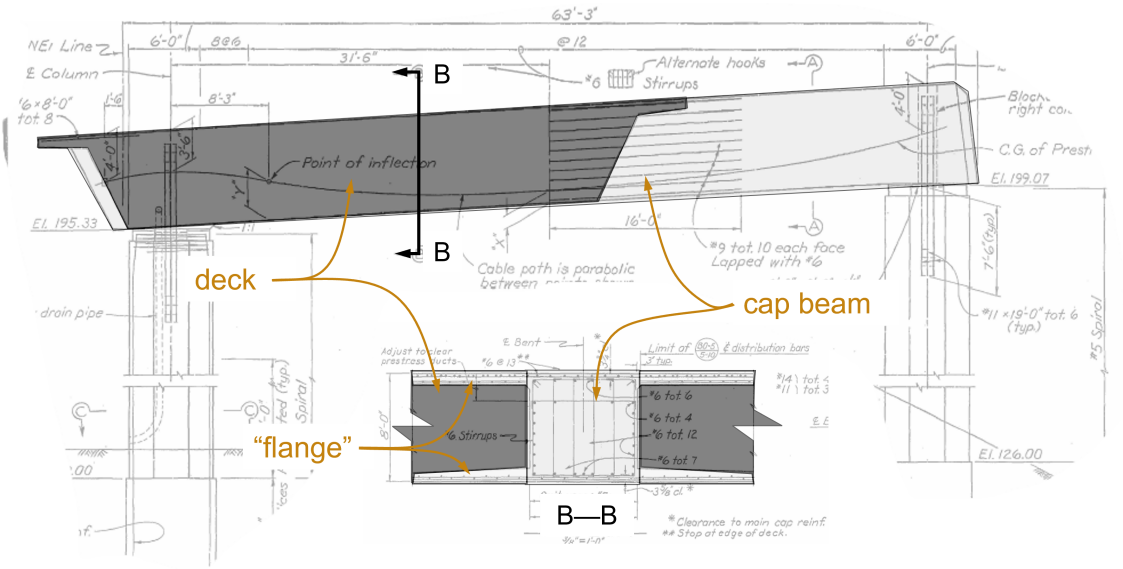


Fig. 41: Hayward Bridge cap beam "flanges" factored into the moment of inertia calculations to represent additional rotational inertia from the superstructure.

8.4 Superstructure

The superstructure (Fig. 42) is modeled as elastic beam-column elements with a consistent mass matrix and cross-sectional area and moment of inertia calculated from the as-built drawings (Fig. 43).

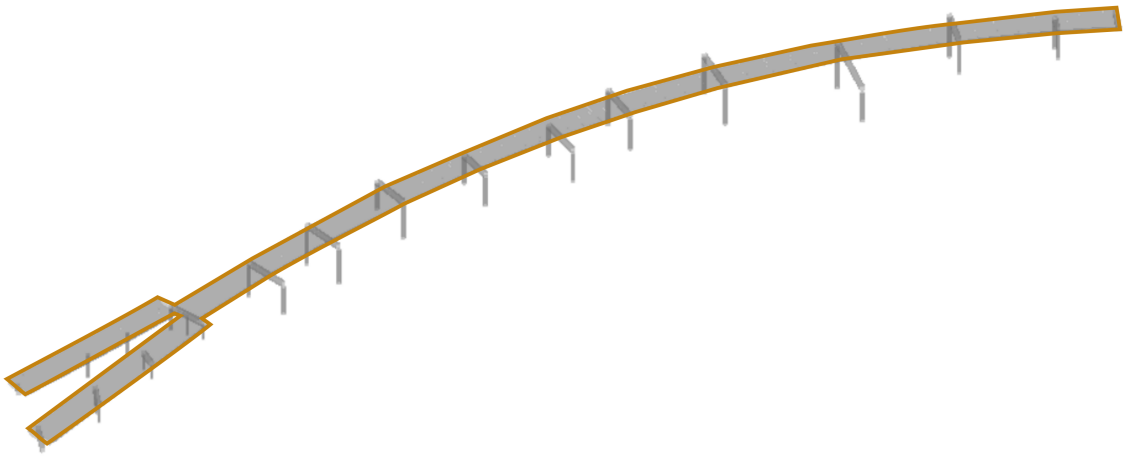


Fig. 42: Hayward Bridge numerical model: Superstructure.

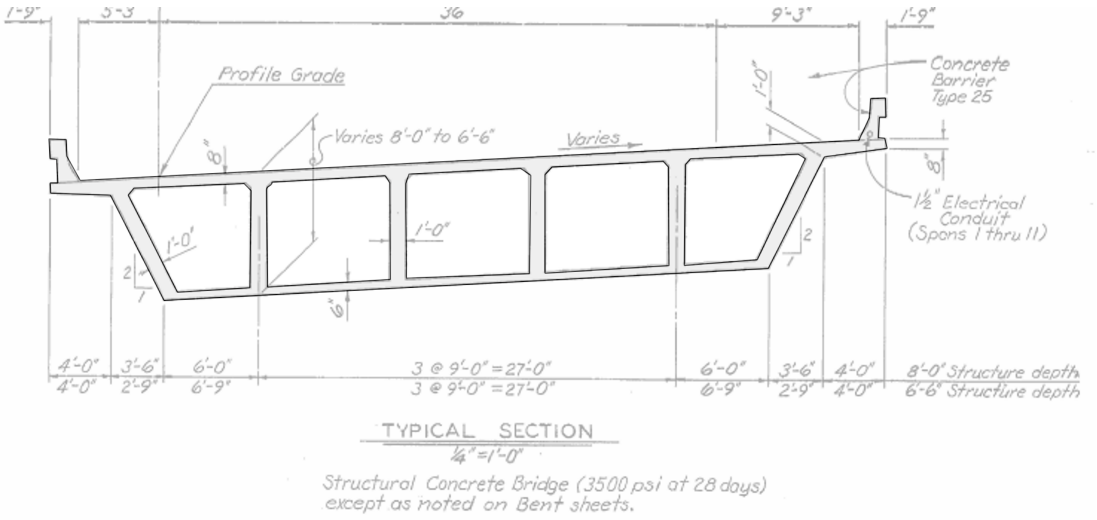


Fig. 43: Hayward Bridge typical superstructure girder cross section.

8.5 In-span Hinges

The in-span hinges (Fig. 44) are modeled as rotational releases in the superstructure with simplified nonlinear force-displacement relationships defining their translational behavior with variables defined in Fig. 45. The model accounts for resistance due to pounding, longitudinal restrainers, elastomeric bearings, and shear keys. The translational in-span hinge models are defined in Fig. 46 and Eqs. (8.2) to (8.4). Fig. 47 describes a simplified model used to improve the numerical stability for the conducted NRHA.

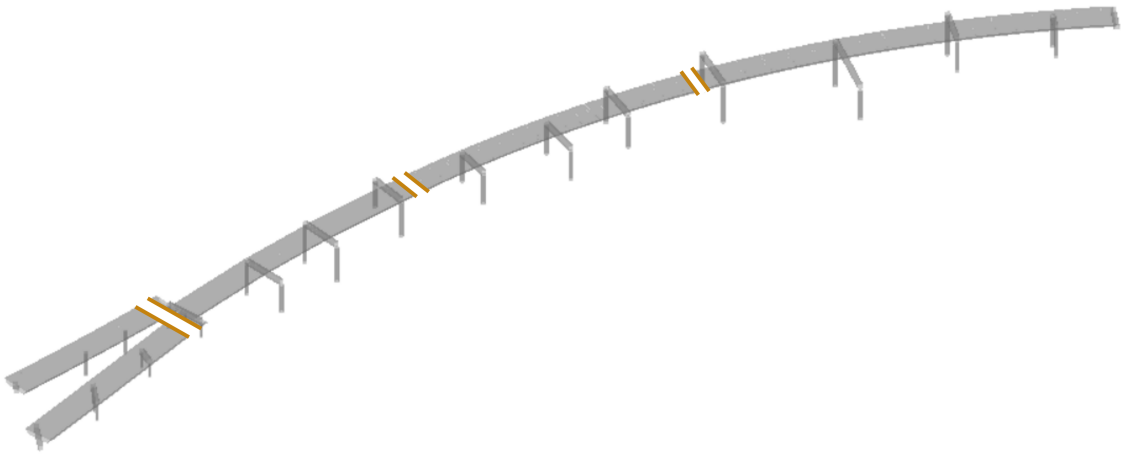


Fig. 44: Hayward Bridge numerical model: In-span hinges.

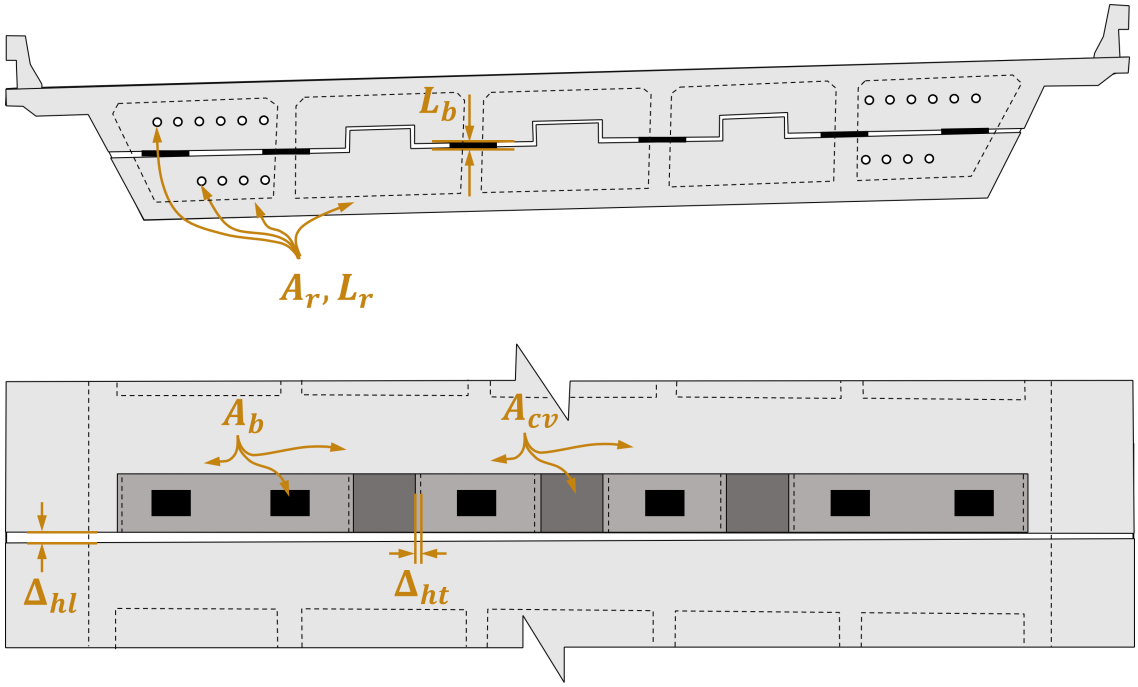
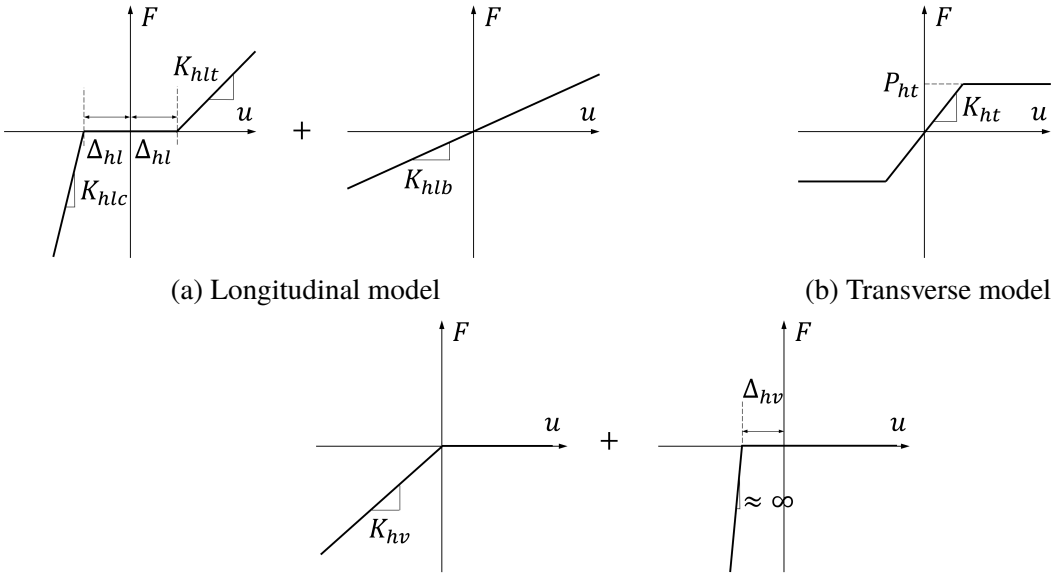


Fig. 45: Hayward Bridge in-span hinge model variables.

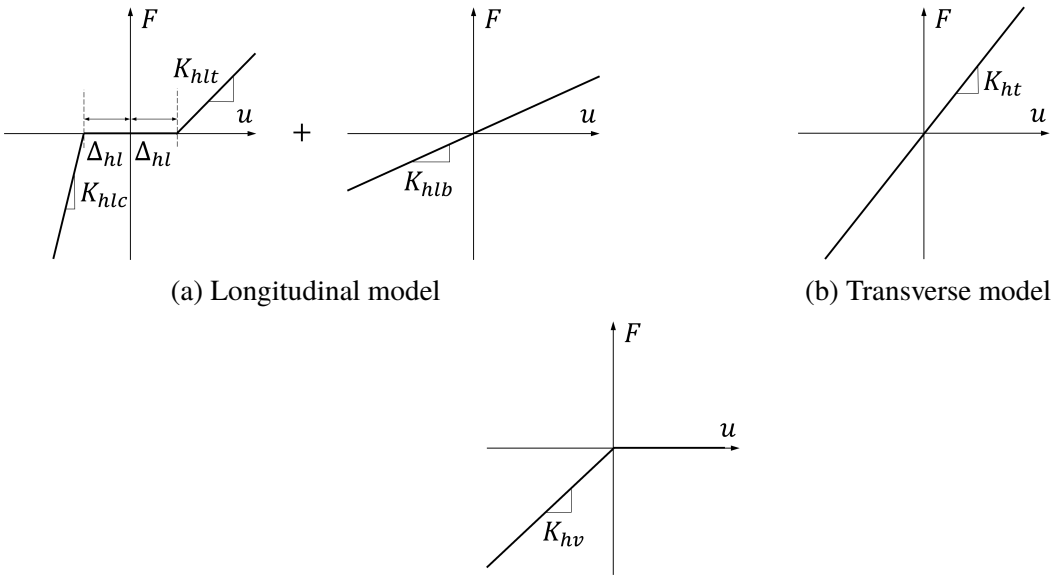


(a) Longitudinal model

(b) Transverse model

(c) Vertical model

Fig. 46: Hayward Bridge in-span hinge model.



(a) Longitudinal model

(b) Transverse model

(c) Vertical model

Fig. 47: Hayward Bridge simplified in-span hinge model.

The longitudinal in-span hinge model is based on a simplification of pounding behavior from [64], the tensile resistance of the longitudinal restrainer rods, and the shear stiffness of the elastomeric bearings. The terms in the longitudinal in-span hinge model are defined by Eq. (8.2) and the variables are defined in Fig. 45.

$$\begin{aligned} K_{hlc} &= 25,000 \text{ kip/in}, \\ K_{hlt} &= \frac{E_r A_r}{L_r}, \\ \Delta_{hl} &= \text{Hinge expansion gap}, \\ K_{hlb} &= \frac{C_{Gh} G_b A_b}{L_b}, \\ G_b &= 0.1 \text{ ksi}, \end{aligned} \tag{8.2}$$

where

K_{hlc} : In-span hinge longitudinal compressive stiffness due to pounding,
K_{hlt} : In-span hinge longitudinal restrainer rod tensile stiffness,
E_r : Restrainer rod elastic modulus,
A_r : Restrainer rod total cross-sectional area,
L_r : Restrainer rod length,
K_{hlb} : In-span hinge bearing shear stiffness,
C_{Gh} : In-span hinge "dimensionless" stiffness coefficient,
G_b : Bearing shear modulus,
A_b : Total horizontal cross-sectional area of bearings, and
L_b : Bearing thickness.

The transverse in-span hinge model is based on [57] and approximates the resistance of the shear keys due to concrete cracking. The terms in the transverse in-span hinge model are defined by Eq. (8.3) and the variables are defined in Fig. 45.

$$\begin{aligned} K_{ht} &= \frac{V_N}{0.05\Delta_{ht}}, \\ P_{ht} &= V_N \times A_{cv}, \\ V_N \text{ (psi)} &= 11.3\sqrt{f'_c \text{ (psi)}}, \end{aligned} \tag{8.3}$$

where

K_{ht} : In-span hinge transverse stiffness,
P_{ht} : In-span hinge transverse ultimate capacity due to concrete cracking strength,
V_N : Concrete cracking strength,
Δ_{ht} : In-span hinge shear key gap, and
A_{cv} : Shear key and superstructure interface area.

In the vertical direction, the in-span hinge model is governed by the compressive resistance of the elastomeric bearings. The terms in the vertical in-span hinge model are defined by Eq. (8.4) and the variables are defined in Fig. 45.

$$\begin{aligned} K_{hv} &= \frac{E_b A_b}{L_b}, \\ E_b &= 0.5 \text{ ksi}, \\ \Delta_{hv} &= 0.6 \text{ in}, \end{aligned} \tag{8.4}$$

where

K_{hv} : Bearing vertical stiffness,
E_b : Bearing elastic modulus,
A_b : Total horizontal cross-sectional area of bearings,
L_b : Bearing thickness, and
Δ_{hv} : Flexible portion of bearing.

8.6 Abutments

The abutments (Fig. 48) are modeled as massless rigid elements with boundary conditions defined at their ends. The ends are free to rotate and their translational behaviors are defined as simplified nonlinear force-displacement relationships, representing the resistance of the backwalls, wingwalls, soil backfill, and elastomeric bearings, with variables defined in Fig. 49. The skew of the abutments reduces the stiffness and maximum capacity in the longitudinal and transverse directions. The shear keys are assumed to crack under design loads and are not included in the model. The translational abutment models are defined in Fig. 50 and Eqs. (8.5) to (8.7). Fig. 51 describes a simplified model used to improve the numerical stability for conducting NRHA.

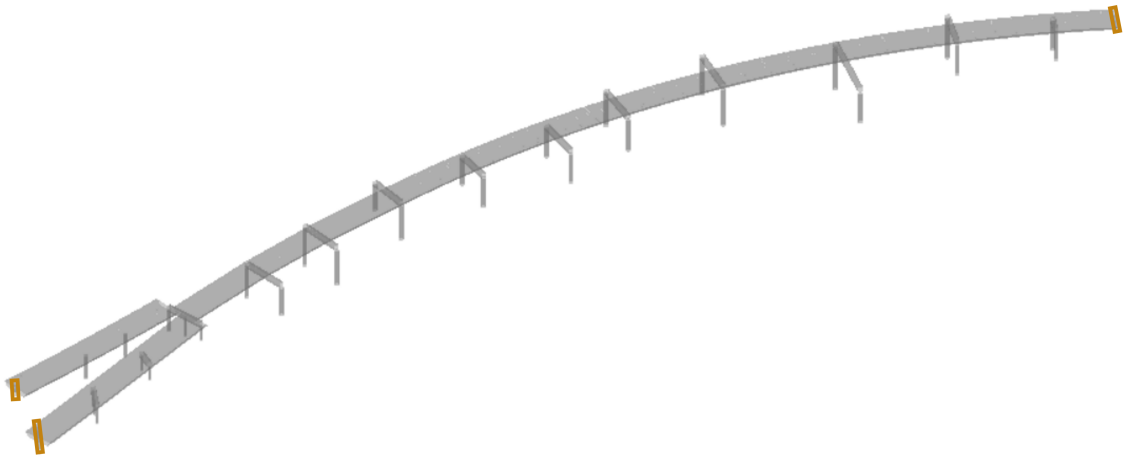


Fig. 48: Hayward Bridge numerical model: Abutments.

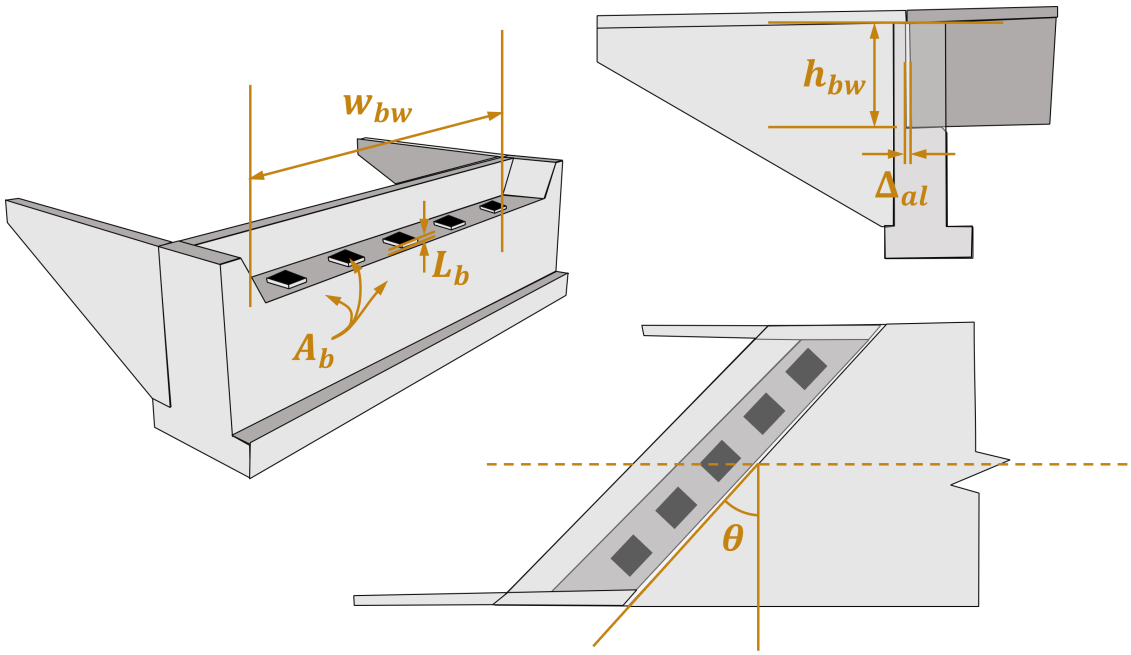
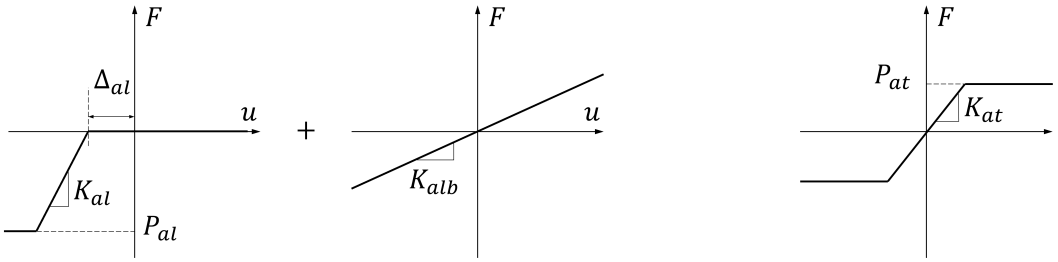
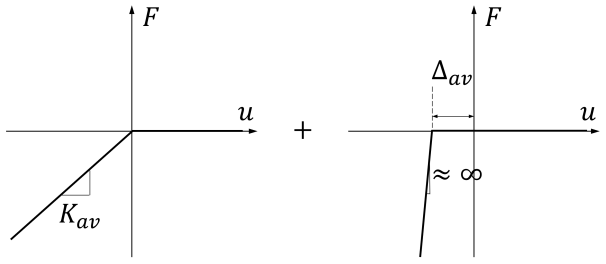


Fig. 49: Hayward Bridge abutment model variables.



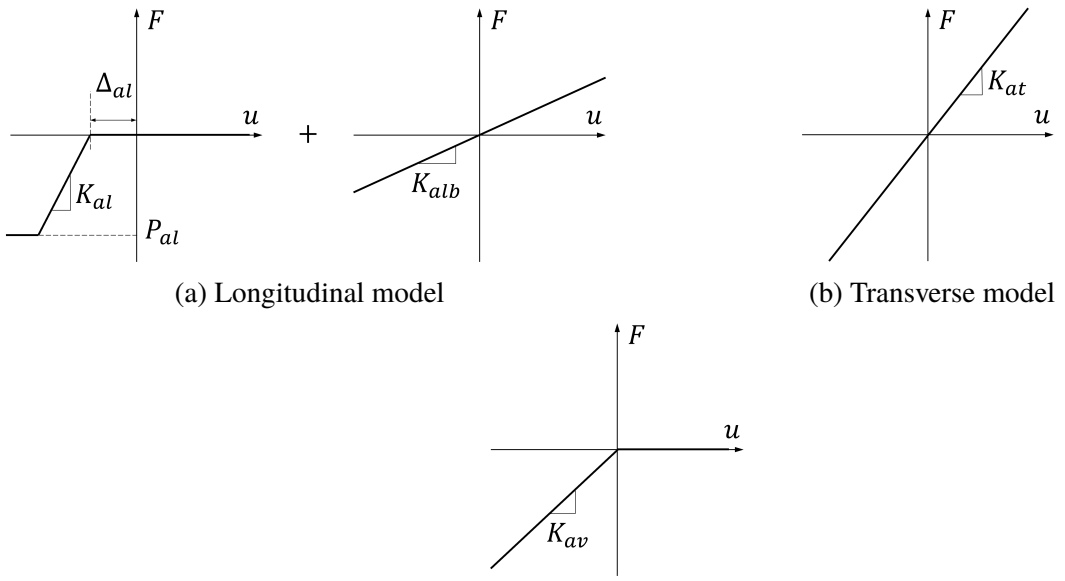
(a) Longitudinal model

(b) Transverse model



(c) Vertical model

Fig. 50: Hayward Bridge abutment model.



(a) Longitudinal model

(b) Transverse model

(c) Vertical model

Fig. 51: Hayward Bridge simplified abutment model.

The longitudinal abutment model is empirically based on Caltrans Seismic Design Criteria (Caltrans) version 2.0, Section 6.3 [15], approximating the backwall resistance acting in parallel with the shear stiffness of the elastomeric bearings. The longitudinal abutment springs are oriented perpendicular to the skew, i.e., perpendicular to the backwall. The terms in the longitudinal abutment model are defined by Eq. (8.5) and the variables defined in Fig. 49.

$$\begin{aligned} K_{al} &= w_{bw}\,(5.5h_{bw} + 20)\,R_{sk}, \\ P_{al} &= w_{bw}\left(\frac{5.5h_{bw}^{2.5}}{1 + 2.37h_{bw}}\right)R_{sk}, \\ \Delta_{al} &= \text{ Abutment expansion gap}, \\ R_{sk} &= e^{\frac{-\theta}{45}}, \\ K_{alb} &= \frac{C_{Ga}G_bA_b}{L_b}, \\ G_b &= 0.1 \text{ ksi}, \end{aligned} \tag{8.5}$$

where

K_{al} : Abutment longitudinal stiffness due to backwall and soil backfill (kip/in),
P_{al} : Abutment longitudinal ultimate capacity due to backwall and soil backfill (kip),
w_{bw} : Width of backwall (ft),
h_{bw} : Height of backwall (ft),
R_{sk} : Skew reduction factor,
θ : Abutment skew angle (degrees),
K_{alb} : Abutment bearing shear stiffness,
C_{Ga} : Abutment stiffness "dimensionless" coefficient,
G_b : Bearing shear modulus,
A_b : Total horizontal cross-sectional area of bearings, and
L_b : Bearing thickness.

The transverse abutment model is based on [43, 52] and approximates the wingwall resistance. The terms in the transverse abutment model are defined by Eq. (8.6).

$$\begin{aligned} K_{at} &= 1/3 \times C_L \times C_W \times K_{al}, \\ P_{at} &= 1/3 \times C_L \times C_W \times P_{al}, \\ C_L &= 2/3, \\ C_W &= 4/3, \end{aligned} \tag{8.6}$$

where

K_{at} : Abutment transverse stiffness due to wingwall and soil backfill
P_{at} : Abutment transverse ultimate capacity due to wingwall and soil backfill
C_L : Abutment wingwall "dimensionless" effectiveness coefficient, and
C_W : Abutment wingwall "dimensionless" participation coefficient.

In the vertical direction, the abutment model is governed by the compressive resistance of the elastomeric bearings. The terms in the vertical abutment model are defined by Eq. (8.7) and the variables are defined in Fig. 49.

$$\begin{aligned} K_{av} &= \frac{E_b A_b}{L_b}, \\ E_b &= 0.5 \text{ ksi}, \\ \Delta_{av} &= 0.6 \text{ in}, \end{aligned} \tag{8.7}$$

where

K_{av} : Bearing vertical stiffness,
E_b : Bearing elastic modulus,
A_b : Total horizontal cross-sectional area of bearings,
L_b : Bearing thickness, and
Δ_{av} : Flexible portion of bearing.

8.7 Soil-Shaft Interaction

With the methodology introduced in Section 4.2.2.3, the Hayward Bridge superstructure accounting for the interaction between soil and drilled shafts is modeled and analyzed [59, 60]. There are two boreholes available near Hayward Bridge structure, described in Table 14. Instrumentation of these boreholes provides the distribution of different soil and rock layers. Historical data of the range of groundwater levels are provided by Caltrans. With these data, a site-specific free-field model is developed to compute the imposed displacement loading at the shafts as multiple-support excitation. A soil-shaft interaction model is developed to compute the shafts' response to the imposed displacement loading. The soil-shaft interaction model uses an array of *p-y* springs arranged in two horizontal directions (N-S and W-E), where the spring nodes are connected by zero-length elements. The *p-y* curves reflect the constitutive model of the corresponding soil type. For Hayward Bridge, the site includes clay and weak rock. The hysteretic model of the soil is developed from the *p-y* curves following the methodology proposed by Rees et al. [72]. The inelastic shaft is modeled as a fiber section [53], where the Hognestad concrete model of [101] (see Section 4.5.1.1) and a bilinear model are employed for the concrete and reinforcing bars, respectively. The used element is force-interpolated, which accounts for nonlinear curvature distribution along the element's length [86]. Finally, the soil-shaft interaction model is integrated into the global model of the bridge by enforcing an equal-degree-of-freedom constraint between the head node of each shaft pile and the bottom node of the corresponding column, as illustrated in Fig. 52.

Table 14: Characteristics of the geotechnical downhole instrumentation at Hayward Bridge.

Array	Latitude	Longitude	Depth (m)	Distance to Bridge (m)	Depths below ground surface (m)
East	37.6896N	122.0962W	91	30	3, 0, 9, 30, 91
West	37.6887N	122.1074W	91	625	12, 0, 5, 9, 18, 37, 91

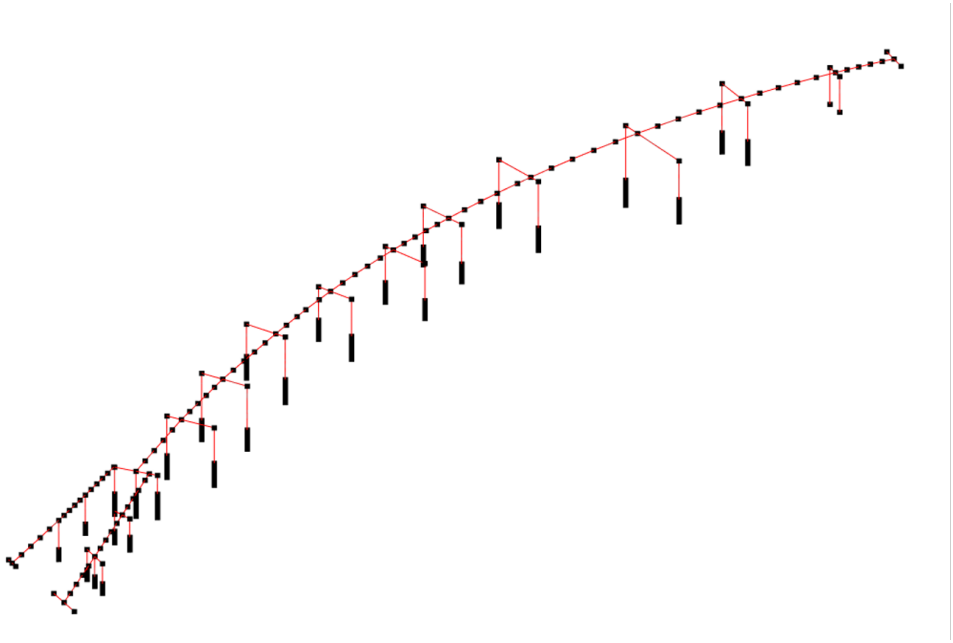


Fig. 52: Rendered model of Hayward Bridge with soil-shaft interaction.

8.8 Solution Procedure

The analysis begins by calculating the response due to gravity loads using a Newton Raphson solver with load control of ten equal increments of constant load. A full general (storing the entire matrix for the system of equations) LAPACK[12] solver is used to perform an eigenvalue analysis to determine the period and shape of the first eight modes of vibration. Subsequently, the dynamic response to ground motion excitation is calculated by nonlinear time-stepping through the record using a Newmark-Beta integrator with parameters $\alpha = 0.5$ and $\beta = 0.25$ and a Newton Raphson solver. Damping options include Rayleigh or modal damping of up to the first eight modes, with damping ratios of any value between 1% and 5%. Element deformations are determined by P-Delta or linear geometric transformations.

8.9 Appendix 8.A: Modeling Glossaries

8.9.1 Symbols: Material Properties

E_c concrete compressive elastic modulus. 65

E_{sh} steel initial strain hardening tangent. 66

E_s steel tensile elastic modulus. 66

G_c concrete shear modulus. 65

ϵ_{c0} unconfined concrete strain at maximum strength. 65

[12] "Linear Algebra Package", a standard library for numerical linear algebra.

ϵ_{sh} steel strain at initial strain hardening. 66

ϵ_{sp} unconfined concrete crushing (spalling) strain. 27, 65

ϵ_{su} steel strain at peak stress. 27, 66

ϵ_t unconfined concrete tensile cracking strain. 27

ν_c concrete Poisson's ratio. 65

f'_{ce} unconfined concrete compressive strength. 65

f_u steel ultimate strength. 66

f_y steel yield strength. 66

8.9.2 Symbols: Modeling Properties

A_b total horizontal cross-sectional area of bearings. 75, 76, 79, 80

A_{cv} shear key and deck interface area. 75

A_r restrainer rod total cross-sectional area. 75

A_{sp} area of transverse spiral reinforcing bar. 67

C_L abutment wingwall effectiveness coefficient. 79

C_W abutment wingwall participation coefficient. 79

D_{tr} diameter of transverse reinforcing spiral. 67

E_b bearing elastic modulus. 76, 80

E_r restrainer rod elastic modulus. 75

G_b bearing shear modulus. 75, 79

K_{alb} abutment bearing shear stiffness. 79

K_{al} abutment longitudinal stiffness due to backwall and soil backfill. 79

K_{at} abutment transverse stiffness due to wingwall and soil backfill. 79

K_{av} abutment bearing vertical stiffness. 80

K_{hlb} in-span hinge bearing shear stiffness. 75

K_{hlc} in-span hinge longitudinal compressive stiffness due to pounding. 75

K_{hlt} in-span hinge longitudinal restrainer rod tensile stiffness. 75

K_{ht} in-span hinge bearing transverse stiffness. 75

K_{hv} in-span hinge bearing vertical stiffness. 76

L_b bearing thickness. 75, 76, 79, 80

L_r restrainer rod length. 75

P_{al} abutment longitudinal ultimate capacity due to backwall and soil backfill. 79

P_{at} abutment transverse ultimate capacity due to wingwall and soil backfill. 79

P_{ht} in-span hinge transverse ultimate capacity due to concrete cracking strength. 75

R_{sk} abutment skew reduction factor. 79

V_N concrete cracking strength. 75

Δ_{al} abutment expansion gap. 79

Δ_{av} flexible portion of abutment bearing. 80

Δ_{hl} in-span hinge expansion gap. 75

Δ_{ht} in-span hinge shear key gap. 75

Δ_{hv} flexible portion of in-span hinge bearing. 76

ϵ_{cu} concrete crushing strain. 27

ρ_{tr} transverse reinforcement ratio. 67

θ abutment skew angle. 79

h_{bw} height of abutment backwall. 79

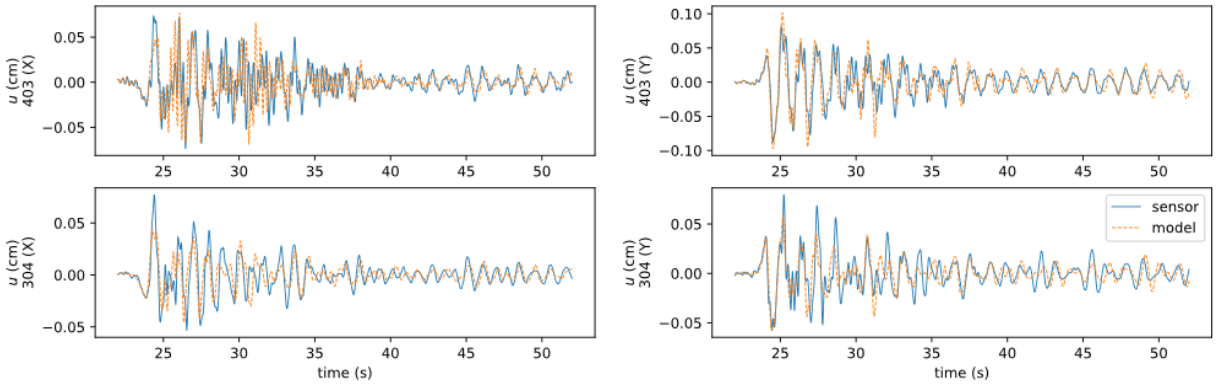
s_{tr} vertical spacing of transverse reinforcing spiral. 67

w_{bw} width of abutment backwall. 79

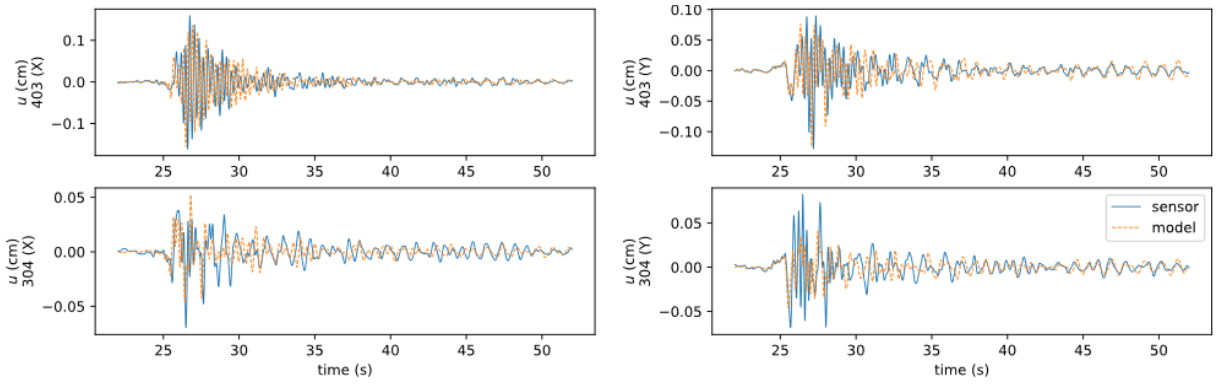
Chapter 9

Example Output

Hayward Bridge with fixed base The displacement responses at the top of Bent 3 (model node 304, sensor channels 14 and 15) and the top of Bent 4 North column (model node 403, sensor channels 19 and 20) are shown for two events, the Mw 4.4 Berkeley event on 01/04/2018 and the Mw 3.9 San Lorenzo event on 06/29/2021 (Fig. 53). The figure shows reasonable comparison between the results obtained from the computational model based on physics (referred to as "model") and the data-driven model using sensor data (referred to as "sensor").



(a) Mw 4.4 Berkeley event on 01/04/2018



(b) Mw 3.9 San Lorenzo event on 06/29/2021

Fig. 53: Displacement response output at two locations for two events of Hayward Bridge.

Hayward Bridge with soil-shaft interaction Applying the Mw 4.4 Berkeley event on 01/04/2018, the natural periods of the whole system are 1.20 and 0.96 seconds for transverse and longitudinal directions, respectively, which are close to the values identified by the system identification methods, i.e., 1.10 and 0.93 seconds, respectively[13]. The displacement responses at the top of Bent 3 (model node 304, sensor channels 14 and 15), the top of Bent 4 North column (model node 403, sensor channels 19 and 20), and the bottom of Bent 4 South column (model node 407, sensor channels 24 and 25) are shown in Fig. 54. In comparison with the response of the fixed-base model, Fig. 53a, the initial magnitudes of displacement (i.e., from about 20 to 30 seconds) appear closer to the sensor recording. Note that Fig. 53 does not include the response for model node 407 because the motion at the base of the columns are the input to the fixed-base model, i.e., the model response is exactly the sensor response at model node 407.

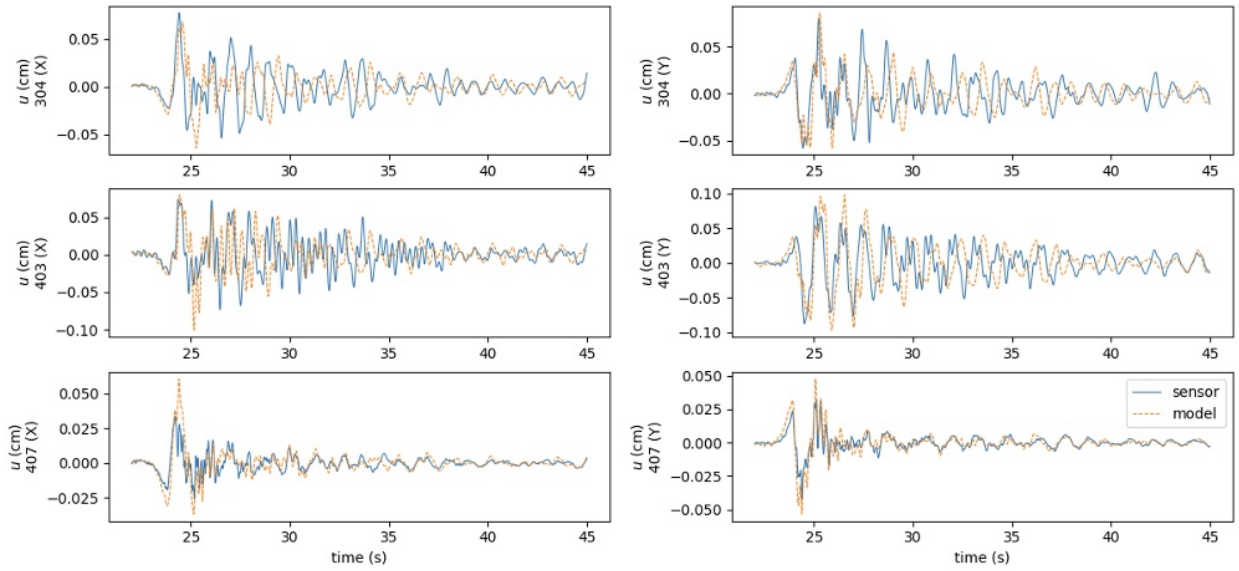
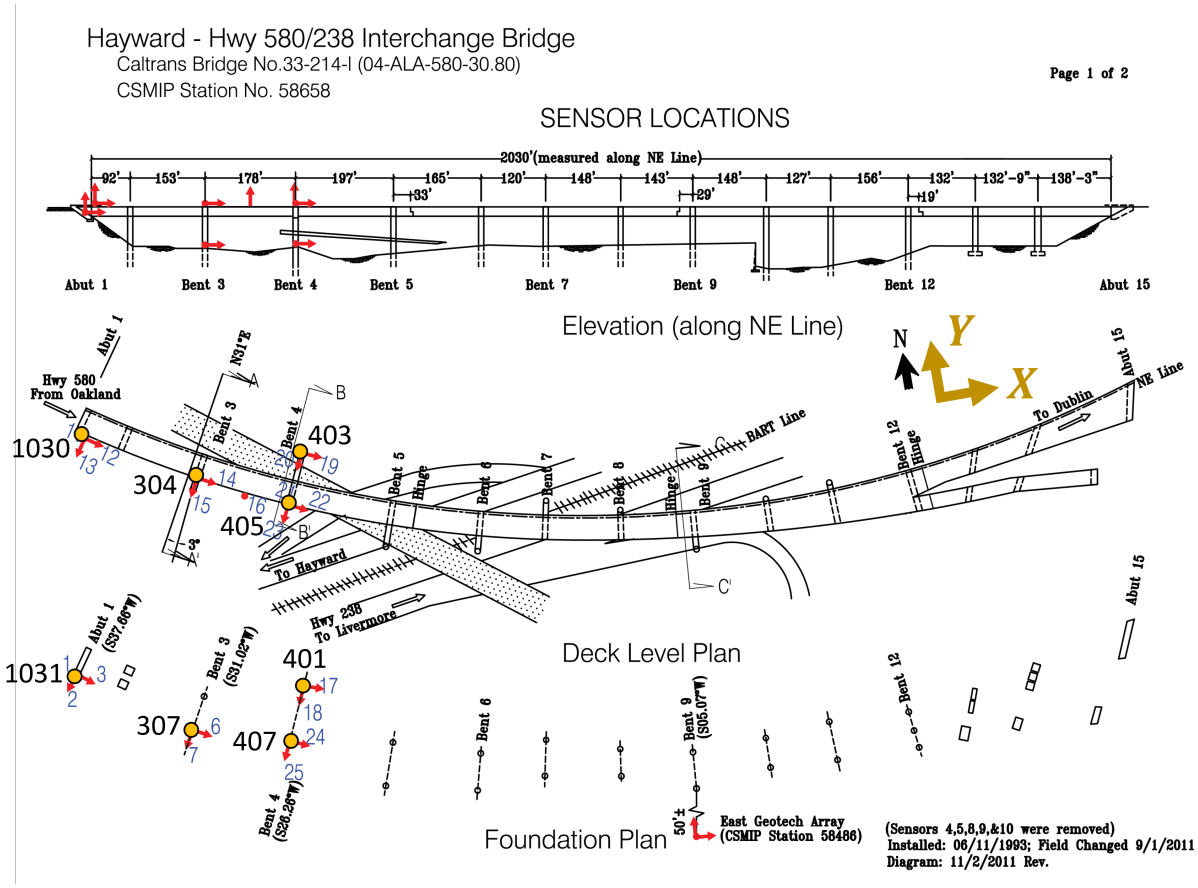


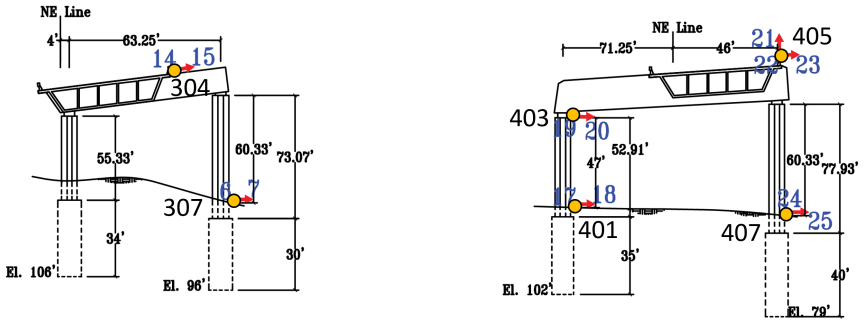
Fig. 54: Displacement response output at three locations for 2018 Mw 4.4 Berkeley earthquake of Hayward Bridge with soil-shaft interaction.

The locations of the responses, discussed above, as well as the "X" and "Y" directions, are shown in Fig. 55. The output response comparisons with sensor data give some confidence in the accuracy of the developed digital twin of Hayward Bridge. Further validations of the developed model for Hayward Bridge and other bridges in *BRACE*^{2} platform will be continuing in the future.

[13]It is noted that these values using fixed-base model are 0.96 and 0.76 seconds, i.e., shorter than those from the bridge model considering the soil-shaft interaction, which is expected.



(a) Plan



Section A - A'
at Bent 3

Section B - B'
at Bent 4

(b) Elevation

Fig. 55: Model node locations and "X" and "Y" directions of Hayward Bridge (source: CGS with modifications).

Part IV

Technical Details

Part IV collects technical documentation which will aid in the successful transfer, deployment, and long-term operation of the health monitoring platform. The target audience includes system administrators (Chapters 10 and 11), future researchers (Chapters 12 and 13), and source code maintainers (Chapter 14).

Chapter 10

The Web Application

The *BRACE*^{2} platform is a dynamic web application for performing real-time structural health monitoring. In Phase I of the *BRACE*^{2} project, the platform operated with a "software as a service" (SaaS) model, where UC-Berkeley has provided the services of a developer and system administrator. However, in the future, it may be desirable to self-host the platform. In Section 10.1, details related to hosting and administering the platform are discussed, and in Section 10.2, the software architecture and source code layout are described. The source code for the platform is currently hosted in a private Git repository on GitHub[14].

10.1 Administration

10.1.1 Hosting

The software requirements are very flexible. The host server should be capable of supporting at least a standard Python 3.8 environment, but a newer version is recommended. The application has been successfully run on both cloud-hosted servers and a local server with a variety of operating systems and databases. All of these would be adequate options for Caltrans. On the Heroku cloud hosting platform, with current demands, the application can easily be hosted for under $20/month.

Both *SQLite* and *PostgreSQL* databases have been successfully used between various deployments, and any database supported by the Django framework should also be acceptable. Currently, Django officially supports *PostgreSQL*, *MariaDB*, *MySQL*, *Oracle*, and *SQLite*[15]. Various deployment environments have been used successfully to host the platform. The Heroku stack that is currently hosting the instance at `https://brace2.herokuapp.com` is summarized in Table 15.

10.1.2 Users and Groups

Users of the application are divided into three *groups*[16], based on their required privileges (see also Figs. 56 and 57):

[14]For details about Git repositories and GitHub, refer to `https://github.com/`.

[15]More information on database support can be found in the Django database documentation page: `https://docs.djangoproject.com/en/5.0/ref/databases/`.

[16]The term *group* should not be confused with the usual *working groups* of projects. Here, the term is used in a standard I.T. sense to refer to a collection of users that are granted a common set of privileges.

Table 15: Heroku stack currently employed in serving the *BRACE*^{2} platform.

Component	Product
Dyno	`eco`
Datastore	`postgresql-asymmetrical-28930`
Gateway	`gunicorn`

Group 1 consists of agents (currently, only California Geological Survey "CGS" and its California Strong Motion Instrumentation Program "CSMIP", which hosts and manages the Center for Engineering Strong Motion Data "CESMD") that provide ground motion data in real-time. This is the only group that will most likely need access from outside of the host's local network. This user group makes `POST` requests over a secured API (Chapter 11) that is provided by the `brace2.events` sub-application. These posts contain a ground motion archive file that must be stored temporarily on the disk while the *predictors* are executed. Any data from these files that is needed for long-term operations on the platform is parsed out and stored as an entry in the `Event` database table.

Group 2 consists of users (likely engineers within the host's local network, i.e., Caltrans) that will be granted privileges to configure inventory predictors. These users will interact with the platform over a standard web browser by accessing HTML[17] pages that are served by the `brace2.inventory` and `brace2.predictors` sub-applications. These actions may result in insertions and deletions in the `Asset` and `Predictor` database tables.

Group 3 consists of users with the same *read* privileges as **Group 2**, but no *write* privileges. The interactions from this group should not result in any changes to the database.

10.1.3 Initializing the Pilot Models

As part of Phase I of the *BRACE*^{2} project, **21** assets have been given partial digital twins, and between these a total of **93** predictors have been configured. Once the platform is deployed, it can be populated further by performing the following steps:

0) Initialize environment: Make sure that the proper Python environment is activated in the working shell and the current working directory is the root of the `HealthMonitoringPlatform` repository.

1) Install initialization dependencies: The initialization process requires a few additional dependencies. These can be installed by running the following command:

```
python -m pip install -Ur init/requirements.txt
```

[17] HyperText Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser.

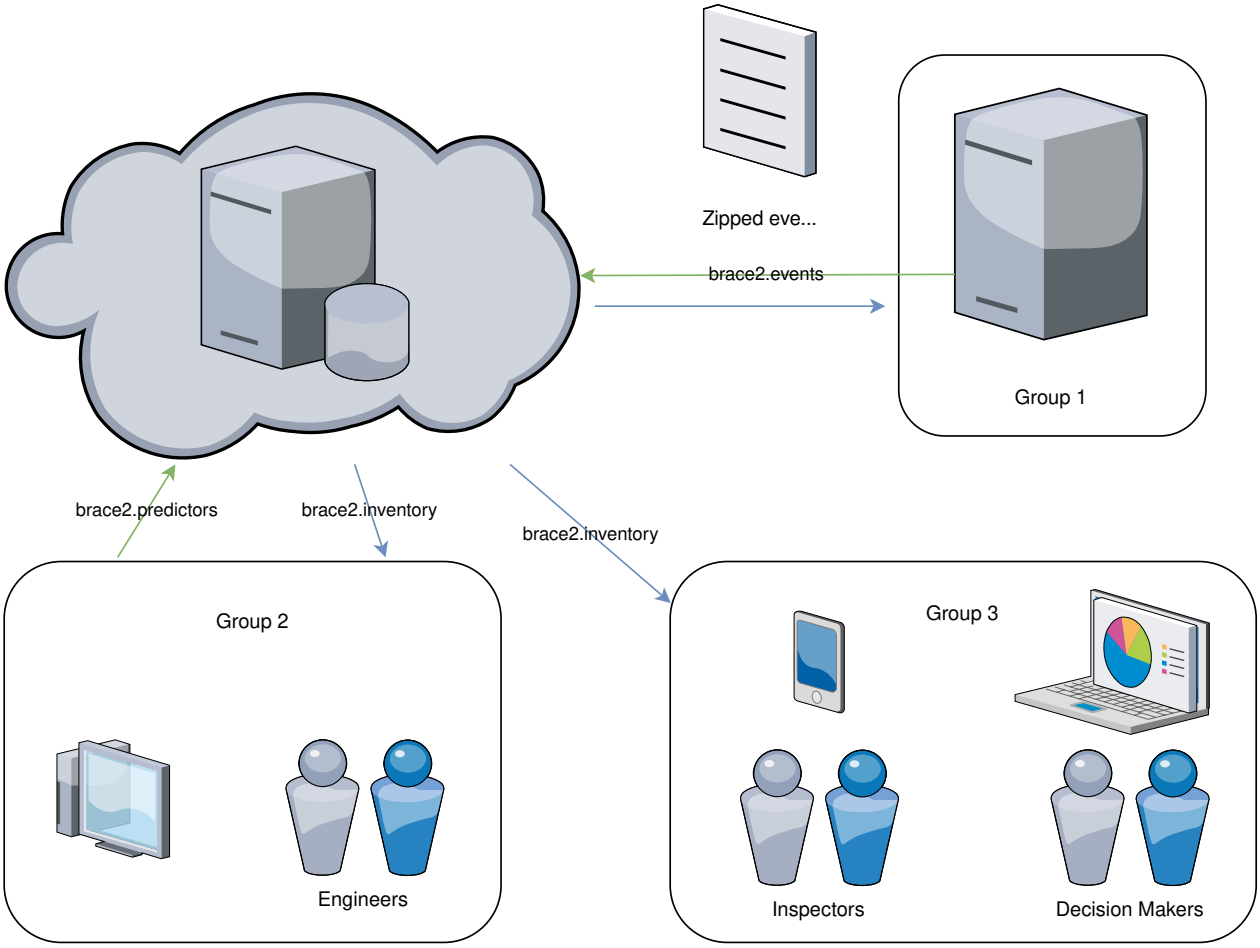


Fig. 56: *BRACE*^{2} platform user access.

2) Configure partial twins (optional): Assets and predictors will be configured from the files `init/bridges.py` and `init/calid.py`. These can be modified or extended before proceeding.

3) Download CGS data: The following script will execute the file `init/bridges.py` and pull CGS metadata for the bridges listed there:

```
python init/getCGSData.py  > data/cgs_data.json
```

4) Create assets (bridges): Run the following command to create an `Asset` in the database for each of the bridges listed in `bridges.py`:

```
python manage.py shell < init/init_assets.py
```

5) Create predictors: Run the following command to create a `PredictorModel` in the database for each of the predictors listed in `bridges.py`:

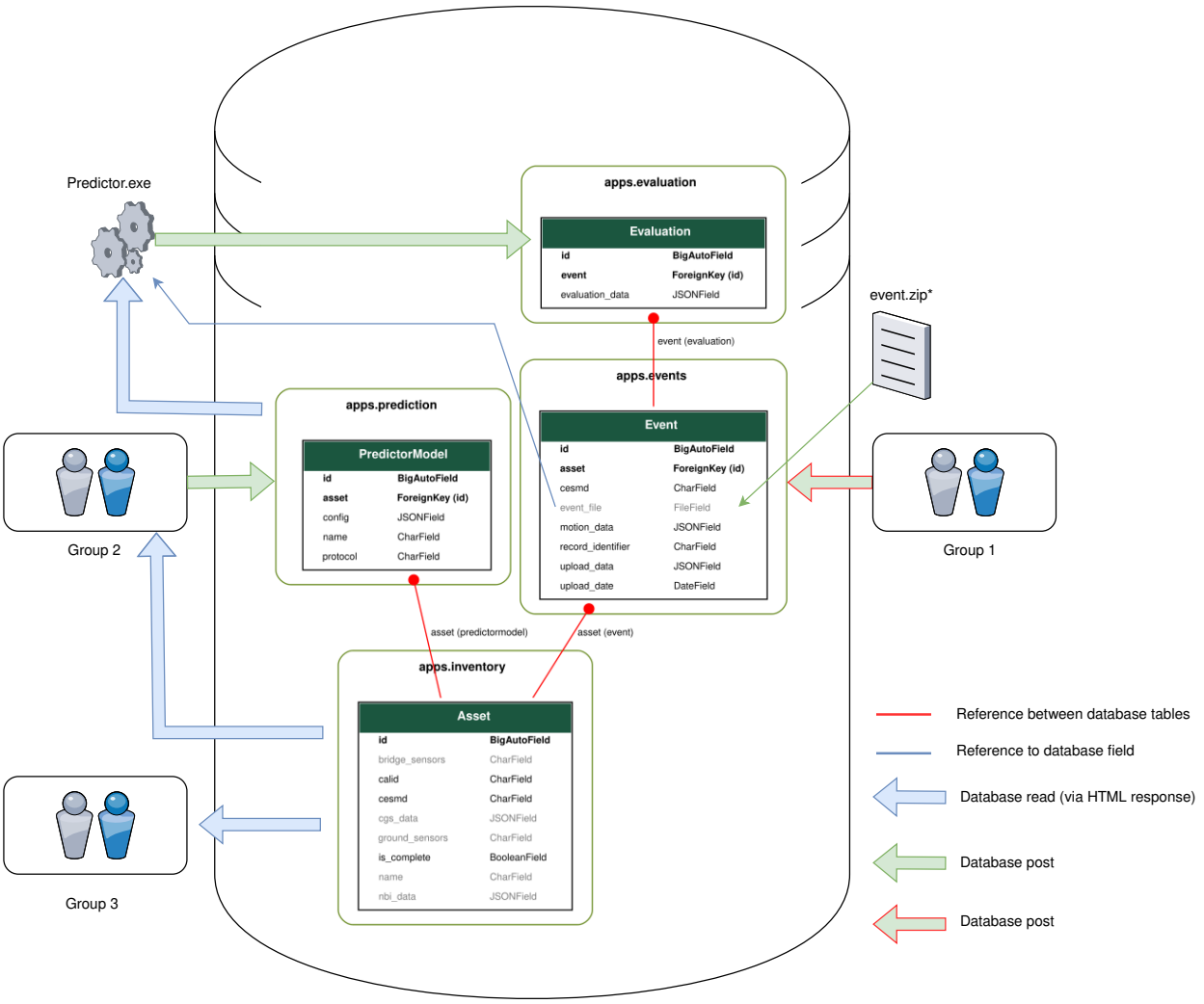


Fig. 57: *BRACE*^{2} platform database implementation details.

```
python manage.py shell < init/init_predictors.py
```

Once this setup is completed, users can be created with credentials in the standard manner using the Django admin page. For more information, refer to the Django documentation: https://docs.djangoproject.com/en/5.0/ref/.

10.2 Architecture

The *BRACE*^{2} web application is primarily built on top of the Django Python framework. Django is an open-source framework that is released under a 3-clause BSD license[18] and has been actively maintained by the Django Software Foundation since 2008. The web application's source code is organized into several modules. With the exception of the `core` module, these are all contained in the `apps/` directory of the main repository. The `core` module contains basic server configuration and links the remaining application modules together to form a whole application. The application modules are responsible for implementing the main abstractions and functionality of the platform. The health monitoring functionality of the platform is implemented across four main *monitoring* modules whose architecture is developed in Section 10.2.1. The remaining modules are discussed in Section 10.2.2.

10.2.1 Monitoring Modules

The structural health monitoring functionality of the *BRACE*^{2} platform is organized into four *modules*. Each of these modules is responsible for one table in the database. These modules are listed in Table 16 alongside their respective database tables.

Table 16: Structural health monitoring modules of the *BRACE*^{2} platform.

Module	Database Table
`inventory`	`Asset`
`events`	`Event`
`evaluation`	`Evaluation`
`prediction`	`PredictorModel`

10.2.1.1 `inventory` Module

The `inventory` module implements the basic concept of an asset and essentially provides the *digital twin* with its identity. This module is centered around the creation and management of entries in the `Asset` database table (Fig. 57), which is configured in `inventory/models.py`. At the conclusion of Phase I of the project, this database table is managed manually at the system administration level. The `inventory` module imports the `prediction` module so that an `Asset` entry may collect all `PredictorModel` entries which point to it (see Section 10.2.1.4).

10.2.1.2 `events` Module

The `events` module enriches the *digital twins* of the `inventory` module with the ability to *respond* to real-life events. This module is centered around the creation and management of entries in the `Event` database table. The creation of `Event` entries is triggered by HTTP[19] **POST** requests

[18] The 3-clause BSD license is a descendant of the original BSD license which was named after the Berkeley Software Distribution, a Unix-like operating system.

[19] HyperText Transfer Protocol (HTTP) is an application layer protocol that loads web pages from hypertext documents.

to the `event` API (Chapter 11). The `events` module directly interacts with the `inventory` and `evaluation` modules. When a **POST** request is made, the request handler creates an entry in the `Event` table and then performs the following tasks:

1. Extract metadata from the HTTP request to populate the new `Event` table entry.
2. Use the `cesmd` field extracted metadata to identify the unique `Asset` entry with a matching `cesmd` from the the `inventory.Asset` table.
3. Invoke the `evaluation.Evaluation::create` factory method to perform the following:
    - Create an entry in the `evaluation.Evaluation` table.
    - Instantiate an instance of the `EvaluationDaemon` class to manage the population of the new entry (see Section 10.2.1.3 below).

10.2.1.3 `evaluation` Module

The `Evaluation` table stores the results of all evaluations. The `evaluation.daemon` submodule implements the helper `EvaluationDaemon` class, which is responsible for scheduling predictors to populate `Evaluation` entries. This allows a schedule to be crafted which is uniquely tailored for both the `Asset` and `Event` properties. An `EvaluationDaemon` instance is created by the static `create` method of the `Evaluation` model class, which is generally invoked through the `events` module in response to the creation of a new `Event` entry.

10.2.1.4 `prediction` Module

The `prediction` module implements the predictor abstraction. The `PredictorModel` table is configured in `prediction/models.py` and stores the configuration for the predictors that will be invoked to populate the `Evaluation` entries for a particular `Asset`. `PredictorModel` entries are created by users through the **Configure Predictors** page of an asset. The `prediction.predictors` submodule implements the abstract `PredictorDaemon` class and its subclasses, which abstract away the implementation details for various predictors.

10.2.2 Remaining Modules

Two additional modules implement the following miscellaneous application features:

- `documents`: This module serves various PDF documents that have been produced through the *BRACE*^{2} project.
- `site`: This module implements the login and registration pages and serves the main landing page.

Chapter 11

Event API

11.1 Motion API

The *BRACE*^{2} Motion API provides a controlled point of contact for sharing and managing ground motion records in real-time. The primary component of the tool is a server-hosted REST API through which all motion data passes. The Motion API has been designed to adhere to standard web protocols, allowing the endpoint to be accessed in a uniform manner from virtually any programming language (see Section 11.3 for examples in Python, C, Bourne Shell, Go, Rust, and more). This API returns JSON-serialized ground motion data adhering to the schema that is developed in the following sections. Cryptographic token authentication is being used to authenticate requests.

11.2 Design

File uploads are sent as HTTP POST operations with a MIME (Multipurpose Internet Mail Extensions) type of `multipart/form-data` [RFC 2046]. These uploads are handled by the REST API that is integrated directly into the `events` application of the health monitoring platform. This application listens to a server port for API requests and fulfills these requests by interacting with the database.

11.3 API Examples

This section collects examples that demonstrate interactions with the API from various programming languages. Note that placeholder credentials and hostname have been hard-coded into the programs for the sake of demonstration, but this should not be the case in practice.

11.3.1 Get all motions

11.3.1.1 Python

The following Python script uses the `requests` module to perform an HTTP request.

```python
import sys
import json
import requests
# Framework parameters
# ---------------------------------
username, password = ["brace2"] * 2
host = "http://localhost:8000"

# Setup API request
# ---------------------------------
headers = {
    "Content-Type": "Application/JSON",
}

response = requests.get(
    host + "/api/events/", headers=headers, auth=(username, password)
)
# print(json.loads(response.text))
print(response.json())
```

11.3.1.2 Shell (`curl`)

The following Shell script uses the `GET` method of the `curl` command-line tool to perform an HTTP request to a server.

```sh
#!/bin/sh
hostname="localhost:8000"
username="brace2"
password="brace2"
event_id="$1"

curl -X GET \
  -H 'Content-Type:Application/JSON' \
  -u $username:$password \
  $hostname/api/events/

printf "\n"
```

11.3.2 Get a single motion

11.3.2.1 C

The following C program uses the `libcurl` library to perform an HTTP request to a server.

```c
#if 0
EXEC=${0%.*}
test -x "$EXEC" || clang "$0" -lcurl -o "$EXEC"
exec "$EXEC"; exit
{
#endif
#include <stdio.h>
#include <stdlib.h>
#include "curl/curl.h"

int main(void) {
  int status = EXIT_SUCCESS;

  const char
    *username = "brace2",
    *password = "brace2",
    *hostname = "localhost:8000";
  char *url = strfmt("https://%s:%s@%s/api/events/",
    username,password,hostname
  );

  CURL *curl;
  char buffer[CURL_ERROR_SIZE];
  if ((curl = curl_easy_init()) != NULL) {
          curl_easy_setopt(curl, CURLOPT_URL, url);
          curl_easy_setopt(curl, CURLOPT_FOLLOWLOCATION, 1);
          curl_easy_setopt(curl, CURLOPT_ERRORBUFFER, buffer);
          if (curl_easy_perform(curl) != CURLE_OK) {
                  fprintf(stderr, "%s\n", buffer);
                  return EXIT_FAILURE;
          }
          curl_easy_cleanup(curl);
  }
  free(url);
  return EXIT_SUCCESS;
}
#if 0
}
#endif
```

11.3.2.2 Go

The following program is written in the *Go* programming language and uses the `net/http` module of the standard library to perform the HTTP request.

```go
// Claudio Perez
package main

import (
        "fmt"
        "io/ioutil"
        "log"
        "net/http"
)

func main() {
  hostname := "localhost:8000"
  username := "brace2"
  password := "brace2"
  endpoint :=  fmt.Sprintf("%s/api/events/",hostname)

  client := &http.Client{}
  req, err := http.NewRequest("GET", endpoint, nil)

  if err != nil {log.Fatal(err)}

  req.Header.Set("Content-Type", "Application/JSON")
  req.SetBasicAuth(username, password)
  resp, err := client.Do(req)
  if err != nil {
        log.Fatal(err)
    }
  defer resp.Body.Close()
  bodyText, err := ioutil.ReadAll(resp.Body)
  if err != nil {
   log.Fatal(err)
  }
  fmt.Printf("%s\n", bodyText)
}
```

11.3.3 Upload a new motion

In the following examples, a string-type variable named `event_file` is used to store the path to a ground motion file that is to be uploaded from disk, i.e, from a hard drive.

11.3.3.1 Python

```python
import sys
import json
import requests
# Payload parameters
# ---------------------------------
name = sys.argv[1]
event_file = "/path/to/data/58658_007_20210426_10.09.54.P.zip"
# Framework parameters
# ---------------------------------
username, password = ["brace2"] * 2
hostname = "http://localhost:8000"
# Setup API request
# ---------------------------------
headers = {
    "Content-Type": "multipart/form-data",
}
files = {
    "event_file": (event_file, open(event_file, "rb")),
    "name": (None, name),
}
response = requests.post(
    hostname + "/api/events/", headers=headers, files=files,
    auth=(username, password)
)
print(json.loads(response))
```

11.3.3.2 Shell (curl)

```shell
#!/bin/sh
username="$MOTION_API_USERNAME"
password="$MOTION_API_PASSWORD"
hostname="$MOTION_API_HOSTNAME"
filename="$1"

curl -X POST -H 'Content-Type:multipart/form-data' \
  -u $username:$password \
  -F "event_file=@$filename" \
  "$hostname/api/events/"

printf "\n"
```

Chapter 12

sdof

The sdof package is a highly-optimized library for single-degree-of-freedom integration that has been developed for efficient real-time computations within the *BRACE*^{2} platform. The sdof library is primarily implemented in standard C and interfaces are provided for both Python and Javascript.

12.1 Python Package

The sdof Python package is published on the Python Package Index (PyPI) and can be installed simply by executing the command:

```
pip install sdof
```

The following functions are exposed by the package to interact with the C library:

```
sdof.integrate(f, dt: float, k: float, c: float, m: float, u0: float =
               0.0, v0: float = 0.0, out=None, fy: float = None, beta:
               float = 0.25, gamma: float = 0.5, alpha_m: float = 1.0,
               alpha_f: float = 1.0)
```

This function integrates scalar differential equations of the form:

$$m\ddot{u} + c\dot{u} + ku = f(t)$$

for constant coefficients m, c, and k.

Parameters

- f (list) – Loading
- dt (float) – Time step
- k (float) – Stiffness
- c (float) – Damping
- m (float) – Mass
- u0 (float, optional) – Initial displacement
- v0 (float, optional) – Initial velocity

- `out` (array, optional) – Array to store outputs
- `alpha_m` (float, optional) – Integrator α_m parameter
- `alpha_f` (float, optional) – Integrator α_f parameter
- `beta` (float, optional) – Newmark β parameter
- `gamma` (float, optional) – Newmark γ parameter

Integration is carried out using a Generalized $\alpha-$scheme by calling the C function **sdof_integrate**().

```
sdof.spectrum(f, dt, damping, periods=None, interp=None, threads: int=
              None, **kwds)
```

This function is a wrapper around the C function `sdof_spectrum()`.

Parameters

`f` (list), dt (float) – Excitation array and time step

12.2 C Library

12.2.1 Data Structures

struct `sdof_alpha`

Parameters for the generalized alpha method.

double `alpha_m`

WBZ alpha parameter

double `alpha_f`

HHT alpha parameter

double `gamma`

Newmarks γ parameter

double `beta`

Newmarks β parameter

struct `sdof_peaks`

Struct to return peak response quantities from threads by value.

double `max_displ`

Peak displacement

double `max_veloc`

Peak velocity

double `max_accel`

Peak acceleration

12.2.2 Integration

```
int sdof_integrate(struct sdof_alpha *conf, double M, double C, double K,
                   double scale, int n, double *p, double dt, double
                   *response)
```

This function integrates scalar differential equations of the form:

$$m\ddot{u} + c\dot{u} + ku = f(t)$$

for constant coefficients m, c, and k using the Generalized $\alpha-$method.

Parameters

- `conf` (`struct sdof_alpha*`) – Struct holding integration parameters
- `load` (`const double[n]`) – Pointer to excitation series
- `n` (`const int`) – Size of excitation series
- `dt` (`const double`) – Time step of excitation data
- `M` (`double`) – Mass
- `C` (`double`) – Damping
- `K` (`double`) – Stiffness
- `response` (`double[n][3]`) – Pointer to the beginning of an $n \times 3$ array of doubles. displacement at `i`: `response[i][0]` velocity at `i`: `response[i][1]` acceleration at `i`: `response[i][2]`

Note: The first row of the response array is expected to be initialized, ie

```
response[0][0] = u0;
response[0][1] = v0;
```

```
int sdof_integrate_plastic(struct sdof_alpha *conf, double M, double C,
                           double K, double scale, int n, double *p,
                           double dt, double *response)
```

Integrate an elastic-perfectly plastic system [80].

Parameters

- `fy` (`double`) – Yield force
- `a` (`double`) – Kinematic hardening ratio

```
int sdof_integrate_peaks(struct sdof_alpha *conf, double M, double C,
                         double K, double scale, int n, double *p, double
                         dt, struct sdof_peaks *response)
```

Integrate a linear system and store only peak values. Apart from the type of `response`, the remaining parameters are the same as for `sdof_integrate()`.

12.2.3 Response Spectra

```
int sdof_spectrum(struct sdof_alpha *conf, const double *load, const int
                  n, const double dt, const double t_min, const double
                  t_max, const int n_periods, const double damp, int
                  n_threads, struct sdof_peaks *response)
```

Threaded response spectrum over regularly spaced periods. This function spawns `n_threads` threads to perform a total of `n_periods` integrations over regularly spaced periods.

Parameters

- `conf` (`struct sdof_alpha*`) –
- `load` (`const double[n]`) – Pointer to excitation series
- `n` (`const int`) – Size of excitation series
- `dt` (`const double`) – Time step of excitation data
- `t_min` (`const double`) – Start period
- `t_max` (`const double`) – End period
- `n_periods` (`const int`) – Number of periods to integrate
- `damp` (`const double`) –
- `n_threads` (`int`) –
- `*response` (`struct sdof_peaks`) –

Chapter 13

`mdof`

The `mdof` library contains implementations of data-driven dynamical models and is used by the *BRACE*^{2} platform to execute Type II Predictions (Chapter 5). The library also stands alone as a tool for engineers to study dynamic response data. This chapter serves as a guide to the theory behind system identification methods (Section 13.1) and the `mdof` library's Python interface (Section 13.2).

13.1 Theory

System identification is the practice of identifying the dynamic equations governing a physical system based on measured outputs of the system [40, 38]. A structure's physical properties are often summarized by extracting representative quantities from these dynamic equations, such as the frequency, damping ratio, and shape of the natural modes of vibration. These quantities can also include a *realization* (see Section 13.1.1.5) of the system, which provides the response quantities (acceleration, velocity, and displacement) for any given dynamic imposed load.

The state of the structure is assumed to be characterized by its displacement $\mathbf{u}$ and velocity $\dot{\mathbf{u}}$. It is governed by Newton's laws of motion:

$$\mathbf{m}\ddot{\mathbf{u}} = \mathbf{f}^{\text{ext}} - \mathbf{f}^{\text{int}}(\mathbf{u}, \dot{\mathbf{u}}), \tag{13.1}$$

where $\mathbf{f}^{\text{int}}$ is a function that characterizes the resisting force produced by the structure in response to its configuration and $\mathbf{f}^{\text{ext}}$ is an imposed load. For a given load $\mathbf{f}^{\text{ext}}$ and acceleration $\ddot{\mathbf{u}}$, system identification is used to obtain the mass $\mathbf{m}$ and resisting force function $\mathbf{f}^{\text{int}}$.

Current methods for system identification in the *time-domain* rely on regression (computation with a pseudo-inverse) and principal component analysis (computation with a singular value decomposition) after defining subspaces of interest for the state vector [40]. Then, the focus is either put on impulse responses, or correlations among state, input, and output. The most well-established method for working with impulse response is the Ho-Kalman/ERA algorithm [29, 37]. Methods for working with correlations include those termed N4SID [90] for Numerical algorithm for Subspace State Space System IDentification and SRIM [35] for System Realization using Information Matrix. However, all of the aformentioned methods still lack enough establishment to have been adopted into the common vernacular of dynamics researchers. Therefore, a detailed report on the theory of time-domain methods, specific algorithmic steps and computational considerations for Ho-Kalman and SRIM, and example applications of time-domain methods are presented in Sections 13.1.1 to 13.1.7.

On the other hand, system identification in the *frequency-domain* does have a widely-accepted standard. All commonly-accepted methods rely on a Fast Fourier Transform (FFT), which is expected to provide most of the information that can be extracted from frequency-domain signal analysis. A summary of Fourier analysis is appendicized to this chapter (Section 13.6) for completeness.

13.1.1 State-Space Representation of Structural Dynamics

The *state-space representation* of a physical system's dynamics involves three time-dependent vectors: (1) the state vector, $\mathbf{x}$, (2) the input vector, $\mathbf{z}$, and (3) the output vector, $\mathbf{y}$. These three vectors are organized into a first order matrix differential equation. This section provides increasingly generalized examples for the construction of this differential equation that is the state space representation for various structural systems. All examples are Linear Time-Invariant (LTI) systems; that is, the operators on input and state within the state space representations are linear in state space and constant in time. Each example takes system input as the support accelerations $\ddot{\mathbf{u}}_g(t)$ and system output as the structural accelerations $\ddot{\mathbf{u}}_f(t)$. We begin with Single Degree Of Freedom (SDOF) systems in continuous-time subject to uniform excitation, then generalize incrementally to Multiple Degree Of Freedom (MDOF) systems in discrete-time subject to multiple-support excitation.

13.1.1.1 SDOF system subject to uniform excitation

For the SDOF system in Fig. 58, the equation of motion is expressed as follows,

$$m\ddot{u}_f(t) + c\dot{u}_f(t) + ku_f = -m\ddot{u}_g(t), \tag{13.2}$$

where superposed dots indicate time (t) derivatives and other variables are defined in Fig. 58.

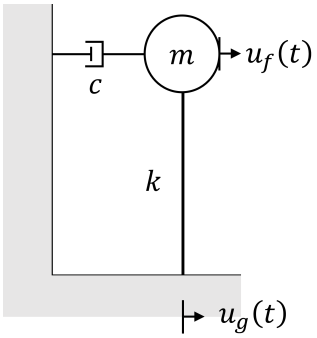


Fig. 58: SDOF system subject to uniform ground excitation.

The state-space representation of the system, with the state $\mathbf{x}=\begin{bmatrix} u_f(t) \\ \dot{u}_f(t) \end{bmatrix}$, the state derivative $\dot{\mathbf{x}}=\begin{bmatrix} \dot{u}_f(t) \\ \ddot{u}_f(t) \end{bmatrix}$, the input $\mathbf{z}=\ddot{u}_g(t)$, and the output $\mathbf{y}=\ddot{u}_f(t)$, is expressed as follows,

$$
\begin{aligned}
\dot{\mathbf{x}} &= \mathbf{A}_c\mathbf{x} + \mathbf{B}_c\mathbf{z}, \\
\begin{bmatrix} \dot{u}_f(t) \\ \ddot{u}_f(t) \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -k/m & -c/m \end{bmatrix} \begin{bmatrix} u_f(t) \\ \dot{u}_f(t) \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} \ddot{u}_g(t), \\
\mathbf{y} &= \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{z}, \\
\ddot{u}_f(t) &= \begin{bmatrix} -k/m & -c/m \end{bmatrix} \begin{bmatrix} u_f(t) \\ \dot{u}_f(t) \end{bmatrix} + \begin{bmatrix} -1 \end{bmatrix} \ddot{u}_g(t),
\end{aligned}
\tag{13.3}
$$

where

$\mathbf{A}_c$: Continuous state transition matrix,
$\mathbf{B}_c$: Continuous input influence matrix,
$\mathbf{C}$: Output influence matrix, and
$\mathbf{D}$: Direct transmission or feed-through matrix.

13.1.1.2 MDOF system subject to uniform excitation

For the MDOF system in Fig. 59, the system of governing equations of motion is as follows,

$$
\begin{aligned}
\mathbf{m}\ddot{\mathbf{u}}_f(t) + \mathbf{c}\dot{\mathbf{u}}_f(t) + \mathbf{k}\mathbf{u}_f(t) &= -\mathbf{m}\boldsymbol{\iota}\ddot{u}_g(t), \\
\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{bmatrix} \ddot{u}_{f1}(t) \\ \ddot{u}_{f2}(t) \end{bmatrix} + \begin{bmatrix} c_1 & 0 \\ 0 & c_2 \end{bmatrix} \begin{bmatrix} \dot{u}_{f1}(t) \\ \dot{u}_{f2}(t) \end{bmatrix} + \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix} \begin{bmatrix} u_{f1}(t) \\ u_{f2}(t) \end{bmatrix} &= - \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \ddot{u}_g(t),
\end{aligned}
\tag{13.4}
$$

where $\boldsymbol{\iota}$ is the *influence vector* [19] and all variables are defined in Fig. 59, which shows an example of a stiffness-coupled system consisting of two DOF.

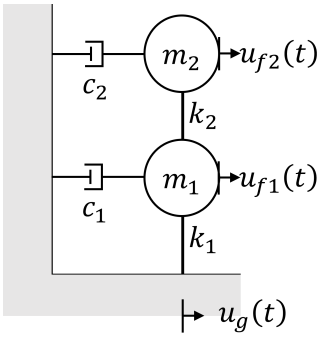


Fig. 59: MDOF system subject to uniform ground excitation.

The state-space representation of the system, with the state $\mathbf{x}= \begin{bmatrix} \mathbf{u}_f(t) \\ \dot{\mathbf{u}}_f(t) \end{bmatrix}$, the state derivative

$\dot{\mathbf{x}}= \begin{bmatrix} \dot{\mathbf{u}}_f(t) \\ \ddot{\mathbf{u}}_f(t) \end{bmatrix}$, the input $\mathbf{z}= \ddot{u}_g(t)$, and the output $\mathbf{y}= \ddot{\mathbf{u}}_f(t)$, is expressed as follows,

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}_c\mathbf{x} + \mathbf{B}_c\mathbf{z}, \\ \begin{bmatrix} \dot{\mathbf{u}}_f(t) \\ \ddot{\mathbf{u}}_f(t) \end{bmatrix} &= \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{m}^{-1}\mathbf{k} & -\mathbf{m}^{-1}\mathbf{c} \end{bmatrix} \begin{bmatrix} \mathbf{u}_f(t) \\ \dot{\mathbf{u}}_f(t) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ -\boldsymbol{\iota} \end{bmatrix} \ddot{u}_g(t), \\ \mathbf{y} &= \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{z}, \\ \ddot{\mathbf{u}}_f(t) &= \begin{bmatrix} -\mathbf{m}^{-1}\mathbf{k} & -\mathbf{m}^{-1}\mathbf{c} \end{bmatrix} \begin{bmatrix} \mathbf{u}_f(t) \\ \dot{\mathbf{u}}_f(t) \end{bmatrix} + \begin{bmatrix} -\boldsymbol{\iota} \end{bmatrix} \ddot{u}_g(t), \end{aligned} \tag{13.5}$$

where $\mathbf{I}$ and $\mathbf{0}$ are the *identity matrix* and *null matrix*, respectively, and all other matrices and vectors are defined similar to their counterparts in the previous section.

13.1.1.3 MDOF system subject to multiple support excitation

When a MDOF system is subject to multiple support excitation, such as in Fig. 60, the displacement vector is extended to include the support DOF. The system of governing equations for dynamic equilibrium (Eq. (13.8)) is derived based on [19] in the following paragraphs.

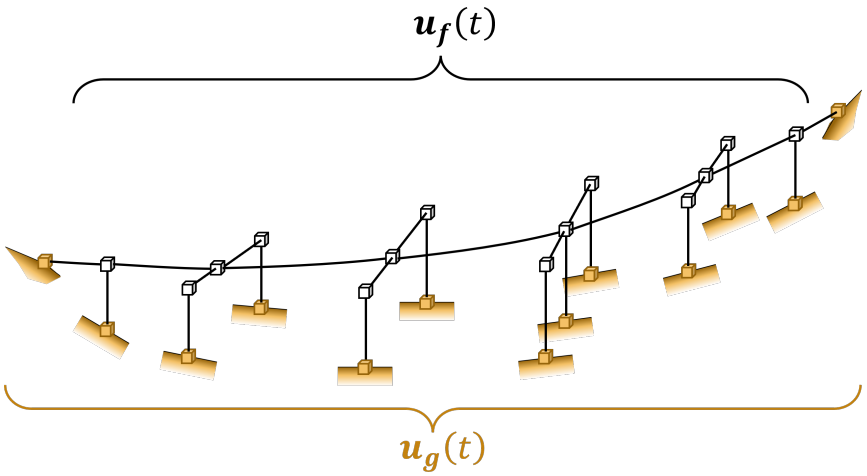


Fig. 60: MDOF system subject to multiple support excitation.

Begin by forming a partitioned system of equations for dynamic equilibrium for all the DOF:

$$\begin{bmatrix} \mathbf{m} & \mathbf{m}_g \\ \mathbf{m}_g^T & \mathbf{m}_{gg} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{u}}_f^t \\ \ddot{\mathbf{u}}_g \end{bmatrix} + \begin{bmatrix} \mathbf{c} & \mathbf{c}_g \\ \mathbf{c}_g^T & \mathbf{c}_{gg} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{u}}_f^t \\ \dot{\mathbf{u}}_g \end{bmatrix} + \begin{bmatrix} \mathbf{k} & \mathbf{k}_g \\ \mathbf{k}_g^T & \mathbf{k}_{gg} \end{bmatrix} \begin{bmatrix} \mathbf{u}_f^t \\ \mathbf{u}_g \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{f}_g \end{bmatrix}, \tag{13.6}$$

where the subscripts g and f indicate support and structural DOF, respectively, the superscript T indicates the transpose, and the superscript t indicates the total of quasi-static ($\mathbf{u}_f^s$, due to static application of support displacements) and dynamic ($\mathbf{u}_f$, evaluated by dynamic analysis) structural displacements.

Taking the first half of the partitioned equilibrium, separating the structural motions ($\mathbf{u}_f^t = \mathbf{u}_f^s + \mathbf{u}_f$, $\dot{\mathbf{u}}_f^t = \dot{\mathbf{u}}_f^s + \dot{\mathbf{u}}_f$, and $\ddot{\mathbf{u}}_f^t = \ddot{\mathbf{u}}_f^s + \ddot{\mathbf{u}}_f$), and moving all $\mathbf{u}_g$ and $\mathbf{u}_f^s$ terms to the right side,

$$\mathbf{m}\ddot{\mathbf{u}}_f + \mathbf{c}\dot{\mathbf{u}}_f + \mathbf{k}\mathbf{u}_f = -(\mathbf{m}\ddot{\mathbf{u}}_f^s + \mathbf{m}_g\ddot{\mathbf{u}}_g) - (\mathbf{c}\dot{\mathbf{u}}_f^s + \mathbf{c}_g\dot{\mathbf{u}}_g) - (\mathbf{k}\mathbf{u}_f^s + \mathbf{k}_g\mathbf{u}_g). \tag{13.7}$$

The term $(\mathbf{k}\mathbf{u}_f^s + \mathbf{k}_g\mathbf{u}_g) = \mathbf{0}$ due to static equilibrium, allowing it to be dropped and giving $\mathbf{u}_f^s = -\mathbf{k}^{-1}\mathbf{k}_g\mathbf{u}_g = \iota\mathbf{u}_g$, i.e., $\ddot{\mathbf{u}}_f^s = \iota\ddot{\mathbf{u}}_g$, where $\iota = -\mathbf{k}^{-1}\mathbf{k}_g$ is the *influence matrix* [19], the term $(\mathbf{c}\dot{\mathbf{u}}_f^s + \mathbf{c}_g\dot{\mathbf{u}}_g)$ can be dropped because it is usually small relative to the inertia term, and finally the term $\mathbf{m}_g\ddot{\mathbf{u}}_g$ can also be dropped because the mass is usually neglected at the supports. Accordingly, the equilibrium equations are simplified to:

$$\mathbf{m}\ddot{\mathbf{u}}_f + \mathbf{c}\dot{\mathbf{u}}_f + \mathbf{k}\mathbf{u}_f = -\mathbf{m}\iota\ddot{\mathbf{u}}_g. \tag{13.8}$$

The state-space representation here is similar to the uniform excitation case, but with the support displacements $\ddot{\mathbf{u}}_g$ defined by a vector rather than a scalar. The continuous LTI state-space representation of a MDOF structural system subject to multiple-support excitation is the same as Eq. (13.5) with the exception that the scalar $\ddot{u}_g(t)$ is replaced by the vector $\ddot{\mathbf{u}}_g$.

13.1.1.4 Discrete representation of continuous LTI system dynamics

Because signals are measured at discrete time-points, their analysis requires transformation between the continuous (s-plane) domain, where models based on differentials are defined, and the discrete (z-plane) domain, where the actual data resides. In order to move from the continuous to the discrete domain, the coefficients $\mathbf{A}_c$ and $\mathbf{B}_c$ are transformed by solving the first-order differential equation with the input signal's value held constant within each time step, Δt, ("zero-order hold method", Fig. 61). The coefficients $\mathbf{C}$ and $\mathbf{D}$ are unchanged and as defined after Eq. (13.3). The resulting discrete equations are as follows,

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{z}_k, \\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{z}_k, \end{aligned} \tag{13.9}$$

$$\mathbf{x}_k = \mathbf{x}(k\Delta t), \qquad \mathbf{z}_k = \mathbf{z}(k\Delta t), \qquad \mathbf{y}_k = \mathbf{y}(k\Delta t), \tag{13.10}$$

$$\mathbf{A} = e^{\mathbf{A}_c\Delta t}, \qquad \mathbf{B} = \int_0^{\Delta t} e^{\mathbf{A}_c\tau}\mathbf{B}_c d\tau, \tag{13.11}$$

where

A: Discrete state transition matrix, and
B: Discrete input influence matrix.

The zero-order hold method is described as follows. Define the discrete state and input signals, $\mathbf{x}_k = \mathbf{x}(k\Delta t)$, $\mathbf{x}_{k+1} = \mathbf{x}\left((k+1)\Delta t\right)$, $\mathbf{z}_k = \mathbf{z}(k\Delta t)$. Then, given the first part of the continuous form of the state-space representation, i.e., the first-order linear differential of Eq. (13.5), one solves it with the initial condition $\mathbf{x}(t_0)$ to obtain:

$$\mathbf{x}(t) = e^{\mathbf{A}_c(t-t_0)}\mathbf{x}(t_0) + \int_{t_0}^{t} e^{\mathbf{A}_c(t-\tau)}\mathbf{B}_c\mathbf{z}(\tau)d\tau. \tag{13.12}$$

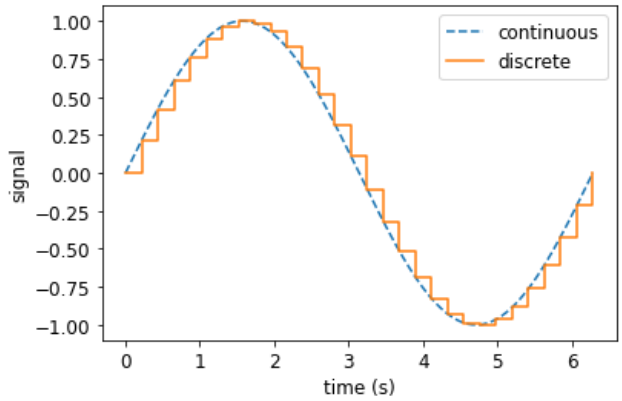


Fig. 61: Signal discretization using the zero-order hold method.

Substituting $t = (k+1)\Delta t$ and $t_0 = k\Delta t$, Eq. (13.12) becomes:

$$\mathbf{x}\left((k+1)\Delta t\right) = e^{\mathbf{A}_c \Delta t}\mathbf{x}(k\Delta t) + \int_{k\Delta t}^{(k+1)\Delta t} e^{\mathbf{A}_c((k+1)\Delta t - \tau)}\mathbf{B}_c\mathbf{z}(\tau)d\tau. \tag{13.13}$$

Assume the input $\mathbf{z}(t)$ is constant within the sample time step Δt:

$$\begin{aligned}
&\mathbf{z}(\tau) = \mathbf{z}(k\Delta t) \text{ for } k\Delta t \le \tau < (k+1)\Delta t,\\
&\mathbf{x}\left((k+1)\Delta t\right) = e^{\mathbf{A}_c \Delta t}\mathbf{x}(k\Delta t) + \int_{k\Delta t}^{(k+1)\Delta t} e^{\mathbf{A}_c((k+1)\Delta t - \tau)} d\tau \ \mathbf{B}_c\mathbf{z}(k\Delta t),\\
&\text{Let} \quad \tau' = (k+1)\Delta t - \tau, \quad d\tau' = -d\tau,\\
&\mathbf{x}\left((k+1)\Delta t\right) = e^{\mathbf{A}_c \Delta t}\mathbf{x}(k\Delta t) + \int_{0}^{\Delta t} e^{\mathbf{A}_c \tau'} d\tau' \ \mathbf{B}_c\mathbf{z}(k\Delta t).
\end{aligned} \tag{13.14}$$

This finally yields the discrete form of the LTI state-space representation:

$$\mathbf{x}_{k+1} = e^{\mathbf{A}_c \Delta t}\mathbf{x}_k + \int_{0}^{\Delta t} e^{\mathbf{A}_c \tau'} d\tau' \ \mathbf{B}_c\mathbf{z}_k. \tag{13.15}$$

13.1.1.5 System realization

A system realization is a set of dynamic equation coefficients that produce the same response as the system for a given input. Most often, these are the state space coefficients, **A**, **B**, **C**, and **D**, refer to Eq. (13.9). A physical system has a unique system realization up to a coordinate transformation, expressed through the *invertible transformation matrix*, **T**. In other words, a given system can have the realization **A**, **B**, **C**, and **D** as well as the realization $\mathbf{T}^{-1}\mathbf{A}\mathbf{T}$, $\mathbf{T}^{-1}\mathbf{B}$, $\mathbf{C}\mathbf{T}$, and **D**.

13.1.2 Modal Properties from State Space Realization

Structural system dynamics are defined in continuous time. With time-domain system identification methods such as OKID-ERA (Sections 13.1.3 and 13.1.4) and SRIM (Section 13.1.6), a structural system's discrete-time state space realization is obtained from data measured by instrumentation on the structure. The process discussed in the following sub-sections recovers the structure's modal properties from the discrete state space realization coefficients, **A** and **C**.

13.1.2.1 Eigendecompositions of $\mathbf{A}_c$ and $\mathbf{A}$

The relationship between the eigendecompositions of $\mathbf{A}_c$, the continuous-time state transition matrix, and $\mathbf{A}$, the discrete-time state transition matrix, can be established as follows,

$$\begin{aligned} \mathbf{A}_c &= \Phi\Lambda\Phi^{-1}, \\ \mathbf{A} &= \Psi\Gamma\Psi^{-1}, \end{aligned} \tag{13.16}$$

$$\mathbf{A} = e^{\mathbf{A}_c\Delta t} = e^{\Phi\Lambda\Phi^{-1}\Delta t} = \Phi e^{\Lambda\Delta t}\Phi^{-1}, \tag{13.17}$$

$$\Psi = \Phi, \quad \Gamma = e^{\Lambda\Delta t}, \tag{13.18}$$

where

$$\Psi = \begin{bmatrix} \psi_1 & \psi_2 & \cdots & \psi_r \end{bmatrix}, \quad \Phi = \begin{bmatrix} \phi_1 & \phi_2 & \cdots & \phi_r \end{bmatrix}, \tag{13.19}$$

$$\Gamma = \begin{bmatrix} \gamma_1 & 0 & \cdots & 0 \\ 0 & \gamma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \gamma_r \end{bmatrix}, \quad \Lambda = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_r \end{bmatrix}, \tag{13.20}$$

- γ_j and ψ_j are the eigenvalues and eigenvectors of $\mathbf{A}$,
- λ_j and ϕ_j are the eigenvalues and eigenvectors of $\mathbf{A}_c$,
- $j \in [1, 2, \cdots, r]$, and r is the *model order* (refer to Section 13.1.2.2).

It is noted that the result obtained in Eq. (13.17) can be proved using the Taylor series expansion of the matrix exponential as follows:

$$\begin{aligned} e^{\mathbf{A}_c\Delta t} &= \mathbf{I} + \Phi\Lambda\Phi^{-1}\Delta t + \frac{1}{2!}\Phi\Lambda\Phi^{-1}\Phi\Lambda\Phi^{-1}(\Delta t)^2 + \cdots \\ &= \Phi\,\Phi^{-1} + \Phi\Lambda\Phi^{-1}\Delta t + \frac{1}{2!}\Phi\Lambda\Phi^{-1}\Phi\Lambda\Phi^{-1}(\Delta t)^2 + \cdots \\ &= \Phi\left[\mathbf{I} + \Lambda\Delta t + \frac{1}{2!}\Lambda^2(\Delta t)^2 + \cdots\right]\Phi^{-1} \\ &= \Phi e^{\Lambda\Delta t}\Phi^{-1}. \end{aligned}$$

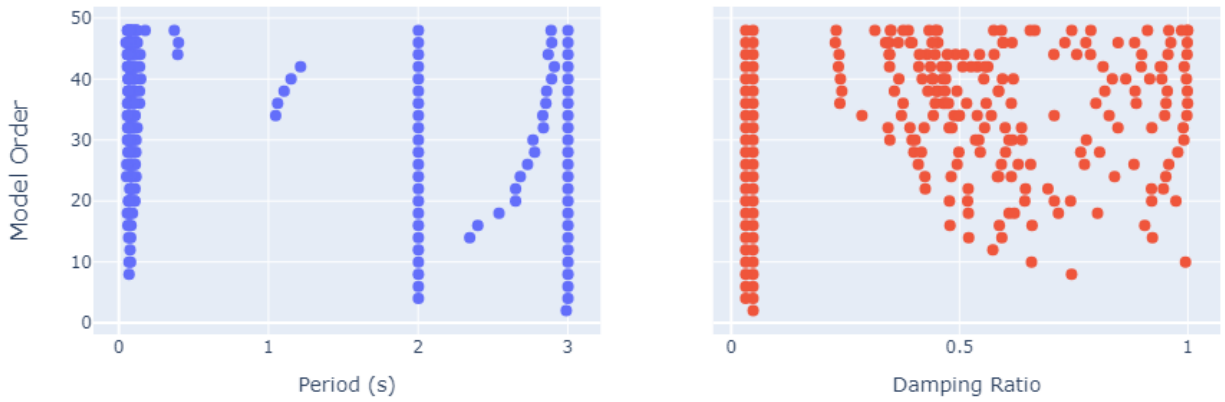
13.1.2.2 Model order and modes

When performing system identification to obtain a state space realization, the *model order* must be configured as a parameter that is chosen before execution. Model order is the rank of $\mathbf{A}$. For the state space representation described above, $\mathbf{A} \in \mathbb{R}^{n\times n}$ is square and assumed to be full-rank, so the model order = n.

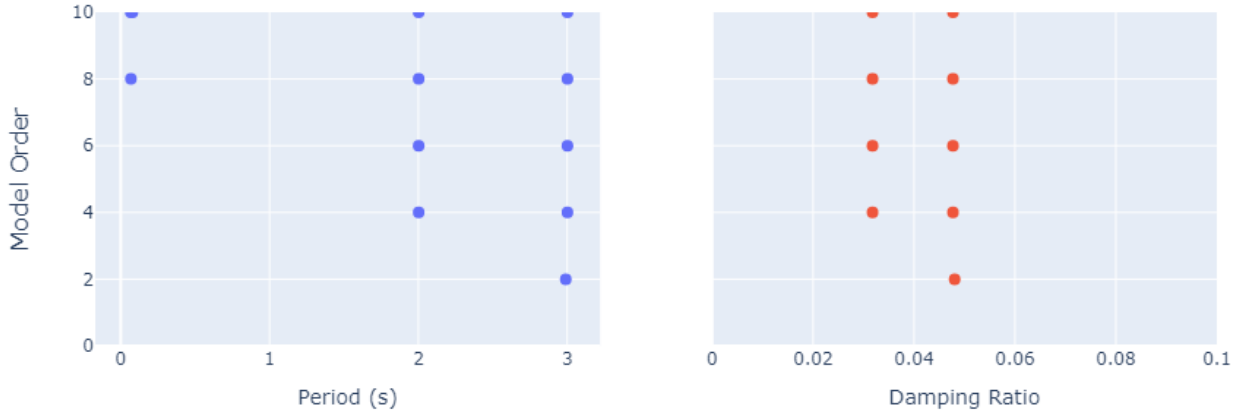
From a theoretical point of view, the model order is equal to twice the number of components in $\mathbf{u}$, i.e., the structural displacement degrees of freedom, which is the same as the number of fundamental modes. However, a real physical structure has infinite degrees of freedom. Therefore, an approximation is made by assuming that some modes are much harder to excite than others, or have much less participation in the dynamic response. Commonly, for standard bridge structures, it is assumed that there are few (usually one to three) modes governed by the movement of the deck in each

direction: lateral, longitudinal, and vertical. From a numerical experimentation point of view using field data from instrumented bridges, it has been found from [5] that when system identification is performed with model orders larger than 50 on measured structural vibrations, the results include spurious, non-meaningful modes. A model order of 4 and 20 is the minimum to respectively capture the first two and ten natural modes of vibration. Based on prior studies, e.g., [5], it is generally expected that a two-span standard bridge will have fewer than 10 significantly contributing modes.

Optimal model order for the system can be calibrated using a *stabilization diagram*. For several model orders, a state-space realization algorithm is used to identify modal properties. The minimum model order that corresponds to consistent modal property values is taken as the required model order for system identification of the given structural system. An example is shown in Fig. 62. In the example, state-space realizations are computed for a 2 DOF system for model orders between 2 and 50. The fundamental periods, 2 and 3 seconds, are captured after increasing the model order to 4. The damping ratios, 0.048 and 0.032, respectively, are also captured after increasing the model order to 4. After increasing the model order to 8 and above, spurious modes appear for both periods and damping ratios. Thus, the optimal model order for this system is 4 or 6. Without this detailed calibration, one must estimate the required model order as twice the approximate number of contributing modes.



(a) Un-zoomed plot



(b) Zoomed plot

Fig. 62: Stabilization diagram of a 2 DOF system.

13.1.2.3 Natural frequencies and modal damping ratios

From the eigendecompositions of $\mathbf{A}_c$ and $\mathbf{A}$ and using definitions in Eq. (13.20), we have

$$\Gamma = e^{\Lambda \Delta t} \implies \lambda_j = (\ln \gamma_j)/\Delta t. \tag{13.21}$$

Considering modal damping, the eigenvalue λ_j of the j-th mode contains the damped natural frequency ω_j and the modal damping factor ζ_j according to the following derivation:

$$\lambda_j = -\zeta_j \omega_j \pm i\left(\omega_j \sqrt{1-\zeta_j^2}\right), \quad i = \sqrt{-1},$$

$$\begin{aligned}
\lambda_j \bar{\lambda}_j &= \zeta_j^2 \omega_j^2 + \omega_j^2 \left(1-\zeta_j^2\right) \\
&= \omega_j^2 \left(\zeta_j^2 + 1 - \zeta_j^2\right) \\
&= \omega_j^2,
\end{aligned}$$

where the overline symbol $\bar{\bullet}$ indicates the complex conjugate. Therefore, we obtain,

$$\begin{aligned}
\omega_j &= \sqrt{\lambda_j \bar{\lambda}_j} = |\lambda_j|, \\
\zeta_j &= -\frac{\mathrm{Re}(\lambda_j)}{\omega_j},
\end{aligned} \tag{13.22}$$

where $\mathrm{Re}(\bullet)$ indicates the real part of the complex number.

13.1.2.4 Mode shapes

The eigenvectors Φ of the continuous state transition matrix $\mathbf{A}_c$ transform the modal coordinates $\mathbf{q}$ into the state coordinates $\mathbf{x}$. We define the mode shapes, Φ_{modal}, as the transformation from modal coordinates $\mathbf{q}$ to the output coordinates $\mathbf{y}$. Therefore, we obtain:

$$\begin{aligned}
\mathbf{x}(t) &= \Phi \mathbf{q}(t), \\
\mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{z}(t), \\
\Phi_{\text{modal}} &= \mathbf{C}\Phi.
\end{aligned} \tag{13.23}$$

13.1.2.5 Modal indicators

The modal indicators, extended modal amplitude coherence (EMAC) and modal phase collinearity (MPC), are measures of trustworthiness of modal parameter prediction [68]. EMAC and MPC are defined for the i-th mode in Eqs. (13.24) and (13.25), respectively.

$$\mathrm{EMAC}_i = R_{ij} W_{ij}, \tag{13.24}$$

where

$$R_{ij} = \min\left\{\frac{\Phi_{ij,T}}{\tilde{\Phi}_{ij,T}}, \frac{\tilde{\Phi}_{ij,T}}{\Phi_{ij,T}}\right\},$$

$$W_{ij} = \max\left\{1 - \frac{\arg\left(\Phi_{ij,T} \,/\, \tilde{\Phi}_{ij,T}\right)}{\pi/4}, 0\right\},$$

$\Phi_{ij,T}$ = measured j^{th} component of the i^{th} mode at time T, and

$\tilde{\Phi}_{ij,T}$ = predicted j^{th} component of the i^{th} mode at time T.

Note that $\theta = \arg(Z)$, where $Z = a + ib$ is a complex number, is computed as $\theta = \tan^{-1}(b/a)$.

$$\text{MPC}_i = \left(\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2}\right)^2, \tag{13.25}$$

where

$$\lambda_{1,2} = \frac{S_{xx} + S_{yy}}{2} \pm S_{xy}\sqrt{\eta^2 + 1},$$

$$\eta = \frac{S_{yy} - S_{xx}}{2S_{xy}},$$

$$S_{xx} = \text{Re}(\Phi_i)^T \text{Re}(\Phi_i),$$

$$S_{yy} = \text{Im}(\Phi_i)^T \text{Im}(\Phi_i),$$

$$S_{xy} = \text{Re}(\Phi_i)^T \text{Im}(\Phi_i),$$

$\text{Re}(\Phi_i)$ = real part of the complex vector Φ_i, representing the i-th mode shape, and

$\text{Im}(\Phi_i)$ = imaginary part of the complex vector Φ_i.

13.1.3 Ho-Kalman/Eigensystem Realization Algorithm (ERA)

As discussed in Section 13.1.1, a structural system's dynamics can be represented by the four coefficients (**A**, **B**, **C**, and **D**) of its discrete LTI state-space representation. The Ho-Kalman algorithm [29] or Eigensystem Realization Algorithm (ERA) [37], produces a *reduced order model* for estimating these four coefficients ($\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$, $\tilde{\mathbf{C}}$, and $\tilde{\mathbf{D}}$) based on an impulse input and its corresponding response output, refer to the illustration of Fig. 63.

With the discrete LTI model, refer to Eq. (13.9), a unit impulse input with zero initial conditions, refer to Eq. (13.26), produces an output of constants ($\mathbf{D}$, $\mathbf{CB}$, $\mathbf{CAB}$, $\mathbf{CA}^2\mathbf{B}$, $\cdots$, $\mathbf{CA}^{k-1}\mathbf{B}$). These constants are called *Markov parameters*; they must be unique for a given system – there is only one possible output for a unit impulse input.

$$\begin{aligned}
\mathbf{z}_0, \mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_k &= \mathbf{I}, \mathbf{0}, \mathbf{0}, \cdots, \mathbf{0}, \\
\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_k &= \mathbf{0}, \mathbf{B}, \mathbf{AB}, \mathbf{A}^2\mathbf{B}, \cdots, \mathbf{A}^{k-1}\mathbf{B}, \\
\mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_k &= \mathbf{D}, \mathbf{CB}, \mathbf{CAB}, \mathbf{CA}^2\mathbf{B}, \cdots, \mathbf{CA}^{k-1}\mathbf{B}.
\end{aligned} \tag{13.26}$$

Knowing that the impulse response output data directly give the Markov parameters, the data

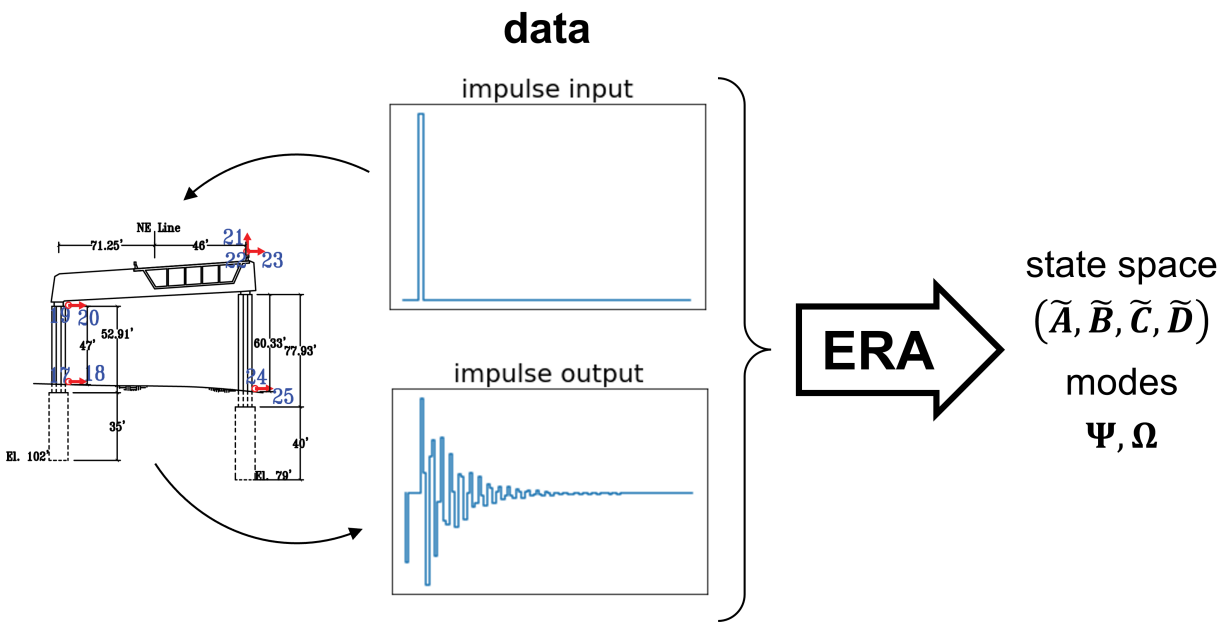


Fig. 63: Ho-Kalman/Eigensystem Realization Algorithm (ERA).

can then be stacked into the generalized Hankel matrix (Section 13.4.1) $\mathbf{H}$ in Eq. (13.27)[20]:

$$\mathbf{H} = \begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_{m_c} \\ \mathbf{y}_2 & \mathbf{y}_3 & \cdots & \mathbf{y}_{m_c+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_{m_o} & \mathbf{y}_{m_o+1} & \cdots & \mathbf{y}_{m_o+m_c-1} \end{bmatrix} = \begin{bmatrix} \mathbf{CB} & \mathbf{CAB} & \cdots & \mathbf{CA}^{m_c-1}\mathbf{B} \\ \mathbf{CAB} & \mathbf{CA}^2\mathbf{B} & \cdots & \mathbf{CA}^{m_c}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{CA}^{m_o-1}\mathbf{B} & \mathbf{CA}^{m_o}\mathbf{B} & \cdots & \mathbf{CA}^{m_c+m_o-2}\mathbf{B} \end{bmatrix} = \mathcal{O}C, \quad (13.27)$$

where the Hankel matrix dimensions, namely, m_c and m_o, are discussed below. Moreover, $\mathcal{O}$ and C are the observability and controllability matrices of the system:

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{m_o-1} \end{bmatrix}, \qquad C = \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} & \cdots & \mathbf{A}^{m_c-1}\mathbf{B} \end{bmatrix}. \quad (13.28)$$

The shifted Hankel matrix, $\mathbf{H}'$ (one time step ahead of $\mathbf{H}$), is expressed as follows,

$$\mathbf{H}' = \begin{bmatrix} \mathbf{y}_2 & \mathbf{y}_3 & \cdots & \mathbf{y}_{m_c+1} \\ \mathbf{y}_3 & \mathbf{y}_4 & \cdots & \mathbf{y}_{m_c+2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_{m_o+1} & \mathbf{y}_{m_o+2} & \cdots & \mathbf{y}_{m_o+m_c} \end{bmatrix} = \begin{bmatrix} \mathbf{CAB} & \mathbf{CA}^2\mathbf{B} & \cdots & \mathbf{CA}^{m_c}\mathbf{B} \\ \mathbf{CA}^2\mathbf{B} & \mathbf{CA}^3\mathbf{B} & \cdots & \mathbf{CA}^{m_c+1}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{CA}^{m_o}\mathbf{B} & \mathbf{CA}^{m_o+1}\mathbf{B} & \cdots & \mathbf{CA}^{m_c+m_o-1}\mathbf{B} \end{bmatrix} = \mathcal{O}\mathbf{A}C. \quad (13.29)$$

[20]The matrix is a *generalized* Hankel matrix because it is not necessarily square and it has constant skew-diagonals in a block-by-block sense rather than in an element-by-element sense.

By taking the dominant terms of the Singular Value Decomposition (SVD) of $\mathbf{H}$ and transforming the relationship between $\mathbf{H} = \mathcal{O}\mathcal{C}$ and $\mathbf{H}' = \mathcal{O}\mathbf{A}\mathcal{C}$, a reduced-order model, Eq. (13.32), is constructed.

$$\mathbf{H} = \mathbf{U}\Sigma\mathbf{V}^H = \begin{bmatrix}\tilde{\mathbf{U}} & \mathbf{U}_t\end{bmatrix}\begin{bmatrix}\tilde{\Sigma} & \mathbf{0} \\ \mathbf{0} & \Sigma_t\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{V}}^H \\ \mathbf{V}_t^H\end{bmatrix} \approx \tilde{\mathbf{U}}\tilde{\Sigma}\tilde{\mathbf{V}}^H, \tag{13.30}$$

where the superscript H denotes conjugate transpose and the subscript t indicates elements to be truncated such that only the first r dominant singular values in $\tilde{\Sigma}$ are retained. The modal properties of the structural system can be estimated from this reduced-order model according to Section 13.1.2.

$$\begin{aligned}
\tilde{\mathbf{A}} &= \tilde{\Sigma}^{-1/2}\tilde{\mathbf{U}}^H\mathbf{H}'\tilde{\mathbf{V}}\tilde{\Sigma}^{-1/2}, \\
\tilde{\mathbf{B}} &= \tilde{\Sigma}^{1/2}\tilde{\mathbf{V}}^H\begin{bmatrix}\mathbf{I}_q \\ \mathbf{0}\end{bmatrix}, \\
\tilde{\mathbf{C}} &= \begin{bmatrix}\mathbf{I}_p & \mathbf{0}\end{bmatrix}\tilde{\mathbf{U}}\tilde{\Sigma}^{1/2}, \\
\tilde{\mathbf{D}} &= \mathbf{y}_0,
\end{aligned} \tag{13.31}$$

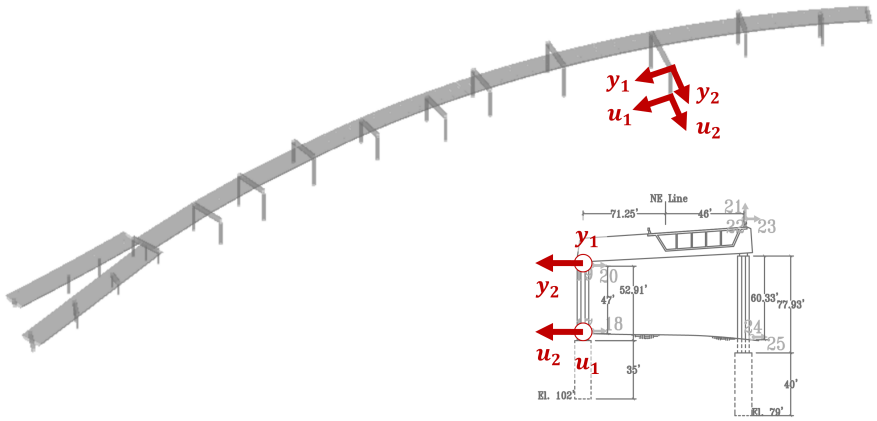
where p and q refer to the numbers of outputs and inputs, respectively. Finally, the discrete LTI model in Eq. (13.9) reduces to the following,

$$\begin{aligned}
\tilde{\mathbf{x}}_{k+1} &= \tilde{\mathbf{A}}\tilde{\mathbf{x}}_k + \tilde{\mathbf{B}}\mathbf{z}_k, \\
\mathbf{y}_k &= \tilde{\mathbf{C}}\tilde{\mathbf{x}}_k + \tilde{\mathbf{D}}\mathbf{z}_k.
\end{aligned} \tag{13.32}$$

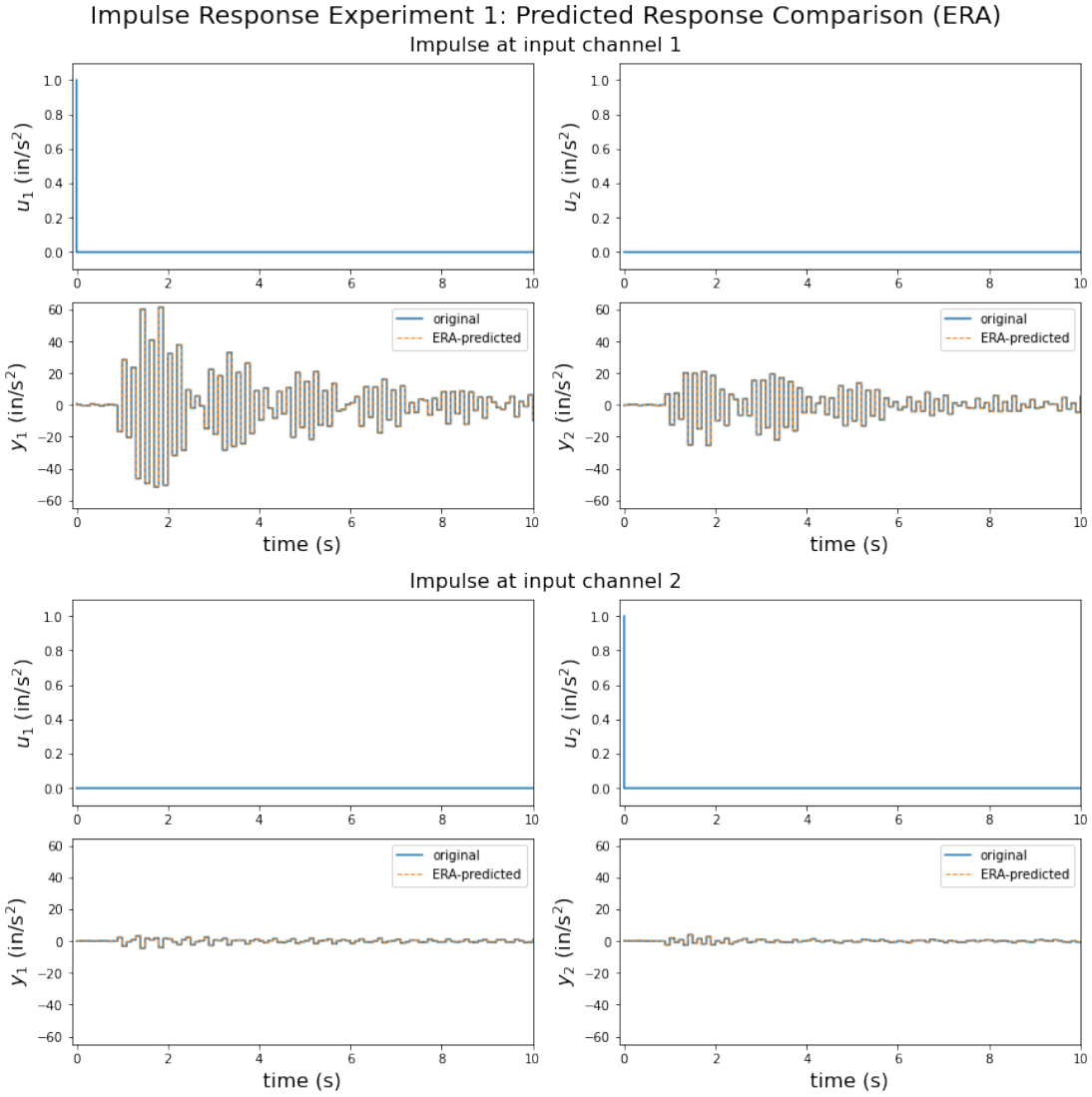
The implementation of the Ho-Kalman algorithm benefits from the following common algorithmic considerations:

- The SVD can be computationally expensive to compute, but can be made more efficient with randomization techniques and knowledge of the required model order.
- $\tilde{\Sigma}$ does not need to be inverted and taken to the half power. The reciprocals and square roots of the individual singular values can be used to save computational cost.
- Indexing of the first q columns and first p rows in the computations of $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{C}}$ can be used in lieu of the multiplication by $\mathbf{I}_q$ and $\mathbf{I}_p$.
- When choosing the Hankel matrix dimensions (m_o, m_c) and the reduced model order (r), the following constraints apply:
  - $m_o + m_c < K$, where K indicates the total number of time steps,
  - $r < \min\{m_o, m_c\}$,
  - $p < m_o$, and
  - $q < m_c$.

Impulse responses were generated by the numerical model for the Hayward Bridge (Chapter 8) for two input-output configurations (Figs. 64a and 65a). The Ho-Kalman algorithm was then used to predict the output data using a reduced model order of $r = 50$ (Figs. 64 and 65). The natural periods of the reduced models were then compared to the eigensolution of the numerical model (Fig. 66). Configuration IO_2 seems to capture Mode 5, while IO_1 does not.

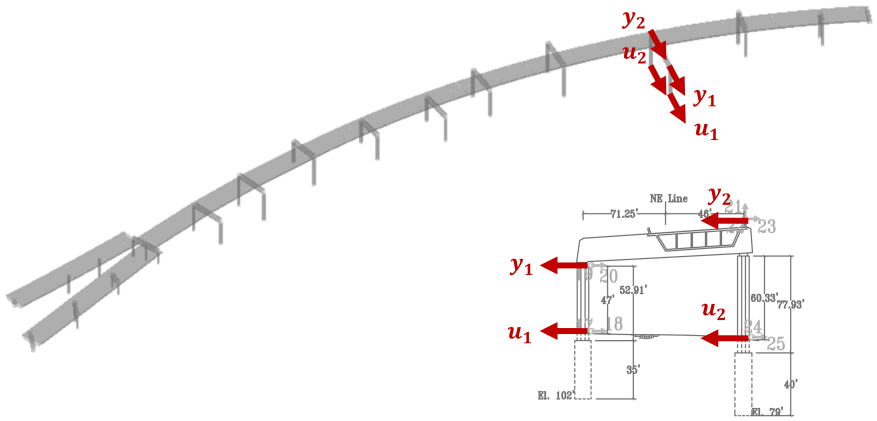


(a) Input-output configuration 1 (IO_1)

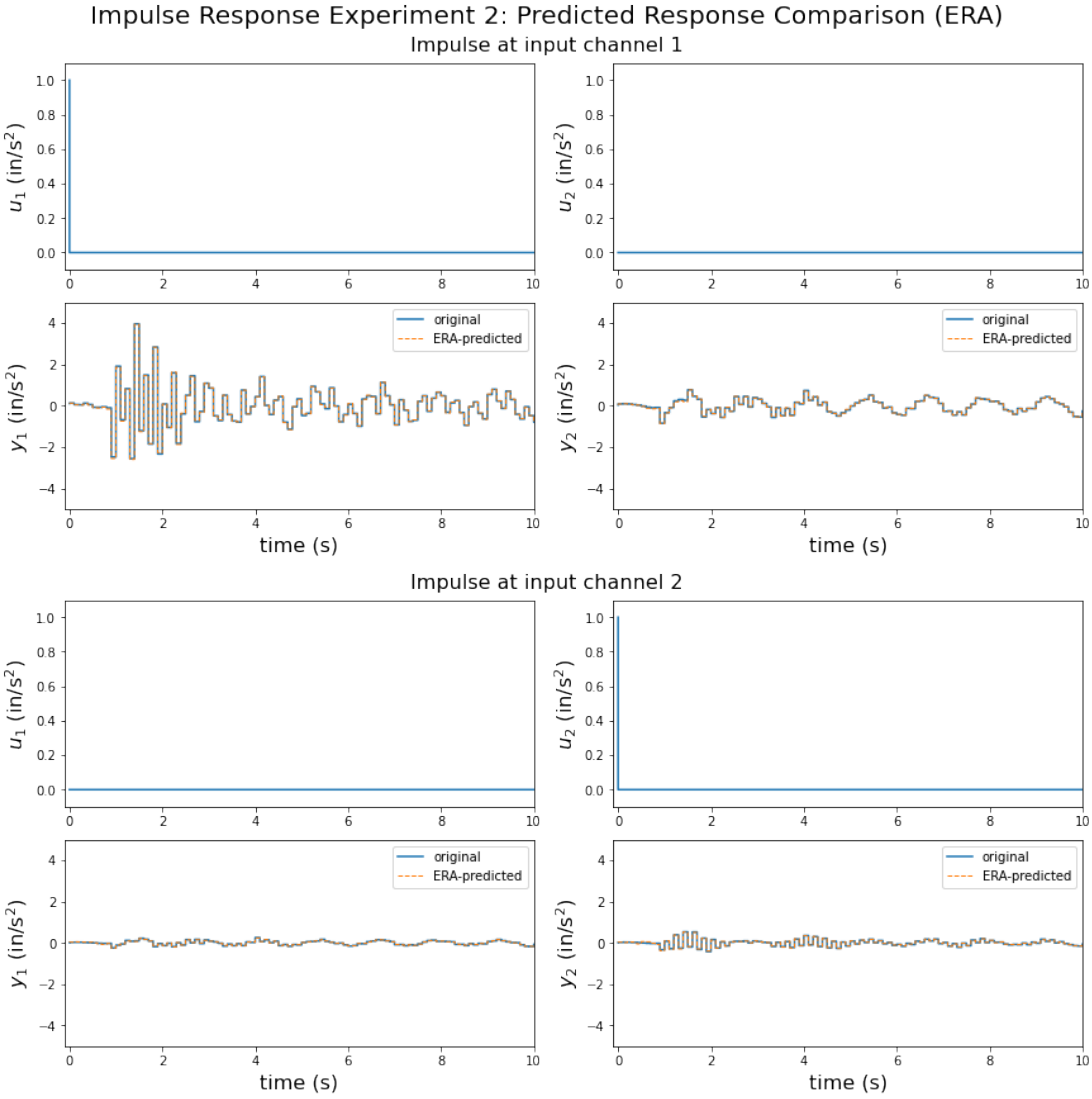


(b) Predicted vs. original impulse response

Fig. 64: Ho-Kalman algorithm prediction of impulse response for numerically modeled data with input-output configuration 1 (IO_1).



(a) Input-output configuration 2 (IO_2)



(b) Predicted vs. original impulse response

Fig. 65: Ho-Kalman algorithm prediction of impulse response for numerically modeled data with input-output configuration 2 (IO_2).

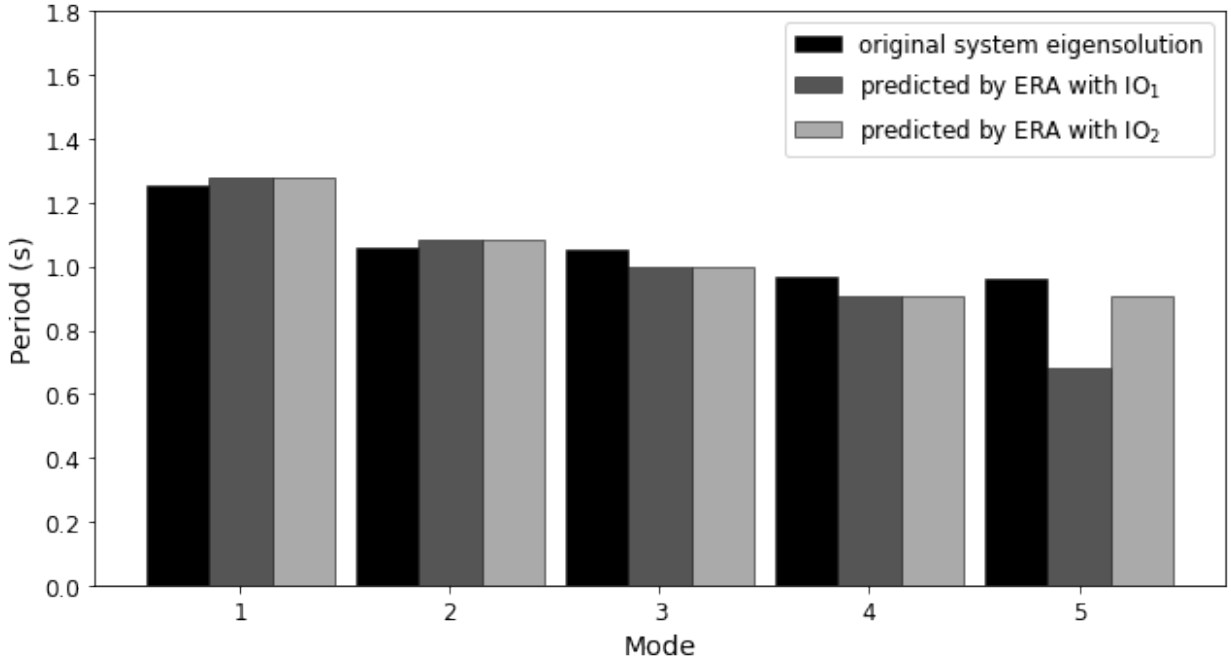


Fig. 66: Ho-Kalman algorithm natural period predictions from numerically modeled impulse response data.

13.1.4 Observer Kalman Filter Identification – ERA (OKID-ERA)

Structural dynamics from real data of instrumented structures are noisy, hard to measure, and lightly damped. Therefore, the Ho-Kalman algorithm/ERA is intended only to characterize impulse responses rather than responses to time histories. However, available data from ambient or small ground motion excitations during structure service can be *de-noised* and used to estimate the impulse response data. Then, the Ho-Kalman algorithm can be used to obtain a reduced order model even if the available data are not a "clean impulse response". This process is called Observer Kalman Filter Identification (OKID) [39] or OKID-ERA when OKID is combined with Eigensystem Realization Algorithm (ERA), refer to Fig. 67.

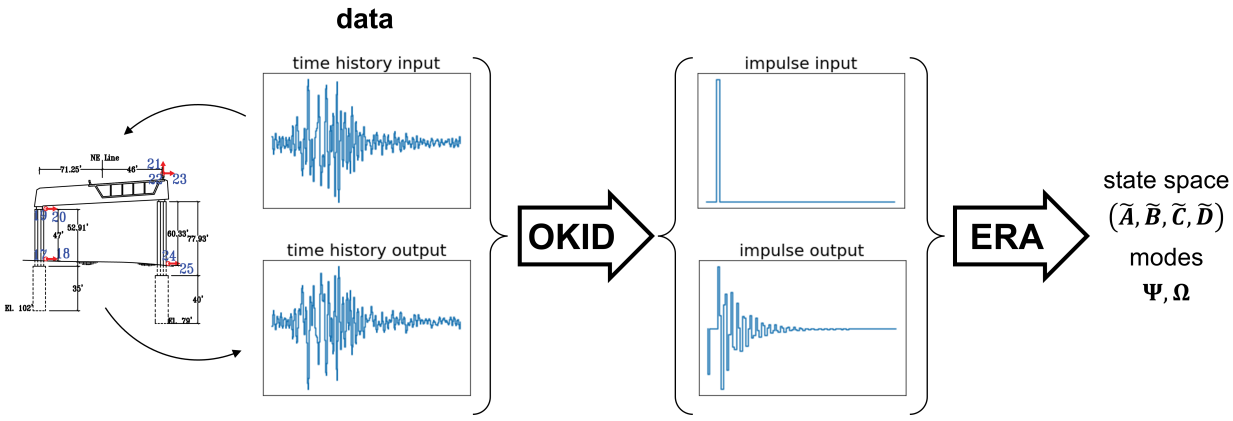


Fig. 67: Observer Kalman Filter Identification – Eigensystem Realization Algorithm (OKID-ERA).

When noise is incorporated into the discrete LTI state-space representation of a structural system, it becomes a *linear Gaussian model* of a *hidden Markov process*, refer to Section 13.5.1. Because the data are assumed to follow a linear Gaussian model, Kalman filtering can estimate an impulse response that is most consistent with the input-output data (Section 13.5.2). The estimated model after filtering is the same as that used by the Ho-Kalman algorithm, i.e., Eq. (13.9).

The state-space evolution includes terms transforming the input at all previous time-points:

$$\begin{aligned}
&\mathbf{z}_0, \mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_k \;\text{:=given input,}\\
&\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_k = \mathbf{0}, (\mathbf{B}\mathbf{z}_0), (\mathbf{A}\mathbf{B}\mathbf{z}_0 + \mathbf{B}\mathbf{z}_1), \cdots, (\mathbf{A}^{k-1}\mathbf{B}\mathbf{z}_0 + \mathbf{A}^{k-2}\mathbf{B}\mathbf{z}_1 + \cdots + \mathbf{B}\mathbf{z}_{k-1}),\\
&\mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_k = \mathbf{D}\mathbf{z}_0, (\mathbf{C}\mathbf{B}\mathbf{z}_0 + \mathbf{D}\mathbf{z}_1), (\mathbf{C}\mathbf{A}\mathbf{B}\mathbf{z}_0 + \mathbf{C}\mathbf{B}\mathbf{z}_1 + \mathbf{D}\mathbf{z}_2), \cdots,\\
&\qquad\qquad (\mathbf{C}\mathbf{A}^{k-1}\mathbf{B}\mathbf{z}_0 + \mathbf{C}\mathbf{A}^{k-2}\mathbf{B}\mathbf{z}_1 + \cdots + \mathbf{D}\mathbf{z}_k).
\end{aligned} \tag{13.33}$$

The output data is expressed in terms of the Markov parameters and an upper triangular *data matrix* $\mathbf{Z}$ built from the input data (subscript δ indicates that the response comes from an impulse input):

$$\underbrace{\begin{bmatrix}\mathbf{y}_0 & \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_m\end{bmatrix}}_{\mathbf{Y}} = \underbrace{\begin{bmatrix}\mathbf{y}_0 & \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_m\end{bmatrix}_\delta}_{\mathbf{Y}_\delta} \underbrace{\begin{bmatrix}\mathbf{z}_0 & \mathbf{z}_1 & \cdots & \mathbf{z}_m\\ \mathbf{0} & \mathbf{z}_0 & \cdots & \mathbf{z}_{m-1}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{z}_0\end{bmatrix}}_{\mathbf{Z}}, \tag{13.34}$$

where m is the number of Markov parameters after $\mathbf{y}_{0_\delta}$ to compute, a configuration option chosen prior to executing the algorithm. It is noted that inverting $\mathbf{Z}$ is often computationally expensive or ill-conditioned. The Kalman filter is applied by augmenting the system with the outputs $\mathbf{y}_i$ to form the *augmented data matrix* $\mathbf{W}$:

$$\mathbf{W} = \begin{bmatrix} \mathbf{z}_0 & \mathbf{z}_1 & \cdots & \mathbf{z}_l & \cdots & \mathbf{z}_m \\ \mathbf{0} & \mathbf{v}_0 & \cdots & \mathbf{v}_{l-1} & \cdots & \mathbf{v}_{m-1} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{v}_0 & \cdots & \mathbf{v}_{m-l} \end{bmatrix}, \qquad \mathbf{v}_i = \begin{bmatrix} \mathbf{z}_i \\ \mathbf{y}_i \end{bmatrix}. \tag{13.35}$$

The Markov parameters (i.e., the impulse response) can be estimated as a function of the input and output data as follows,

$$\hat{\mathbf{Y}}_\delta = \mathbf{Y}\mathbf{W}^+, \tag{13.36}$$

where the superscript + indicates the *pseudo-inverse*[21]. Subsequently, one extracts the estimated, or *observer*, Markov parameters from the block columns of $\hat{\mathbf{Y}}_\delta$, as follows,

$$\begin{aligned} &\hat{\mathbf{Y}}_{0_\delta} \in \mathbb{R}^{p\times q}, \\ &\hat{\mathbf{Y}}_{k_\delta} = \begin{bmatrix} \hat{\mathbf{Y}}^{(1)}_{k_\delta} & \hat{\mathbf{Y}}^{(2)}_{k_\delta} \end{bmatrix}, \quad k \in [1, 2, \cdots], \\ &\hat{\mathbf{Y}}^{(1)}_{k_\delta} \in \mathbb{R}^{p\times q}, \quad \hat{\mathbf{Y}}^{(2)}_{k_\delta} \in \mathbb{R}^{p\times p}. \end{aligned} \tag{13.37}$$

Finally, one reconstructs the system Markov parameters, as follows,

$$\mathbf{y}_{0_\delta} = \hat{\mathbf{Y}}_{0_\delta} = \mathbf{D}, \quad \mathbf{y}_{k_\delta} = \hat{\mathbf{Y}}^{(1)}_{k_\delta} + \sum_{i=1}^{k} \hat{\mathbf{Y}}^{(2)}_{k_\delta} \mathbf{y}_{(k-i)_\delta}. \tag{13.38}$$

13.1.5 OKID-ERA with Data Correlations (OKID-ERA-DC)

Additional data correlations can be factored into the system realization using the OKID-ERA with Data Correlations algorithm (OKID-ERA-DC) [36]. This method is intended to reduce bias due to noise. The method is an "add-on" to the OKID-ERA algorithm, where the reduced order model of Eq. (13.32) can be computed using a Hankel matrix constructed using the successive correlations between "lagged" Hankel matrices, H_k and H_{k+l}, with k representing the time-step and l representing the "lag," a configuration option chosen prior to executing the algorithm. For full details of the algorithm, refer to Juang et al., 1987 [36].

13.1.6 System Realization Using Information Matrix (SRIM)

For discrete-time systems, the correlations between inputs, outputs, and state yield information about the system's state evolution and response. In fact, the state equations can be estimated by manipulating correlation matrices through the method of System Realization using Information Matrix (SRIM) [35].

[21] In linear algebra, the Moore-Penrose inverse of a matrix, often called the pseudo-inverse, is the generalization of the inverse matrix. The pseudo-inverse of a $m \times n$ matrix A is expressed from the SVD of A.

13.1.6.1 Discrete-time system matrix equation

The discrete-time state equations that correspond to the structure's dynamics are restated below from Eq. (13.9) using arguments instead of subscripts:

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{z}(k),\\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{z}(k).\end{aligned}$$

Noting the following state evolution:

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{z}(k),\\ \mathbf{x}(k+2) &= \mathbf{A}^2\mathbf{x}(k) + \mathbf{A}\mathbf{B}\mathbf{z}(k) + \mathbf{B}\mathbf{z}(k+1),\\ \mathbf{x}(k+3) &= \mathbf{A}^3\mathbf{x}(k) + \mathbf{A}^2\mathbf{B}\mathbf{z}(k) + \mathbf{A}\mathbf{B}\mathbf{z}(k+1) + \mathbf{B}\mathbf{z}(k+2),\end{aligned}$$

the response for the time-point $(k+p-1)$ is generalized, as follows,

$$\begin{aligned}\mathbf{x}(k+p) &= \mathbf{A}^p\mathbf{x}(k) + \sum_{i=1}^{p}\mathbf{A}^{p-i}\mathbf{B}\mathbf{z}(k+i-1),\\ \mathbf{x}(k+p-1) &= \mathbf{A}^{p-1}\mathbf{x}(k) + \sum_{i=1}^{p-1}\mathbf{A}^{p-i-1}\mathbf{B}\mathbf{z}(k+i-1),\\ \mathbf{y}(k+p-1) &= \mathbf{C}\mathbf{A}^{p-1}\mathbf{x}(k) + \sum_{i=1}^{p-1}\mathbf{C}\mathbf{A}^{p-i-1}\mathbf{B}\mathbf{z}(k+i-1) + \mathbf{D}\mathbf{z}(k+p-1).\end{aligned}$$

The p successive time-points are stacked into a column vector $\mathbf{y}_p(k)$, as follows,

$$\begin{aligned}\mathbf{y}_p(k) &= \mathcal{O}_p\mathbf{x}(k) + \mathcal{T}_p\mathbf{z}_p(k),\\ \begin{bmatrix}\mathbf{y}(k)\\ \mathbf{y}(k+1)\\ \vdots\\ \mathbf{y}(k+p-1)\end{bmatrix} &= \begin{bmatrix}\mathbf{C}\\ \mathbf{C}\mathbf{A}\\ \mathbf{C}\mathbf{A}^2\\ \vdots\\ \mathbf{C}\mathbf{A}^{p-1}\end{bmatrix}\mathbf{x}(k) + \begin{bmatrix}\mathbf{D} & & & & \\ \mathbf{C}\mathbf{B} & \mathbf{D} & & & \\ \mathbf{C}\mathbf{A}\mathbf{B} & \mathbf{C}\mathbf{B} & \mathbf{D} & & \\ \vdots & \vdots & \vdots & \ddots & \\ \mathbf{C}\mathbf{A}^{p-2}\mathbf{B} & \mathbf{C}\mathbf{A}^{p-3}\mathbf{B} & \mathbf{C}\mathbf{A}^{p-4}\mathbf{B} & \cdots & \mathbf{D}\end{bmatrix}\begin{bmatrix}\mathbf{z}(k)\\ \mathbf{z}(k+1)\\ \vdots\\ \mathbf{z}(k+p-1)\end{bmatrix}.\end{aligned} \tag{13.39}$$

The matrix $\mathcal{O}_p$ is the p^{th}-order observability matrix and the matrix $\mathcal{T}_p$ is a Toeplitz matrix (Section 13.4.2). Finally, we horizontally stack N successive time-points of the column vectors $\mathbf{y}_p$ in a matrix to obtain the following matrix equation:

$$\mathbf{Y}_p(k) = \mathcal{O}_p\mathbf{X}(k) + \mathcal{T}_p\mathbf{Z}_p(k), \tag{13.40}$$

where

$$\begin{aligned}
\mathbf{Y}_p(k) &= \begin{bmatrix} \mathbf{y}_p(k) & \mathbf{y}_p(k+1) & \cdots & \mathbf{y}_p(k+N-1) \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{y}(k) & \mathbf{y}(k+1) & \cdots & \mathbf{y}(k+N-1) \\ \mathbf{y}(k+1) & \mathbf{y}(k+2) & \cdots & \mathbf{y}(k+N) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}(k+p-1) & \mathbf{y}(k+p) & \cdots & \mathbf{y}(k+N+p-2) \end{bmatrix}, \\
\mathbf{X}(k) &= \begin{bmatrix} \mathbf{x}(k) & \mathbf{x}(k+1) & \cdots & \mathbf{x}(k+N-1) \end{bmatrix}, \\
\mathbf{Z}_p(k) &= \begin{bmatrix} \mathbf{z}_p(k) & \mathbf{z}_p(k+1) & \cdots & \mathbf{z}_p(k+N-1) \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{z}(k) & \mathbf{z}(k+1) & \cdots & \mathbf{z}(k+N-1) \\ \mathbf{z}(k+1) & \mathbf{z}(k+2) & \cdots & \mathbf{z}(k+N) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{z}(k+p-1) & \mathbf{z}(k+p) & \cdots & \mathbf{z}(k+N+p-2) \end{bmatrix}.
\end{aligned}$$

13.1.6.2 Observability matrix from information matrix

By post-multiplying Eq. (13.40) by $\frac{1}{N}\mathbf{Z}_p^T(k)$, $\frac{1}{N}\mathbf{Y}_p^T(k)$, or $\frac{1}{N}\mathbf{X}_p^T(k)$, we obtain relationships between correlation matrices $\mathbf{R}_{yy}$, $\mathbf{R}_{yz}$, $\mathbf{R}_{zz}$, and $\mathbf{R}_{xx}$ (derivation of this relationship in Eq. (13.41) is given in Section 13.1.6.4).

$$\mathbf{R}_{yy} - \mathbf{R}_{yz}\mathbf{R}_{zz}^{-1}\mathbf{R}_{yz}^T = \mathcal{O}_p\mathbf{R}_{xx}\mathcal{O}_p^T, \tag{13.41}$$

where

$$\begin{aligned}
\mathbf{R}_{yy} &= \frac{1}{N}\mathbf{Y}_p(k)\mathbf{Y}_p^T(k), \quad \mathbf{R}_{yz} = \frac{1}{N}\mathbf{Y}_p(k)\mathbf{Z}_p^T(k), \\
\mathbf{R}_{zz} &= \frac{1}{N}\mathbf{Z}_p(k)\mathbf{Z}_p^T(k), \quad \mathbf{R}_{xx} = \frac{1}{N}\mathbf{X}(k)\mathbf{X}^T(k).
\end{aligned} \tag{13.42}$$

The left-hand side of Eq. (13.41) is found from the input and output measurements. It is called the *information matrix* of the data and its SVD (because the information matrix is symmetric, this is exactly its orthogonal decomposition) yields the *observability matrix* $\mathcal{O}_p$, i.e.,

$$\mathbf{R}_{yy} - \mathbf{R}_{yz}\mathbf{R}_{zz}^{-1}\mathbf{R}_{yz}^T = \mathbf{U}\Sigma\mathbf{U}^T = \mathcal{O}_p\mathbf{R}_{xx}\mathcal{O}_p^T. \tag{13.43}$$

13.1.6.3 State equation matrices from observability matrix

The state equation matrices $\mathbf{A}$ and $\mathbf{C}$ can be obtained from the observability matrix $\mathcal{O}_p$, as follows,

$$\mathcal{O}_p = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{p-1} \end{bmatrix}, \quad \mathcal{O}_p(:-1) = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{p-2} \end{bmatrix}, \quad \mathcal{O}_p(1:) = \begin{bmatrix} \mathbf{CA} \\ \mathbf{CA}^2 \\ \mathbf{CA}^3 \\ \vdots \\ \mathbf{CA}^{p-1} \end{bmatrix}, \tag{13.44}$$

$$\mathbf{A} = \mathcal{O}_p(:-1)^{+}\mathcal{O}_p(1:), \tag{13.45}$$

$$\mathbf{C} = \mathcal{O}_p(0), \tag{13.46}$$

where $(:-1)$ indicates all but the last block row, $(1:)$ indicates all but the first block row, and (0) indicates only the first block row. Examining the block rows of $\mathcal{O}_p$, it can be seen that the first block row of $\mathcal{O}_p$ is exactly $\mathbf{C}$. By examining $\mathcal{O}_p(:-1)$ and $\mathcal{O}_p(1:)$, one obtains:

$$\mathcal{O}_p(:-1)^{+}\mathcal{O}_p(1:) = \left(\mathbf{C}\begin{bmatrix}\mathbf{I}\\ \mathbf{A}\\ \mathbf{A}^2\\ \vdots\\ \mathbf{A}^{p-2}\end{bmatrix}\right)^{+}\mathbf{C}\begin{bmatrix}\mathbf{I}\\ \mathbf{A}\\ \mathbf{A}^2\\ \vdots\\ \mathbf{A}^{p-2}\end{bmatrix}\mathbf{A} = \begin{bmatrix}\mathbf{I}\\ \mathbf{A}\\ \mathbf{A}^2\\ \vdots\\ \mathbf{A}^{p-2}\end{bmatrix}^{+}\mathbf{C}^{+}\mathbf{C}\begin{bmatrix}\mathbf{I}\\ \mathbf{A}\\ \mathbf{A}^2\\ \vdots\\ \mathbf{A}^{p-2}\end{bmatrix}\mathbf{A} = \mathbf{A}. \tag{13.47}$$

13.1.6.4 Manipulation of discrete-time system matrix equation into the correlation matrix relationship

In [35], the discrete-time system matrix equation, i.e., Eq. (13.40), is manipulated into a form describing the relationship between the correlation matrices $\mathbf{R}_{yy}$, $\mathbf{R}_{yz}$, $\mathbf{R}_{zz}$, and $\mathbf{R}_{xx}$. This summarized in the following paragraphs.

Post-multiplying the discrete-time system matrix equation Eq. (13.40) by $\frac{1}{N}\mathbf{Z}_p^T(k)$:

$$\begin{aligned}\frac{1}{N}\mathbf{Y}_p(k)\mathbf{Z}_p^T(k) &= \mathcal{O}_p\frac{1}{N}\mathbf{X}(k)\mathbf{Z}_p^T(k) + \mathcal{T}_p\frac{1}{N}\mathbf{Z}_p(k)\mathbf{Z}_p^T(k),\\ \mathbf{R}_{yz} &= \mathcal{O}_p\mathbf{R}_{xz} + \mathcal{T}_p\mathbf{R}_{zz},\\ \mathcal{T}_p &= \left(\mathbf{R}_{yz} - \mathcal{O}_p\mathbf{R}_{xz}\right)\mathbf{R}_{zz}^{-1}.\end{aligned}$$

Post-multiplying by $\frac{1}{N}\mathbf{Y}_p^T(k)$:

$$\begin{aligned}\frac{1}{N}\mathbf{Y}_p(k)\mathbf{Y}_p^T(k) &= \mathcal{O}_p\frac{1}{N}\mathbf{X}(k)\mathbf{Y}_p^T(k) + \mathcal{T}_p\frac{1}{N}\mathbf{Z}_p(k)\mathbf{Y}_p^T(k),\\ \mathbf{R}_{yy} &= \mathcal{O}_p\mathbf{R}_{yx}^T + \mathcal{T}_p\mathbf{R}_{yz}^T,\\ \mathbf{R}_{yy} &= \mathcal{O}_p\mathbf{R}_{yx}^T + \left(\mathbf{R}_{yz} - \mathcal{O}_p\mathbf{R}_{xz}\right)\mathbf{R}_{zz}^{-1}\mathbf{R}_{yz}^T.\end{aligned}$$

Post-multiplying by $\frac{1}{N}\mathbf{X}_p^T(k)$:

$$\begin{aligned}\frac{1}{N}\mathbf{Y}_p(k)\mathbf{X}_p^T(k) &= \mathcal{O}_p\frac{1}{N}\mathbf{X}(k)\mathbf{X}_p^T(k) + \mathcal{T}_p\frac{1}{N}\mathbf{Z}_p(k)\mathbf{X}_p^T(k),\\ \mathbf{R}_{yx} &= \mathcal{O}_p\mathbf{R}_{xx} + \mathcal{T}_p\mathbf{R}_{xz}^T,\\ \mathbf{R}_{yx} &= \mathcal{O}_p\mathbf{R}_{xx} + \left(\mathbf{R}_{yz} - \mathcal{O}_p\mathbf{R}_{xz}\right)\mathbf{R}_{zz}^{-1}\mathbf{R}_{xz}^T.\end{aligned}$$

Substituting the equation for $\mathbf{R}_{yx}$ into the equation for $\mathbf{R}_{yy}$:

$$\begin{aligned}\mathbf{R}_{yy} &= \mathcal{O}_p\left(\mathcal{O}_p\mathbf{R}_{xx} + \left(\mathbf{R}_{yz} - \mathcal{O}_p\mathbf{R}_{xz}\right)\mathbf{R}_{zz}^{-1}\mathbf{R}_{xz}^T\right)^T + \left(\mathbf{R}_{yz} - \mathcal{O}_p\mathbf{R}_{xz}\right)\mathbf{R}_{zz}^{-1}\mathbf{R}_{yz}^T\\ &= \mathcal{O}_p\mathbf{R}_{xx}\mathcal{O}_p^T + \mathcal{O}_p\mathbf{R}_{xz}\mathbf{R}_{zz}^{-1}\left(\mathbf{R}_{yz}^T - \mathbf{R}_{xz}^T\mathcal{O}_p^T\right) + \left(\mathbf{R}_{yz} - \mathcal{O}_p\mathbf{R}_{xz}\right)\mathbf{R}_{zz}^{-1}\mathbf{R}_{yz}^T\\ &= \mathcal{O}_p\mathbf{R}_{xx}\mathcal{O}_p^T + \mathcal{O}_p\mathbf{R}_{xz}\mathbf{R}_{zz}^{-1}\mathbf{R}_{yz}^T - \mathcal{O}_p\mathbf{R}_{xz}\mathbf{R}_{zz}^{-1}\mathbf{R}_{xz}^T\mathcal{O}_p^T + \mathbf{R}_{yz}\mathbf{R}_{zz}^{-1}\mathbf{R}_{yz}^T - \mathcal{O}_p\mathbf{R}_{xz}\mathbf{R}_{zz}^{-1}\mathbf{R}_{yz}^T\\ &= \mathcal{O}_p\mathbf{R}_{xx}\mathcal{O}_p^T - \mathcal{O}_p\mathbf{R}_{xz}\mathbf{R}_{zz}^{-1}\mathbf{R}_{xz}^T\mathcal{O}_p^T + \mathbf{R}_{yz}\mathbf{R}_{zz}^{-1}\mathbf{R}_{yz}^T.\end{aligned}$$

Moving all of the terms that can be composed with measured data to the left-hand side:

$$\begin{aligned}\mathbf{R}_{yy} - \mathbf{R}_{yz}\mathbf{R}_{zz}^{-1}\mathbf{R}_{yz}^{T} &= \mathcal{O}_p\mathbf{R}_{xx}\mathcal{O}_p^T - \mathcal{O}_p\mathbf{R}_{xz}\mathbf{R}_{zz}^{-1}\mathbf{R}_{xz}^{T}\mathcal{O}_p^T \\ &= \mathcal{O}_p\left(\mathbf{R}_{xx} - \mathbf{R}_{xz}\mathbf{R}_{zz}^{-1}\mathbf{R}_{xz}^{T}\right)\mathcal{O}_p^T.\end{aligned}$$

We make the assumption that all current and future input data is uncorrelated with the current state, i.e., the average of the products $\mathbf{x}(k)\mathbf{z}(k+i)$, $i \in [0, 1, 2, \cdots]$ is zero. This gives:

$$\begin{aligned}\mathbf{R}_{xz} &= \frac{1}{N}\begin{bmatrix} \sum_{j=0}^{N-1}\mathbf{x}(k+j)\mathbf{z}(k+j) \\ \sum_{j=0}^{N-1}\mathbf{x}(k+j)\mathbf{z}(k+j+1) \\ \sum_{j=0}^{N-1}\mathbf{x}(k+j)\mathbf{z}(k+j+2) \\ \vdots \\ \sum_{j=0}^{N-1}\mathbf{x}(k+j)\mathbf{z}(k+j+p-1)\end{bmatrix}^T \\ &= \mathbf{0},\end{aligned}$$

which yields the required results, i.e., Eq. (13.41).

13.1.7 Example Modal Parameter Identification

SDOF simple system Three system identification methods, OKID-ERA, OKID-ERA-DC, and SRIM, are used to identify the modal parameters of a Single Degree Of Freedom (SDOF) system with known input forcing function and analytical solution for the response. The SDOF oscillator shown in Fig. 68 is assigned the properties $m = 1$, $k = 30$, and $\zeta = 0.01$ (with $c = 2\zeta m\omega_n$ and $\omega_n = \sqrt{k/m}$). It is subjected to the following forcing function with known analytical solution for the response (Fig. 69):

$$f(t) = \sin 0.017\omega_n t + \sin 0.14\omega_n t + \sin 0.34\omega_n t. \tag{13.48}$$

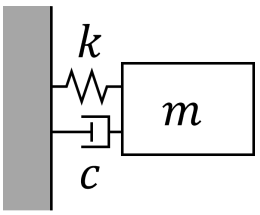


Fig. 68: SDOF oscillator for example modal parameter identification.

System identification methods are used to identify the state-space coefficients and reproduce the response (Fig. 70). Subsequently, the modal parameters in terms of the natural period and damping ratio are identified from each method (Fig. 71). The results from this example demonstrate that the considered three methods and their implementations (refer to Section 13.2) give comparable results to the exact "true" solution, especially the OKID-ERA-DC and SRIM.

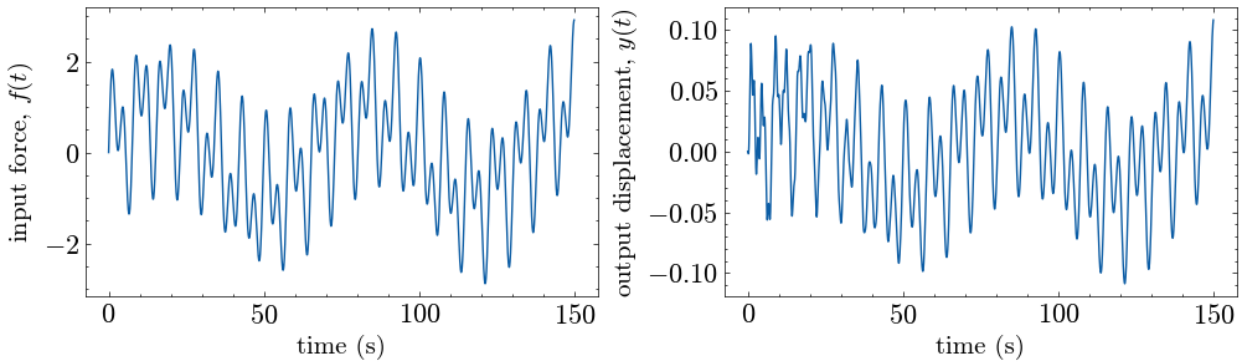


Fig. 69: Forcing function and analytical displacement response for SDOF oscillator example.

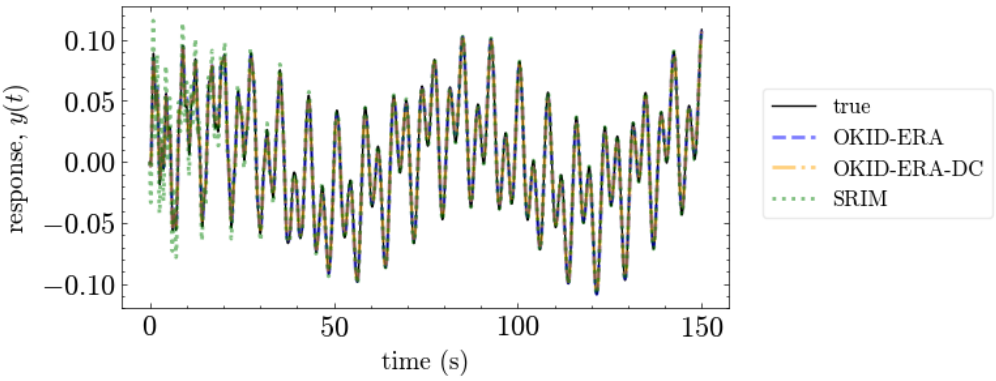


Fig. 70: Response prediction for the SDOF oscillator example.

MDOF structure in the field Five system identification methods, including three state-space methods: OKID-ERA, OKID-ERA-DC, and SRIM, and two input-ouput methods: FSTF and RSTF (refer to Chapter 5), are used to identify the modal parameters of a Multiple Degree Of Freedom (MDOF) system, namely, a bridge structure in the field, the Painter Street Overcrossing in Rio-Dell (Caltrans bridge No. `04-0236`). The transverse natural periods are identified from six events, which occurred between October 2013 and October 2023 (Table 17 and Fig. 72). For the first 5 events in Table 17 with low excitation, i.e., less than 1 g peak recorded acceleration, the identified natural periods are consistent among the events and with literature (Section 6.20). The identified fundamental period is approximately 0.27 seconds. There is a secondary period of approximately 0.23 seconds, and a third, higher mode period, at approximately 0.17 seconds. These are clearly shown in Fig. 72d and its zoomed version Fig. 73, particularly using the SRIM method. For an event with high excitation, i.e., the December 20, 2022 Ferndale earthquake which caused a peak recorded acceleration of 1.38 g, the identified natural periods are elongated, at approximately 0.49 and 0.32 seconds (refer to Fig. 72f). Caltrans conducted an on-site inspection with the assistance of an Unmanned Aerial Vehicle (UAV) after the event and found that the earthquake caused minor concrete cover spalling, shear key cracking, and a vertical displacement of about 1 inch at the south (acute) corner of the east abutment. No other signs of displacement or damage were observed.

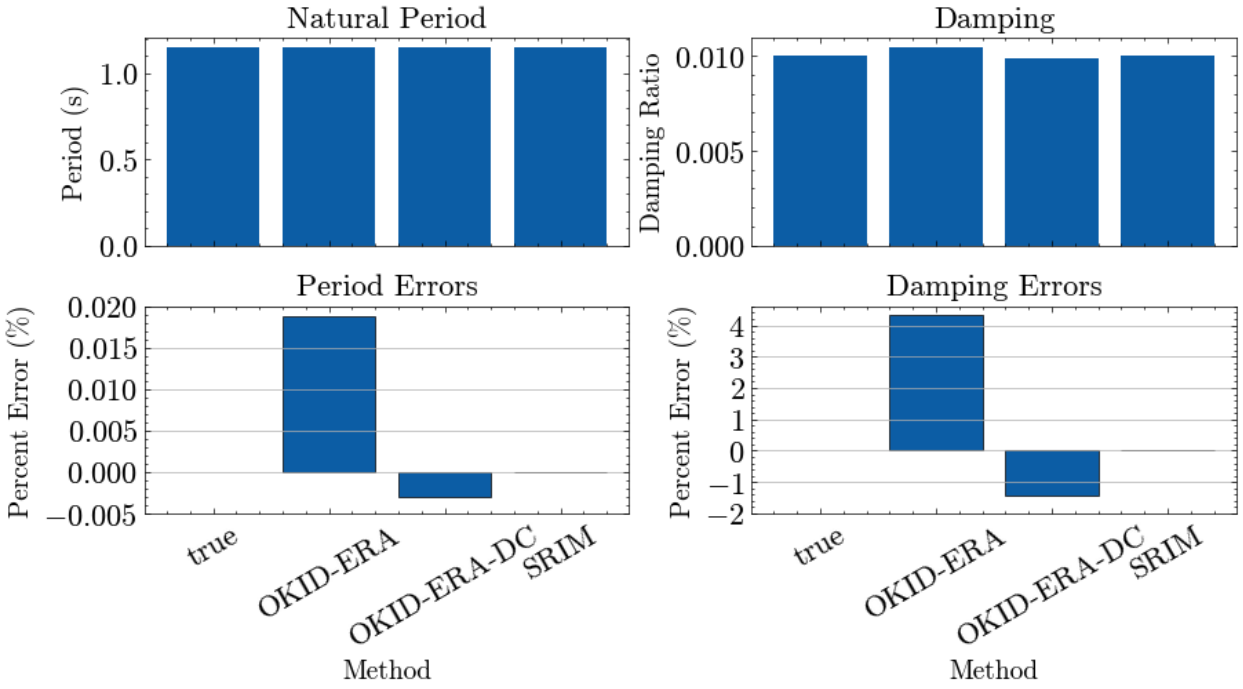
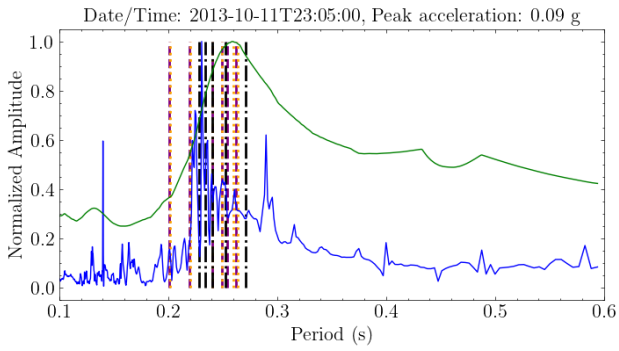


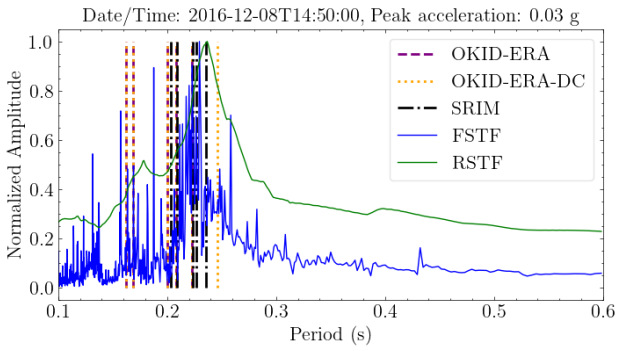
Fig. 71: Period and damping ratio identification for the SDOF oscillator example.

Table 17: First natural period of (s) of the transverse mode identified for Painter Street Overcrossing (Caltrans bridge No. `04-0236`) from six events

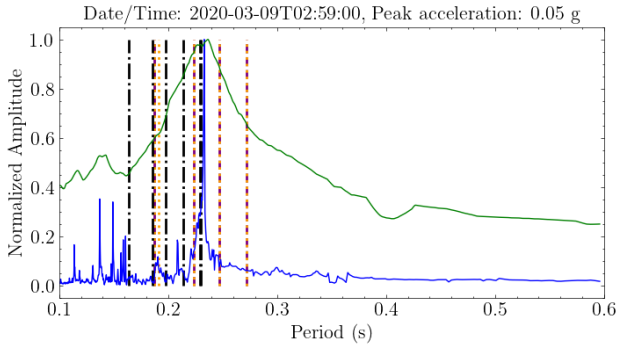
Event Date	**Peak acc. (g)**	**OKID-ERA**	**OKID-ERA-DC**	**SRIM**	**FSTF**	**RSTF**
2013-10-11	0.090	0.262	0.264	0.271	0.290	0.259
2016-12-08	0.031	0.223	0.246	0.236	0.258	0.237
2020-03-09	0.055	0.272	0.272	0.230	0.233	0.237
2023-10-16	0.089	0.248	0.203	0.274	0.257	0.251
2023-09-30	0.038	0.289	0.289	0.267	0.293	0.258
2022-12-20	1.384	0.448	0.448	0.318	0.498	0.550



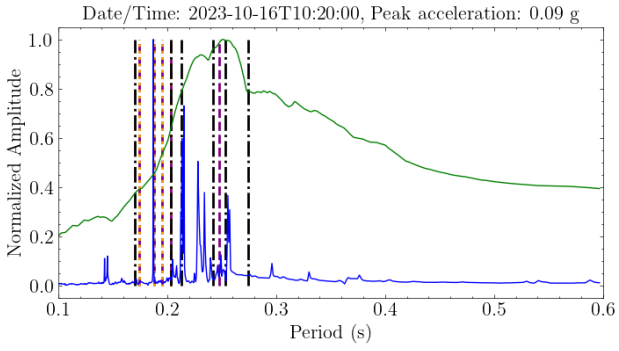
(a) Event of 2013-10-11



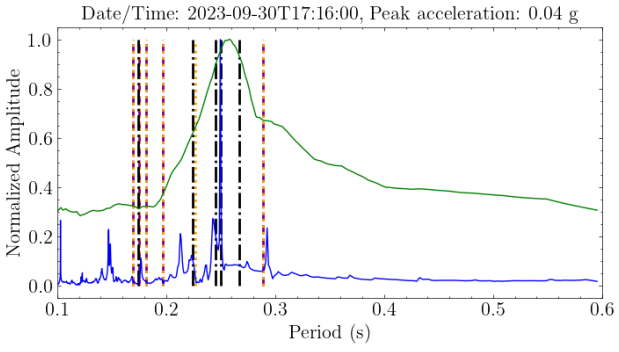
(b) Event of 2016-12-08



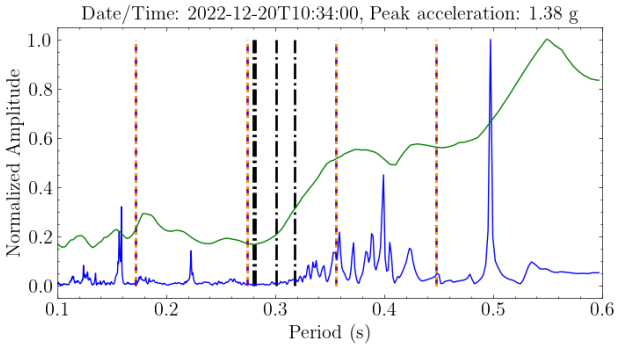
(c) Event of 2020-03-09



(d) Event of 2023-10-16

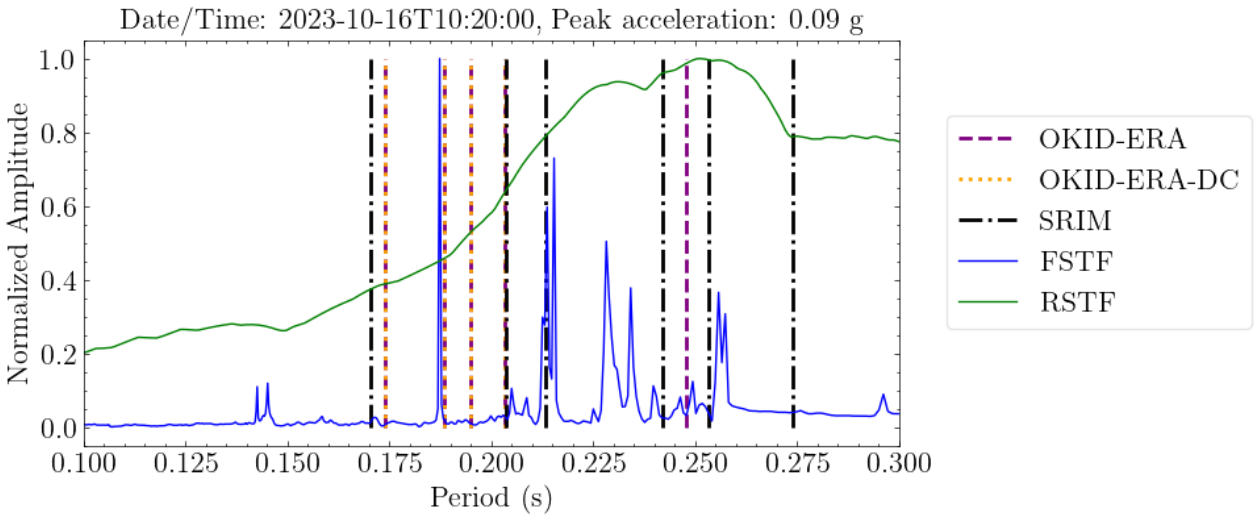


(e) Event of 2023-09-30



(f) High peak acceleration event of 2022-02-20

Fig. 72: Natural period of vibration identified for the Painter Street Overcrossing (Caltrans bridge No. `04-0236`) from six events. The key in Fig. 72b applies to all plots.



.

Fig. 73: Zoomed plot from Fig. 72d of natural period of vibration identified for the event of 2023-10-16 at the Painter Street Overcrossing (Caltrans bridge No. `04-0236`)

13.2 Python Package Reference and Implementation

The *BRACE*^{2} platform uses `mdof` Version 0.0.10. As an actively developed and open-source library, `mdof`'s most current documentation lives on its homepage, https://stairlab.github.io/mdof/, and its project page, https://pypi.org/project/mdof/. The `mdof` Python package is published on the Python Package Index (PyPI) and can be installed simply by executing the following command.

```
pip install mdof
```

13.2.1 Top-Level Functions

Functions that give an at-a-glance view of dynamical characteristics of a system from its vibration data are directly accessible from the top level of `mdof`.

`mdof.sysid(inputs, outputs, **options)` State space system realization from input and output data, Fig. 74.

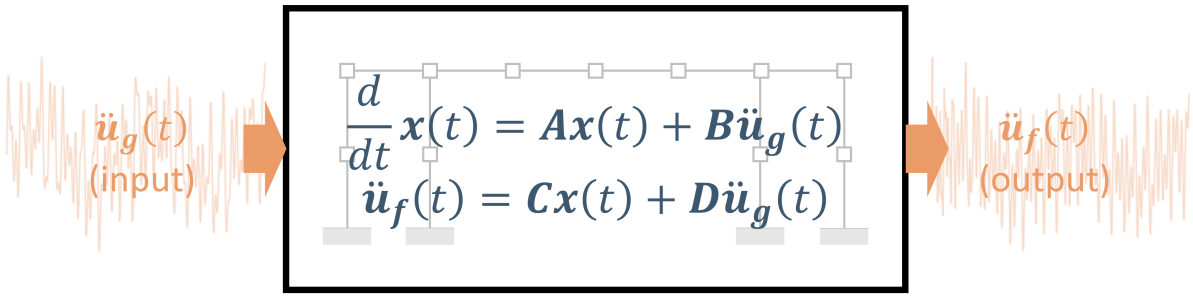


Fig. 74: State Space Realization with `mdof.sysid()`.

Parameters:

- **inputs** (*array*) – input time history. Dimensions: (q, nt), where q = number of inputs and nt = number of time-steps.
- **outputs** (*array*) – output response history. Dimensions: (p, nt), where p = number of outputs and nt = number of time-steps.
- **method** (*string, optional*) – system identification method. Default is "srim", other options are "okid-era" and "okid-era-dc".
- **decimation** (`int`, *optional*) – decimation factor. Default: 8.

Returns: system realization in the form of state space coefficients `(A,B,C,D)`.

Return type: tuple of arrays.

`mdof.eigid(inputs, outputs, **options)` System eigenspace identification from input and output data, Fig. 75.

Parameters:

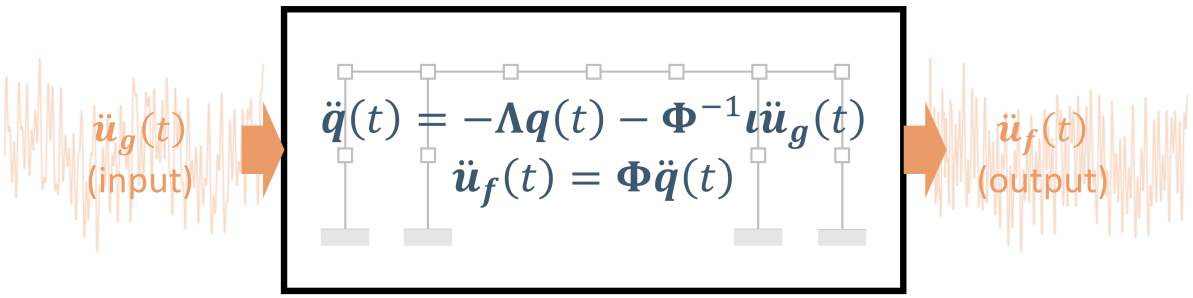


Fig. 75: Inverse Eigenanalysis with `mdof.eigid()`.

- **inputs** (*array*) – input time history. Dimensions: (q, nt), where q = number of inputs and nt = number of time-steps.
- **outputs** (*array*) – output response history. Dimensions: (p, nt), where p = number of outputs and nt = number of time-steps.
- **dt** (`float`) – time-step.
- **decimation** (`int`, *optional*) – decimation factor. Default: 1.

Returns: (eigenvalues, eigenvectors).

Return type: tuple of 1D array, ND array.

`mdof.modes(`*`inputs`*`, `*`outputs`*`, `*`dt`*`, `*`**options`*`)` Modal identification using state space system realization from input and output data, Fig. 76.

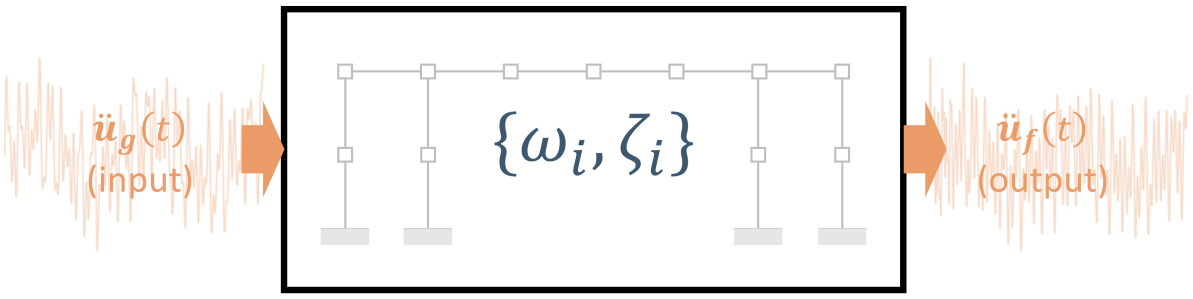


Fig. 76: Modal estimation with `mdof.modes()`.

Parameters:

- **inputs** (*array*) – input time history. Dimensions: (q, nt), where q = number of inputs and nt = number of time-steps.
- **outputs** (*array*) – output response history. Dimensions: (p, nt), where p = number of outputs and nt = number of time-steps.
- **dt** (`float`) – time-step.

- **method** (*string, optional*) – system identification method. Default is "srim", other options are "okid-era", and "okid-era-dc".
- **decimation** (`int`, *optional*) – decimation factor. default: 8.

Returns: system modes, including natural frequencies, damping ratios, mode shapes, condition numbers, and modal validation metrics EMAC and MPC.

Return type: dictionary.

13.2.2 Modules

The rest of the `mdof` library is organized into modules that perform different steps of system identification, including impulse response estimation (`mdof.markov`), state space realization (`mdof.realize`), modal parameter estimation (`mdof.modal`), and frequency domain transformations (`mdof.transform`).

13.2.2.1 `mdof.markov` module

`mdof.markov.okid(inputs, outputs, **options)` Identify Markov parameters or discrete impulse response data for a given set of input and output data using Observer Kalman Identification Algorithm (OKID) [39].

Parameters:

- **inputs** (*array*) – input time history. Dimensions: (q, nt), where q = number of inputs and nt = number of time-steps.
- **outputs** (*array*) – output response history. Dimensions: (p, nt), where p = number of outputs and nt = number of time-steps.
- **m** (`int`, *optional*) – number of Markov parameters to compute. Default: $min(300, nt)$.

Returns: the Markov parameters, with dimensions $(p, q, m + 1)$.

Return type: array.

13.2.2.2 `mdof.realize` module

`mdof.realize.era(Y, **options)` System realization from Markov parameters (discrete impulse response data) using Ho-Kalman [29]/Eigensystem Realization Algorithm (ERA) [37].

Parameters:

- **Y** (*array*) – Markov parameters. Dimensions: (p, q, nt), where p = number of outputs, q = number of inputs, and nt = number of Markov parameters.
- **horizon** (`int`, *optional*) – number of block rows in Hankel matrix = order of observability matrix. Default: $min(150, (nt - 1)/2)$.
- **nc** (`int`, *optional*) – number of block columns in Hankel matrix = order of controllability matrix. Default: $min(150, max(nt - 1 -$ `horizon`$, (nt - 1)/2))$.
- **order** (`int`, *optional*) – model order. Default: $min(20,$ `horizon`$/2)$.

Returns: realization in the form of state space coefficients `(A,B,C,D)`.

Return type: tuple of arrays.

`mdof.realize.era_dc(Y, **options)` System realization from Markov parameters (discrete impulse response data) using Eigensystem Realization Algorithm with Data Correlations (ERA/DC) [36].

Parameters:

- **Y** (*array*) – Markov parameters. Dimensions: (p, q, nt), where p = number of outputs, q = number of inputs, and nt = number of Markov parameters.
- **horizon** (`int`, *optional*) – number of block rows in Hankel matrix = order of observability matrix. Default: $min(150, (nt-1)/2)$.
- **nc** (`int`, *optional*) – number of block columns in Hankel matrix = order of controllability matrix. Default: $min(150, max(nt-1-$ `horizon`$, (nt-1)/2))$.
- **order** (`int`, *optional*) – model order. Default: $min(20,$ `horizon`$/2)$.
- **a** (`int`, *optional*) – (α) number of block rows in Hankel of correlation matrix. Default: 0.
- **b** (`int`, *optional*) – (β) number of block columns in Hankel of correlation matrix. Default: 0.
- **l** (`int`, *optional*) – initial lag for data correlations. Default: 0.
- **g** (`int`, *optional*) – lags (gap) between correlation matrices. Default: 1.

Returns: realization in the form of state space coefficients `(A,B,C,D)`.

Return type: tuple of arrays.

`mdof.realize.srim(inputs, outputs, **options)` System realization from input and output data, with output error minimization method using System Realization Using Information Matrix (SRIM) [35].

Parameters:

- **inputs** (*array*) – input time history. Dimensions: (q, nt), where q = number of inputs and nt = number of time-steps.
- **outputs** (*array*) – output response history. Dimensions: (p, nt), where p = number of outputs, and nt = number of time-steps.
- **horizon** (`int`, *optional*) – number of steps used for identification (prediction horizon). Default: $min(300, nt)$.
- **order** (`int`, *optional*) – model order. Default: $min(20,$ `horizon`$/2)$.
- **full** (`bool`, *optional*) – if True, full SVD. Default: True.
- **find** (*string, optional*) – "ABCD" or "AC". Default: "ABCD".
- **threads** (`int`, *optional*) – number of threads used during the output error minimization method. Default: 6.
- **chunk** (`int`, *optional*) – chunk size in output error minimization method. Default: 200.

Returns: realization in the form of state space coefficients `(A,B,C,D)`.

Return type: tuple of arrays.

13.2.2.3 mdof.modal module

mdof.modal.system_modes(*realization*, *dt*, ***options*) Modal identification from a state space system realization.

Parameters:

- **realization** (*tuple*) – realization in the form of state space coefficients `(A,B,C,D)`.
- **dt** (`float`) – time-step.
- **decimation** (`int`, *optional*) – decimation factor. Default: 1.
- **Observability** (*array, optional*) – Observability matrix; can be reused from `mdof.realize.srim()`. Default: None.

Returns: system modes, including natural frequencies, damping ratios, mode shapes, condition numbers, and modal validation metrics EMAC and MPC.

Return type: dictionary.

mdof.modal.spectrum_modes(*periods*, *amplitudes*, ***options*) Modal identification from a transfer function.

Parameters:

- **periods** (*array*) – transfer function periods.
- **amplitudes** (*array*) – transfer function amplitudes.

Returns: (fundamental_periods, fundamental_amplitudes).

Return type: tuple.

13.2.2.4 mdof.transform module

mdof.transform.power_transfer(*inputs*, *outputs*, *step*, ***options*) Power spectrum transfer function from input and output data.

Parameters:

- **inputs** (*array*) – input time history. Dimensions: (q, nt), where q = number of inputs and nt = number of time-steps.
- **outputs** (*array*) – output response history. Dimensions: (p, nt), where p = number of outputs and nt = number of time-steps.
- **step** (`float`) – time-step.
- **decimation** (`int`, *optional*) – decimation factor. Default: 1.

Returns: (periods, amplitudes).

Return type: tuple of arrays.

mdof.transform.response_transfer(*inputs*, *outputs*, *step*, ***options*) Response spectrum transfer function from input and output data.

Parameters:

- **inputs** (*array*) – input time history. Dimensions: (q, nt), where q = number of inputs and nt = number of time-steps.
- **outputs** (*array*) – output response history. Dimensions: (p, nt), where p = number of outputs and nt = number of time-steps.
- **step** (`float`) – time-step.
- **pseudo** (`bool`, *optional*) – if True, uses pseudo accelerations. Default: False.
- **decimation** (`int`, *optional*) – decimation factor. Default: 1.

Returns: (periods, amplitudes).

Return type: tuple of arrays.

mdof.transform.fourier_transfer(*inputs*, *outputs*, *step*, ***options*) Fourier spectrum transfer function from input and output data.

Parameters:

- **inputs** (*array*) – input time history. Dimensions: (q, nt), where q = number of inputs and nt = number of time-steps.
- **outputs** (*array*) – output response history. Dimensions: (p, nt), where p = number of outputs and nt = number of time-steps.
- **step** (`float`) – time-step.
- **decimation** (`int`, *optional*) – decimation factor. Default: 1.

Returns: (periods, amplitudes).

Return type: tuple of arrays.

mdof.transform.power_spectrum(*series*, *step*, *period_band=None*, ***options*) Power spectrum of a signal, as a function of period (i.e., periodogram).

Parameters:

- **series** (*1D array*) – time series.
- **step** (`float`) – time-step.
- **period_band** (*tuple, optional*) – minimum and maximum period of interest, in seconds.

Returns: (periods, amplitudes).

Return type: tuple of arrays.

mdof.transform.fourier_spectrum(*series*, *step*, *period_band=None*, ***options*) Fourier amplitude spectrum of a signal, as a function of period.

Parameters:

- **series** (*1D array*) – time series.
- **step** (`float`) – time-step.
- **period_band** (*tuple, optional*) – minimum and maximum period of interest, in seconds.

Returns: (periods, amplitudes).

Return type: tuple of arrays.

13.3 Appendix 13.A: Structural Dynamics Glossary

c MDOF structural damping matrix. 105

m MDOF structural mass matrix. 105

k MDOF structural stiffness matrix. 105

$\ddot{\mathbf{u}}_f$ structural (free DOF) acceleration. 104

$\ddot{\mathbf{u}}_g$ support (ground) acceleration. 104

z input vector to structural system. 104, 106

A discrete state transition matrix. 107, 112

$\mathbf{A}_c$ continuous state transition matrix. 105

B discrete input influence matrix. 107, 112

$\mathbf{B}_c$ continuous input influence matrix. 105

C output influence matrix. 105, 112

D direct transmission or feed-through matrix. 105, 112

ι influence vector/matrix; static displacements due to unit ground displacement. 105

$\tilde{\mathbf{A}}$ reduced order discrete state transition matrix. 112

$\tilde{\mathbf{B}}$ reduced order discrete input influence matrix. 112

$\tilde{\mathbf{C}}$ reduced order output influence matrix. 112

$\tilde{\mathbf{D}}$ reduced order direct transmission or feed-through matrix. 112

y output vector of structural system. 104, 106

x state vector of structural system. 104, 105

$\dot{\mathbf{x}}$ time derivative of state vector. 104, 106

c SDOF structural damping. 104

k SDOF structural stiffness. 104

m SDOF structural mass. 104

13.4 Appendix 13.B: Special Matrices

13.4.1 Hankel Matrix

A square matrix where each ascending skew-diagonal from left to right is constant: $A_{i,j} = A_{i+k,j-k} \ \forall k = 0, \ldots, j - i \ \ | i \leq j$.

Example:

$$\begin{bmatrix} a & b & c & d & e \\ b & c & d & e & f \\ c & d & e & f & g \\ d & e & f & g & h \\ e & f & g & h & i \end{bmatrix}.$$

General form:

$$\begin{bmatrix} a_0 & a_1 & \cdots & a_{n-1} \\ a_1 & a_2 & \cdots & \vdots \\ \vdots & \vdots & \ddots & a_{2n-3} \\ a_{n-1} & a_n & \cdots & a_{2n-2} \end{bmatrix}.$$

A Hankel matrix H is a row-reversed or column-reversed Toeplitz matrix. H has the property $H = TJ_n$, where T is a Toeplitz matrix and J_n is an exchange matrix of the same dimension as H. If T is real symmetric, then H will have the same eigenvalues as T up to a sign.

13.4.2 Toeplitz Matrix

A diagonal-constant (not necessarily square) matrix. Each descending diagonal from left to right is constant: $A_{i,j} = A_{i+1,j+1} = a_{i-j} \ \ | \ \ a_{1-n}, \ldots, a_{n-1}$.

Example:

$$\begin{bmatrix} a & b & c & d & e \\ f & a & b & c & d \\ g & f & a & b & c \\ h & g & f & a & b \end{bmatrix}.$$

General form of an $n \times n$ Toeplitz matrix:

$$\begin{bmatrix} a_0 & a_{-1} & \cdots & a_{-(n-2)} & a_{-(n-1)} \\ a_1 & a_0 & \ddots & & a_{-(n-2)} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ a_{n-2} & & \ddots & a_0 & a_{-1} \\ a_{n-1} & a_{n-2} & \cdots & a_1 & a_0 \end{bmatrix}.$$

Toeplitz matrices can be added in $O(n)$ time and multiplied in $O(n^2)$ time. A positive semi-definite $n \times n$ Toeplitz matrix of rank $r < n$ can be uniquely factored as VDV^H.

13.4.3 Exchange Matrix

It is also known as backward identity or reversal matrix. It is a permutation matrix where the anti-diagonal is all 1 and all other elements are 0.

General form:

$$\begin{bmatrix} 0 & 0 & \cdots & 0 & 0 & 1 \\ 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 0 & \cdots & 1 & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 1 & \cdots & 0 & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 & 0 \end{bmatrix}.$$

13.5 Appendix 13.C: State Estimation

13.5.1 Hidden Markov Processes

A system model with sources of randomness can be represented as a *hidden Markov process*: a latent, i.e., inferred, not observed, state evolution over time where, given the current state X_t, the next state X_{t+1} is independent (symbolically represented with $\perp$) of all the previous states. Herein, the states are assumed to be discrete and finite-dimensional. Moreover, if a state evolution follows 13.49, then it is called *Markovian* or *memoryless*.

$$X_{t+1}|X_t \perp X_{\tau<t}. \tag{13.49}$$

These models also assume *conditional independence of observation data*: given the current state X_t, the current observation Y_t is independent of all other states and observations.

$$Y_t|X_t \perp Y_{\tau\neq t}, X_{\tau\neq t}. \tag{13.50}$$

These conditional independence assumptions yield the Probabilistic Graphical Model (PGM)[22] in Figure 77 where three random variables are defined:

1. The initial state $X_0 \sim p_0(\cdot)$,
2. The states at times $t = 1, 2, \cdots$ $(X_t \mid X_{t-1} = x_{t-1}) \sim p_t(\cdot|x_{t-1})$, and
3. The observations at time $t = 0, 1, 2, \cdots$ $(Y_t \mid X_t = x_t) \sim f_t(\cdot|x_t)$,

and their joint probability distribution is expressed as follows:

$$f_{X_0,X_{1:T},Y_{0:T}}(x_0, x_{1:T}, y_{0:T}) = p_0(x_0) \prod_{t=1}^{T} p_t(x_t|x_{t-1}) \prod_{t=0}^{T} f_t(y_t|x_t). \tag{13.51}$$

It is assumed that the observations are noisy measurements of the hidden states. A common state-space representation is the *linear Gaussian model*,

$$\begin{aligned} \mathbf{X}_t &= \mathbf{F}_t\mathbf{X}_{t-1} + \eta_t, & \eta_t &\sim N(\mathbf{0}, \mathbf{Q}_t), \\ \mathbf{Y}_t &= \mathbf{H}_t\mathbf{X}_t + \epsilon_t, & \epsilon_t &\sim N(\mathbf{0}, \mathbf{R}_t), \end{aligned} \tag{13.52}$$

[22] A probabilistic model for which a graph expresses the conditional dependence structure between random variables. In this case, the PGM is also both a *Markov network* (due to the Markov property) and a *dynamic Bayesian network* (due to the directed edges between nodes).

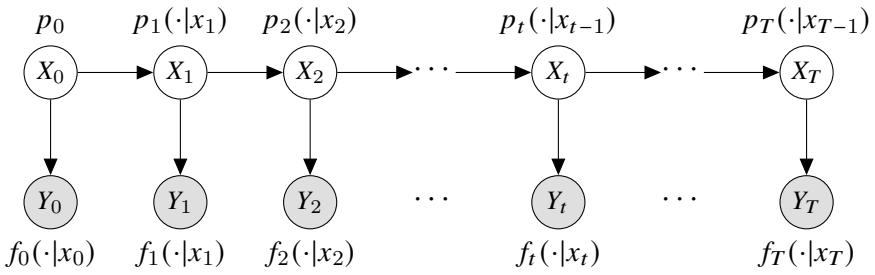


Fig. 77: Probabilistic Graphical Model (PGM) for a hidden Markov process.

where each state $\mathbf{X}_t$ is a linear transformation $\mathbf{F}_t$ of the previous state $\mathbf{X}_{t-1}$ with the addition of Gaussian white noise η_t, or *state disturbance*, and each observation $\mathbf{Y}_t$ is a linear transformation $\mathbf{H}_t$ of the current state $\mathbf{X}_t$ with the addition of Gaussian white noise ϵ_t, or *observation noise*. Here, $\mathbf{X}_t$ and $\mathbf{Y}_t$ are in boldface because they can be multidimensional ($\mathbb{R}^n$) vectors.

State-space representations of time-series allow modeling of the distribution of the observed datapoints, $\mathbf{Y}_t$, and the following tasks can be performed based on the observation data:

1. Filter – estimate the current state $\mathbf{X}_t$.
2. Smooth – estimate the past states $\mathbf{X}_{\tau<t}$.
3. Predict – estimate the future states $\mathbf{X}_{\tau>t}$.

13.5.2 Kalman Filter

State estimation for linear Gaussian hidden Markov models can be performed using Kalman filtering. Beginning with the model in Eq. (13.52) and a Gaussian prior assumption for the state distribution at the first time-step, $\mathbf{X}_0 \sim N(\boldsymbol{\mu}_0, \boldsymbol{\Gamma}_0)$ with mean $\boldsymbol{\mu}_0$ and variance $\boldsymbol{\Gamma}_0$, define the distribution of the state at each time-step, given the observations up to that time-step, as $\mathbf{X}_{t|t} := (\mathbf{X}_t|\mathbf{Y}_{1:t}) \sim N(\mathbf{x}_{t|t}, \mathbf{V}_{t|t})$. It is noted that the mean and variance parameters, namely, $\mathbf{x}_{t|t}$ and $\mathbf{V}_{t|t}$, are the desired outputs of the algorithm. There are two steps to the Kalman filter: (1) the *one-step-ahead prediction* and (2) the *update step*. Starting at $t = 1$, both steps must be performed for each time-step before advancing to the next time-step.

One-step-ahead prediction Assume that the distribution of $(\mathbf{X}_{t-1}|\mathbf{Y}_{1:t-1}) \sim N(\mathbf{x}_{t-1|t-1}, \mathbf{V}_{t-1|t-1})$ is given. This is true at $t = 1$, because $(X_0|Y_{1:0}) = X_0 \sim N(\boldsymbol{\mu}_0, \Gamma_0)$ is given (note that there are no observations Y_t for $t < 1$). After the initial time-step, both the one-step-ahead prediction and the update step are performed to provide $(\mathbf{X}_{t-1}|\mathbf{Y}_{1:t-1})$ for $t \geq 1$. Use the fact that $\mathbf{X}_t$ is dependent on a linear combination of independent Gaussian-distributed variables, Eq. (13.53), and conditionally independent on previous observations (first line of Eq. (13.54)) to compute the distribution parameters of $(\mathbf{X}_t|\mathbf{X}_{t-1}, \mathbf{Y}_{1:t-1})$, namely the mean $\mathbf{x}_{t|t-1}$ and variance $\mathbf{V}_{t|t-1}$ (last two lines of Eq. (13.54)).

$$\mathbf{X}_t = \mathbf{F}_t\mathbf{X}_{t-1} + \eta_t, \quad \eta_t \sim N(\mathbf{0}, \mathbf{Q}_t). \tag{13.53}$$

$$\begin{aligned}(\mathbf{X}_t|\mathbf{X}_{t-1}, \mathbf{Y}_{1:t-1}) &= (\mathbf{X}_t|\mathbf{X}_{t-1}) \sim N(\mathbf{x}_{t|t-1}, \mathbf{V}_{t|t-1}),\\ \mathbf{x}_{t|t-1} &= \mathbf{F}_t\mathbf{x}_{t-1|t-1},\\ \mathbf{V}_{t|t-1} &= \mathbf{F}_t\mathbf{V}_{t|t-1}\mathbf{F}_t^T + \mathbf{Q}_t.\end{aligned} \tag{13.54}$$

Update step In the update step, Bayesian linear regression is used to compute the distribution of $(\mathbf{X}_t|\mathbf{Y}_{1:t})$. The use of Bayesian linear regression requires the definition of a *prior*, a *likelihood*, and a *posterior*. The previous step provides the distribution of the prior, $(\mathbf{X}_t|\mathbf{X}_{t-1}, \mathbf{Y}_{1:t-1})$. The likelihood is $(\mathbf{Y}_t|\mathbf{X}_t, \mathbf{X}_{t-1}, \mathbf{Y}_{1:t-1})$ and its distribution is also known because it is dependent on a linear transformation of the prior, Eq. (13.55). The distribution of the posterior, $(\mathbf{X}_t|\mathbf{Y}_{1:t})$, is the desired result and is computed in a closed form as described in the following derivation.

First, compute the distribution of the likelihood, $(\mathbf{Y}_t|\mathbf{X}_t, \mathbf{X}_{t-1}, \mathbf{Y}_{1:t-1})$, noting that its conditional independence on previous states and observations simplifies it to $(\mathbf{Y}_t|\mathbf{X}_t)$.

$$\begin{aligned}
&\mathbf{Y}_t = \mathbf{H}_t\mathbf{X}_t + \epsilon_t, \quad \epsilon_t \sim N(\mathbf{0}, \mathbf{R}_t),\\
&(\mathbf{Y}_t|\mathbf{X}_t, \mathbf{X}_{t-1}, \mathbf{Y}_{1:t-1}) = (\mathbf{Y}_t|\mathbf{X}_t) \sim N(\mathbf{H}_t\mathbf{X}_t, \mathbf{R}_t).
\end{aligned} \tag{13.55}$$

For simplicity of the notations, the general process of Bayesian linear regression is described with a prior $\mathbf{X}$, a likelihood $(\mathbf{Y}|\mathbf{X})$, and a posterior $(\mathbf{X}|\mathbf{Y})$ in Eqs. (13.56) and (13.57). The Bayes rule states that the distribution function of the posterior, $f_{\mathbf{X}|\mathbf{Y}}$, is proportional to the product of the distribution functions of the likelihood and the prior, i.e., $f_{\mathbf{Y}|\mathbf{X}} \times f_{\mathbf{X}}$. Then, the properties of Gaussian distribution functions and linearity provide the parameters of the posterior distribution, $\mathbf{m}_1$ and $\mathbf{Q}_1$ in Eq. (13.56).

$$\begin{aligned}
&\mathbf{X} \sim N(\mathbf{m}_0, \mathbf{Q}_0), \quad (\mathbf{Y}|\mathbf{X}) \sim N(\mathbf{H}\mathbf{X}, \mathbf{R}), \quad f_{\mathbf{X}|\mathbf{Y}} \propto f_{\mathbf{Y}|\mathbf{X}} \times f_{\mathbf{X}},\\
&(\mathbf{X}|\mathbf{Y}) \sim N(\mathbf{m}_1, \mathbf{Q}_1),\\
&\mathbf{Q}_1 = (\mathbf{I} + \mathbf{Q}_0\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{Q}_0,\\
&\mathbf{m}_1 = (\mathbf{I} + \mathbf{Q}_0\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{m}_0 + \mathbf{Q}_1\mathbf{H}^T\mathbf{R}^{-1}\mathbf{Y}.
\end{aligned} \tag{13.56}$$

Further rearrangement of the posterior variance $\mathbf{Q}_1$ and mean $\mathbf{m}_1$ is a standard practice to simplify the computation of inverses. This is performed using the algebraic result from the Sherman-Morrison-Woodbury formula ([18], Chapter 7), as follows:

$$\begin{aligned}
&(\mathbf{A} + \mathbf{UCV})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{C}^{-1} + \mathbf{V}\mathbf{A}^{-1}\mathbf{U})^{-1}\mathbf{V}\mathbf{A}^{-1},\\
&\mathbf{Q}_1 = \mathbf{Q}_0 - \mathbf{Q}_0\mathbf{H}^T(\mathbf{H}\mathbf{Q}_0\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H}\mathbf{Q}_0,\\
&\mathbf{m}_1 = \mathbf{m}_0 + \mathbf{Q}_0\mathbf{H}^T(\mathbf{H}\mathbf{Q}_0\mathbf{H}^T + \mathbf{R})^{-1}(\mathbf{Y} - \mathbf{H}\mathbf{m}_0).
\end{aligned} \tag{13.57}$$

Plugging in the likelihood from Eq. (13.55) and the prior from Eq. (13.54), the final result of the variance and mean parameters of the posterior, $(\mathbf{X}_t|\mathbf{Y}_{1:t})$, is shown below.

$$\begin{aligned}
&f_{(\mathbf{X}_t|\mathbf{Y}_{1:t})} \propto f_{(\mathbf{Y}_t|\mathbf{X}_t,\mathbf{X}_{t-1},\mathbf{Y}_{1:t-1})} f_{(\mathbf{X}_t|\mathbf{X}_{t-1},\mathbf{Y}_{1:t-1})},\\
&(\mathbf{X}_t|\mathbf{Y}_{1:t}) \sim N(\mathbf{x}_{t|t}, \mathbf{V}_{t|t}),\\
&\mathbf{V}_{t|t} = \mathbf{V}_{t|t-1} - \mathbf{V}_{t|t-1}\mathbf{H}_t^T(\mathbf{H}_t\mathbf{V}_{t|t-1}\mathbf{H}_t^T + \mathbf{R}_t)^{-1}\mathbf{H}_t\mathbf{V}_{t|t-1},\\
&\mathbf{x}_{t|t} = \mathbf{x}_{t|t-1} + \mathbf{V}_{t|t-1}\mathbf{H}_t^T(\mathbf{H}_t\mathbf{V}_{t|t-1}\mathbf{H}_t^T + \mathbf{R}_t)^{-1}(\mathbf{Y}_t - \mathbf{H}_t\mathbf{x}_{t|t-1}).
\end{aligned} \tag{13.58}$$

13.5.3 Autoregressive Models

State estimation can also be performed using autoregressive models, which use regression to estimate the relationship between:

- The observation at the current time-step, $\mathbf{Y_t}$, and

- The observations at a certain number p of previous time-steps, $\mathbf{Y_{t-1}}, \mathbf{Y_{t-2}}, \cdots, \mathbf{Y_{t-p}}$. This number p is the model *order* or number of *autoregressors*.

AR(1): Autoregressive model with order 1 ($p = 1$)

$$Y_t = \phi Y_{t-1} + z_t, \qquad z_t \overset{iid}{\sim} N(0, \sigma^2). \tag{13.59}$$

Note that the notation *iid* stands for "independent and identically distributed," indicating that each random variable has the same probability distribution as the others and all are mutually independent.

AR(3): Autoregressive model with order 3 ($p = 3$)

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \phi_3 Y_{t-3} + z_t, \qquad z_t \overset{iid}{\sim} N(0, \sigma^2), \tag{13.60}$$

$$\begin{bmatrix} Y_t \\ Y_{t-1} \\ Y_{t-2} \end{bmatrix} = \begin{bmatrix} \phi_1 & \phi_2 & \phi_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} Y_{t-1} \\ Y_{t-2} \\ Y_{t-3} \end{bmatrix} + \begin{bmatrix} z_t \\ 0 \\ 0 \end{bmatrix},$$

$$X_t = \begin{bmatrix} \phi_1 & \phi_2 & \phi_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} X_{t-1} + \begin{bmatrix} z_t \\ 0 \\ 0 \end{bmatrix}.$$

AR(p): Autoregressive model with order p

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + z_t, \qquad z_t \overset{iid}{\sim} N(0, \sigma^2), \tag{13.61}$$

$$\underbrace{\begin{bmatrix} Y_t \\ Y_{t-1} \\ \vdots \\ Y_{t-(p-1)} \end{bmatrix}}_{X_t} = \begin{bmatrix} \phi_1 & \phi_2 & \cdots & & \phi_p \\ 1 & 0 & \cdots & & 0 \\ 0 & 1 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & & \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix} \underbrace{\begin{bmatrix} Y_{t-1} \\ Y_{t-2} \\ \vdots \\ Y_{t-p} \end{bmatrix}}_{X_{t-1}} + \begin{bmatrix} z_t \\ 0 \\ 0 \end{bmatrix},$$

$$Y_t = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}_{1 \times p} X_t.$$

13.6 Appendix 13.D: Fourier Analysis

13.6.1 Fourier Series

A Fourier series represents a periodic function $f(x)$ with period P as a linear combination of sinusoidal waves. The sine-cosine form is expressed in Eq. (13.62) and the exponential form is expressed

in Eq. (13.63).

$$
\begin{aligned}
f(x) &= \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \cos \frac{2\pi n x}{P} + b_n \sin \frac{2\pi n x}{P} \right) \\
a_0 &= \frac{2}{P} \int_{-\frac{P}{2}}^{\frac{P}{2}} f(t) dt, \\
a_n &= \frac{2}{P} \int_{-\frac{P}{2}}^{\frac{P}{2}} f(t) \cos \frac{2\pi n t}{P} dt, \text{ and} \\
b_n &= \frac{2}{P} \int_{-\frac{P}{2}}^{\frac{P}{2}} f(t) \sin \frac{2\pi n t}{P} dt.
\end{aligned}
\tag{13.62}
$$

$$
\begin{aligned}
f(x) &= \sum_{n=-\infty}^{n=\infty} c_n e^{i \frac{2\pi n}{P} x}, \\
c_n &= \frac{1}{P} \int_{-\frac{P}{2}}^{\frac{P}{2}} f(t) e^{-i \frac{2\pi n}{P} t} dt.
\end{aligned}
\tag{13.63}
$$

Note that the harmonics are indexed by an integer, n, which is also the number of cycles the corresponding sinusoids make in the interval P.

13.6.2 Sine-Cosine Fourier Series to Exponential Series

The exponential form, Eq. (13.63), is shown to be algebraically equivalent to the sine-cosine form, Eq. (13.62), in the following derivation.

$$
\begin{aligned}
f(x) = &\frac{1}{P} \int_{-\frac{P}{2}}^{\frac{P}{2}} f(t) dt \\
&+ \sum_{n=1}^{\infty} \frac{2}{P} \left[\int_{-\frac{P}{2}}^{\frac{P}{2}} f(t) \cos \frac{2\pi n t}{P} dt \ \cos \frac{2\pi n x}{P} + \int_{-\frac{P}{2}}^{\frac{P}{2}} f(t) \sin \frac{2\pi n t}{P} dt \ \sin \frac{2\pi n x}{P} \right].
\end{aligned}
$$

$$
\text{Noting that } \cos u = \frac{1}{2} \left(\cos u + i \sin u + \cos u - i \sin u \right) = \frac{1}{2} \left(e^{iu} + e^{-iu} \right) \text{ and}
$$
$$
\sin u = \frac{1}{2i} \left(\cos u + i \sin u - \cos u + i \sin u \right) = \frac{1}{2i} \left(e^{iu} - e^{-iu} \right),
$$

$$
\begin{aligned}
f(x) = \frac{1}{P}\int_{-\frac{P}{2}}^{\frac{P}{2}} f(t)dt \\
&+ \sum_{n=1}^{\infty}\frac{2}{P}\left[\int_{-\frac{P}{2}}^{\frac{P}{2}} f(t)\frac{1}{2}\left(e^{i\frac{2\pi nt}{P}} + e^{-i\frac{2\pi nt}{P}}\right)dt\ \frac{1}{2}\left(e^{i\frac{2\pi nx}{P}} + e^{-i\frac{2\pi nx}{P}}\right)\right. \\
&\qquad\left. + \int_{-\frac{P}{2}}^{\frac{P}{2}} f(t)\frac{1}{2i}\left(e^{i\frac{2\pi nt}{P}} - e^{-i\frac{2\pi nt}{P}}\right)dt\ \frac{1}{2i}\left(e^{i\frac{2\pi nx}{P}} - e^{-i\frac{2\pi nx}{P}}\right)\right] \\
&= \frac{1}{P}\int_{-\frac{P}{2}}^{\frac{P}{2}} f(t)dt \\
&+ \sum_{n=1}^{\infty}\frac{2}{P}\left[\frac{1}{4}\int_{-\frac{P}{2}}^{\frac{P}{2}} f(t)\left(\cancel{e^{i\frac{2\pi n}{P}(x+t)}} + e^{i\frac{2\pi n}{P}(x-t)} + e^{i\frac{2\pi n}{P}(-x+t)} + \cancel{e^{i\frac{2\pi n}{P}(-x-t)}}\right.\right. \\
&\qquad\left.\left. -\cancel{e^{i\frac{2\pi n}{P}(x+t)}} + e^{i\frac{2\pi n}{P}(x-t)} + e^{i\frac{2\pi n}{P}(-x+t)} - \cancel{e^{i\frac{2\pi n}{P}(-x-t)}}\right)dt\right] \\
&= \frac{1}{P}\int_{-\frac{P}{2}}^{\frac{P}{2}} f(t)dt + \sum_{n=1}^{\infty}\frac{1}{2P}\int_{-\frac{P}{2}}^{\frac{P}{2}} f(t)\left(2e^{i\frac{2\pi n}{P}(x-t)} + 2e^{i\frac{2\pi n}{P}(-x+t)}\right)dt \\
&= \frac{1}{P}\int_{-\frac{P}{2}}^{\frac{P}{2}} f(t)e^{-i\frac{2\pi 0}{P}t}dt\ e^{i\frac{2\pi 0}{P}x} \\
&+ \sum_{n=1}^{\infty}\frac{1}{P}\left[\int_{-\frac{P}{2}}^{\frac{P}{2}} f(t)e^{-i\frac{2\pi n}{P}t}dt\ e^{i\frac{2\pi n}{P}x} + \int_{-\frac{P}{2}}^{\frac{P}{2}} f(t)e^{-i\frac{2\pi(-n)}{P}t}dt\ e^{i\frac{2\pi(-n)}{P}x}\right] \\
&= \sum_{n=-\infty}^{n=\infty}\frac{1}{P}\int_{-\frac{P}{2}}^{\frac{P}{2}} f(t)e^{-i\frac{2\pi n}{P}t}dt\ e^{i\frac{2\pi n}{P}x}\ .
\end{aligned}
$$

This reduces to the exponential form, Eq. (13.63).

13.6.3 Exponential Series to Continuous Fourier Transform

The continuous Fourier transform, which can be applied to certain non-periodic functions, can be derived by taking the limit as $P \to \infty$ in Eq. (13.63), using a change of variable $\omega = 2\pi n/P$ and

applying the Riemann integral definition as follows:

$$\begin{aligned} f(x) &= \lim_{P\to\infty} \sum_{n=-\infty}^{n=\infty} \left(\frac{1}{P} \int_{-\frac{P}{2}}^{\frac{P}{2}} f(t) e^{i\frac{2\pi n}{P}(x-t)} dt \right) \\ &= \int_{\omega=-\infty}^{\omega=\infty} \left(\int_{-\infty}^{\infty} f(t) e^{i\omega(x-t)} dt \right) \frac{d\omega}{2\pi} \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{i\omega x} \left(\int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \right) d\omega. \end{aligned} \tag{13.64}$$

The Fourier transform $\mathcal{F}(f)$ is expressed as follows:

$$\begin{aligned} \mathcal{F}(f) &= \phi(\omega), \\ \phi(\omega) &= \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt, \end{aligned} \tag{13.65}$$

which exists for all functions f that are *piecewise continuous* and *absolutely integrable*. The inverse Fourier transform $\mathcal{F}^{-1}(\phi)$ is accordingly expressed as follows:

$$\begin{aligned} \mathcal{F}^{-1}(\phi) &= f(t), \\ f(t) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \phi(\omega) e^{i\omega x} d\omega. \end{aligned} \tag{13.66}$$

The definitions in Eqs. (13.65) and (13.66) make up the *Fourier integral theorem*, also known as the *Fourier inversion theorem*.

13.6.4 Discrete Fourier Transform (DFT)

The discrete Fourier transform, DFT, is an approximation of the continuous Fourier transform, Eq. (13.65). Instead of expressing the Fourier spectrum as a function over the continuous domain of the frequency, ω, the DFT expresses the spectrum as a function over a discrete domain of a countable number of frequencies, $2\pi n/N$, where n is an integer between 1 and N, and N is the number of time-steps in the signal. Begin by discretizing with time-step Δt over the portion $t \in [0, N\Delta t)$ of the domain of the integral in Eq. (13.65) as follows,

$$\phi(n) = \sum_{k=0}^{N-1} f(k) e^{-i\frac{2\pi n}{N}k}, \tag{13.67}$$

where k indicates the index of the discrete time-step. With $\omega_N = e^{-i\frac{2\pi}{N}}$, Eq. (13.67) is restated as $\phi(\omega_N^n) = \sum_{k=0}^{N-1} f(k)\omega_N^{nk}$, which yields the following matrix equation:

$$\begin{pmatrix} \phi\left(\omega_N^0\right) \\ \phi\left(\omega_N^1\right) \\ \phi\left(\omega_N^2\right) \\ \vdots \\ \phi\left(\omega_N^{N-1}\right) \end{pmatrix} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{N-1} \end{bmatrix} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_N & \omega_N^2 & \cdots & \omega_N^{N-1} \\ 1 & \omega_N^2 & \omega_N^4 & \cdots & \omega_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_N^{N-1} & \omega_N^{2(N-1)} & \cdots & \omega_N^{(N-1)^2} \end{pmatrix} \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{N-1} \end{bmatrix}, \tag{13.68}$$

where $c_n = \phi\left(\omega_N^n\right)$ and $y_k = f(k)$.

13.6.5 Fast Fourier Transform (FFT)

The bare-bones (radix-2) Cooley-Tukey algorithm [20] is an implementation of the Fast Fourier Transform (FFT) that requires the signal length to be a power of 2. The algorithm takes advantage of recursive permutation matrix multiplication to perform the DFT much more quickly. Performing the DFT without recursion has a computational complexity of $O\left(n^2\right)$, while performing the DFT with recursion, i.e., the FFT, has a computational complexity of $O\left(n \log_2 n\right)$. For the definition of "computational complexity," refer to Section 5.2.2. The algorithm is performed as follows:

Base case: $N = 1$

$$c_0 = F_1 y_0 = \begin{bmatrix} 1 \end{bmatrix} y_0. \tag{13.69}$$

The next case: $N = 2$

$$\begin{aligned} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} &= F_2 \begin{bmatrix} y_0 \\ y_1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & \omega_2 \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \end{bmatrix} \\ &= \begin{bmatrix} I_1 & D_1 \\ I_1 & -D_1 \end{bmatrix} \begin{bmatrix} F_1 & 0 \\ 0 & F_1 \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \end{bmatrix}. \end{aligned} \tag{13.70}$$

For $N = 4, 8, 16, 32, \cdots$

$$\begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{N-1} \end{bmatrix} = F_N \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{N-1} \end{bmatrix} = \begin{bmatrix} I_{N/2} & D_{N/2} \\ I_{N/2} & -D_{N/2} \end{bmatrix} \begin{bmatrix} F_{N/2} & 0 \\ 0 & F_{N/2} \end{bmatrix} \begin{bmatrix} y_{\text{even}} \\ y_{\text{odd}} \end{bmatrix}. \tag{13.71}$$

13.6.6 Dirac Delta

Dirac's delta, $\delta(t)$, is a *generalized function* that has the following properties:

$$\int_{-\infty}^{\infty} \delta(t) dt = 1, \tag{13.72}$$

$$\int_{-\infty}^{\infty} f(t)\delta(t - t_o) dt = f(t_o). \tag{13.73}$$

A common interpretation of Dirac's delta is an *instantaneous pulse*. It can be thought of as a sampler of all frequencies when used as an input. Seeing that it is equivalent to the following Fourier coefficient-like integral:

$$\delta(t - t_o) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-ix(t-t_o)} dx, \tag{13.74}$$

the Fourier integral theorem confirms its identity, as follows:

$$\begin{aligned} f(t_o) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{ixt_o}\left\{\int_{-\infty}^{\infty} e^{-ixt} f(t)dt\right\}dx \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-ix(t-t_o)} f(t)dxdt \\ &= \int_{-\infty}^{\infty}\left\{\frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-ix(t-t_o)}dx\right\} f(t)dt \\ &= \int_{-\infty}^{\infty}\delta(t-t_o)f(t)dt. \end{aligned} \tag{13.75}$$

Chapter 14

quakeio

The quakeio library implements parsers for a variety of ground motion file formats, as well as various ground motion processing functions. The library is used by the *BRACE*^{2} platform to process event records. The packaged library is hosted on PyPI.

Ground motion data is represented by compositions of the following data types/containers:

1. QuakeSeries is an array-like data type which contains a single time series and associated metadata, e.g., peak values and units. All data contained by this type is generally closely related to a single physical quantity or *measurement*. An example of a file format that parses to this type is the PEER NGA *.AT2 file.
2. QuakeComponent is a collection of QuakeSeries types which generally represent time series data (e.g., acceleration, velocity, and displacement) which are collected in a single *direction*. An example of a file format that parses into this type is the CSMIP Volume 2 (*.V2) specification.
3. QuakeMotion is a collection of QuakeComponent types which all pertain to a single shared *spatial location*. The data contained by this type is generally free of any spatial variation.
4. QuakeEvent is a collection of QuakeMotion types, often corresponding to a single *site* and *event*. An example of a file format that parses into this type is the CSMIP processed archive (*.zip).

The core functionality of the quakeio library is exposed by the specifically-designed function quakeio.read(filename, format=None). This function returns one of the four objects listed above, depending on the format of the parsed file. For example, the return value of read when parsing a PEER NGA file (file extension .AT2) is a QuakeSeries with acceleration data. On the other hand, when parsing a CSMIP Volume 2 file, the return is a QuakeComponent containing QuakeSeries instances for acceleration, velocity, and displacement values. Finally, when parsing a zip archive of such files, a QuakeEvent is returned. Table 18 summarizes the file formats which are currently supported by the quakeio library.

Table 18: File formats supported by `quakeio` library.

Format	Read	Write	Reference	Data Type[a]
`quakeio.json`	✓	✓	Section 14.1	*Any*
`csmip.v2`	✓	-	[78]	C/S
`csmip.zip`	✓	-	[78]	E/M/C/S
`smc.zip`	✓	-	[82]	E/M/C/S
`eqsig`	✓	✓	[58]	S
`PEER NGA`	✓	-	[3]	S
`PEER SGMD`[b]	*under development*		[69]	S

[a] Each file type naturally pertains to different data types in the quakeio hierarchy. These are identified with the keys:

E `QuakeEvent`

C `QuakeComponent`

M `QuakeMotion`

S `QuakeSeries`

[b] Simulated Ground Motion Database

14.1 Schema for JSON Format

The remainder of this chapter documents the `quakeio` JSON file format which represents a `QuakeEvent` object.

14.1.1 `QuakeEvent` Schema

A `QuakeEvent` is represented by a JSON `object` with properties listed in Table 19.

Table 19: Fields of the `QuakeEvent` JSON `object` representation.

Property	Type	Required	Nullable	Defined by
motions	`array`	Required	cannot be null	QuakeCollection
Additional Properties	Any	Optional	can be null	

14.1.1.1 motions

The field `motions` is an array of `QuakeMotion` objects, where each typically corresponds to a single spatial location in the station. The value in this field must be of type `object[]` (QuakeMotion), as defined by QuakeCollection.

14.1.2 `QuakeMotion` Schema

A `QuakeMotion` is represented by a JSON `object` with properties listed in Table 23.

Table 20: Fields of the `QuakeMotion` JSON `object` representation.

Property	Type	Required	Nullable	Defined by
components	`array`	Required	cannot be null	QuakeCollection
Additional Properties	Any	Optional	can be null	

14.1.3 `QuakeComponent` Schema

A `QuakeComponent` is represented by a JSON `object` with properties listed in Table 21.

Table 21: Fields of the `QuakeComponent` JSON `object` representation.

Property	Type	Required	Nullable	Defined by
accel	`object`	Optional	cannot be null	QuakeComponent
veloc	`object`	Optional	cannot be null	QuakeComponent
displ	`object`	Optional	cannot be null	QuakeComponent
ihdr	`array`	Optional	cannot be null	QuakeComponent
rhdr	`array`	Optional	cannot be null	QuakeComponent
location	`string`	Optional	cannot be null	QuakeComponent
station_no	`string`	Optional	cannot be null	QuakeComponent
azimuth	`string`	Optional	cannot be null	QuakeComponent
instr_period	`number`	Optional	cannot be null	QuakeComponent
instr_period.units	`string`	Optional	cannot be null	QuakeComponent
peak_accel	`number`	Required	cannot be null	QuakeComponent
peak_accel.units	`string`	Required	cannot be null	QuakeComponent
accel.time_step	`number`	Optional	cannot be null	QuakeComponent
peak_accel.time	`number`	Required	cannot be null	QuakeComponent
peak_veloc	`number`	Required	cannot be null	QuakeComponent
peak_veloc.units	`string`	Required	cannot be null	QuakeComponent
peak_veloc.time	`number`	Required	cannot be null	QuakeComponent
peak_displ	`number`	Required	cannot be null	QuakeComponent
peak_displ.units	`string`	Required	cannot be null	QuakeComponent
peak_displ.time	`number`	Required	cannot be null	QuakeComponent
init_displ	`number`	Optional	cannot be null	QuakeComponent
init_displ.units	`string`	Optional	cannot be null	QuakeComponent
init_veloc	`number`	Optional	cannot be null	QuakeComponent
init_veloc.units	`string`	Optional	cannot be null	QuakeComponent
file_name	`string`	Optional	cannot be null	QuakeComponent
Additional Properties	Any	Optional	can be null	

14.1.3.1 accel

The `accel` field stores acceleration data. The value in this field must be of type `object` (QuakeSeries), as defined by QuakeComponent.

14.1.3.2 veloc

The `veloc` field stores velocity data. The value in this field must be of type `object` (QuakeSeries), as defined by (QuakeSeries).

14.1.3.3 displ

The `displ` field stores displacement data. The value in this field must be of type `object` (QuakeSeries), as defined by QuakeComponent.

14.1.3.4 ihdr

The `ihdr` field stores an integer header block, like that provided by [78, 82]. The value in this field must be of type `integer[]`.

14.1.3.5 rhdr

The `rhdr` field stores a floating point header block, like that provided by [78, 82]. The value in this field must be of type `number[]`.

14.1.3.6 location

The `location` field stores a number identifying the station at which the data was recorded. The value in this field must be of type `string` (Location), as defined by QuakeComponent.

14.1.3.7 station_no

The `station_no` field stores a number identifying the station at which the data was recorded. The value in this field must be of type `string` (Station number), as defined by QuakeComponent.

14.1.3.8 azimuth

The `azimuth` field stores an azimuth identifying the direction of ground motion component data. The value in this field must be of type `string` (Azimuth), as defined by QuakeComponent. For example:

```
"37.691N, 122.099W"
```

14.1.3.9 peak_accel

The `peak_accel` field stores the peak value corresponding to the `QuakeSeries` referenced by the accel field. The value in this field must be of type `number` (Peak acceleration), as defined by QuakeComponent. For example:

```
17.433
```

14.1.3.10 peak_accel.units

The `peak_accel.units` field stores a string indicating the units of the `peak_accel` field. The value in this field must be of type `string` (Peak acceleration units), as defined by QuakeComponent. For example:

```
"cm/sec/sec"
```

14.1.3.11 accel.time_step

The `accel.time_step` field stores the time step of the acceleration data (`QuakeComponent.accel`). The value in this field must be of type `number` (Time step of acceleration data), as defined by QuakeComponent.

14.1.3.12 peak_accel.time

The `peak_accel.time` field stores the time at which peak acceleration is attained. The value in this field must be of type `number` (Time of peak acceleration), as defined by QuakeComponent. For example:

```
20.27
```

14.1.3.13 peak_veloc

The `peak_veloc` field stores the peak value corresponding to the `QuakeSeries` referenced by the veloc field. The value in this field must be of type `number` (Peak velocity), as defined by QuakeComponent. For example:

```
0.205
```

14.1.3.14 peak_veloc.units

The `peak_veloc.units` field stores the units of the `peak_veloc` field. The value in this field must be of type `string` (peak_veloc.units), as defined by QuakeComponent. For example:

```
"cm/sec"
```

14.1.3.15 peak_veloc.time

The `peak_veloc.time` field stores the time at which peak velocity is attained. The value in this field must be of type `number` (Time of peak velocity), as defined by QuakeComponent.

14.1.3.16 peak_displ

The `peak_displ` field stores the peak value of the series referenced by the `displ` field. The value in this field must be of type `number` (Peak displacement), as defined by QuakeComponent.

14.1.3.17 peak_displ.units

The `peak_displ.units` field stores the units of the `peak_displ` field. The value in this field must be of type `string` (peak_displ.units), as defined by QuakeComponent. For example:

```
"cm"
```

14.1.3.18 peak_displ.time

The `peak_displ.time` field stores the time at which peak displacement is attained. The value in this field must be of type `number` (Time of peak displacement), as defined by QuakeComponent. For example:

```
    20.27
```

14.1.3.19 file_name

The `file_name` field stores the name of the file from which the record was parsed. The value in this field must be of type `string` (Source file name), as defined by QuakeComponent.

14.1.4 QuakeFilter Schema

A `QuakeFilter` is represented by a JSON `object` with properties listed in Table 22.

Table 22: Fields of the `QuakeFilter` JSON `object` representation.

Property	Type	Required	Nullable	Defined by
filter_type	`string`	Required	cannot be null	QuakeFilter
point	`number`	Optional	cannot be null	QuakeFilter
point.units	`string`	Optional	cannot be null	QuakeFilter
limit_low	`number`	Optional	cannot be null	QuakeFilter
limit_high	`number`	Optional	cannot be null	QuakeFilter
limit.units	`string`	Optional	cannot be null	QuakeFilter
Additional Properties	Any	Optional	can be null	

14.1.4.1 filter_type

The field `filter_type` stores a string indicating the type of the `QuakeFilter` object. The value in this field must be of type `string`, as defined by QuakeFilter.

14.1.4.2 point

The value in the `point` field must be of type `number` (The point schema), as defined by QuakeFilter.

14.1.4.3 point.units

The `point.units` field must be of type `string`, as defined by QuakeFilter.

14.1.4.4 limit_low

The `limit_low` field must be of type `number`, as defined by QuakeFilter.

14.1.4.5 limit_high

The `limit_high` field must be of type `number`, as defined by QuakeFilter.

14.1.4.6 limit.units

The `limit.units` field must be of type `string` (The limit.units schema).

14.1.5 QuakeMotion Schema

A `QuakeMotion` is represented by a JSON `object` with properties listed in Table 23.

Table 23: Fields of the `QuakeMotion` JSON `object` representation.

Property	Type	Required	Nullable	Defined by
key	`string`	Required	cannot be null	QuakeCollection
components	`array`	Required	cannot be null	QuakeCollection
Additional Properties	Any	Optional	can be null	

14.1.5.1 key

The `key` field stores an string that identifies the location corresponding to the component. The value in this field must be of type `string`, as defined by QuakeCollection.

14.1.5.2 components

The `components` field stores the collection of components which define the three-dimensional motion of the `QuakeMotion` object. The value in this field must be of type `object[]` (QuakeComponent), as defined by QuakeCollection.

14.1.6 QuakeSeries Schema

A `QuakeSeries` is represented by a JSON `object` with properties listed in Table 24.

Table 24: Fields of the `QuakeSeries` JSON `object` representation.

Property	Type	Required	Nullable	Defined by
peak_value	`number`	Optional	cannot be null	QuakeSeries
units	`string`	Required	cannot be null	QuakeSeries
peak_time	`number`	Optional	cannot be null	QuakeSeries
shape	`integer`	Required	cannot be null	QuakeSeries
time_step	`number`	Required	cannot be null	QuakeSeries
data	`array`	Optional	cannot be null	QuakeSeries
Additional Properties	Any	Optional	can be null	

14.1.6.1 peak_value

The `peak_value` field stores the element of the series data (field `data`) with maximum absolute value. The value in this field must be of type `number`, as defined by QuakeSeries.

14.1.6.2 units

The `units` field stores stores a label indicating the units of the series data. The value in this field must be of type `string` (Series units), as defined by QuakeSeries.

14.1.6.3 peak_time

The `peak_time` field stores the time at which the value in `peak_value` is attained. The value in this field must be of type `number`(Time of peak value), as defined by QuakeSeries.

14.1.6.4 shape

The `shape` field stores the shape of the series data. The value in this field must be of type `integer` (Series shape), as defined by QuakeSeries.

14.1.6.5 time_step

The `time_step` field stores the time step between samples in the `data` field. The value in this field must be of type `number` (Time step), as defined by QuakeSeries.

14.1.6.6 data

The `data` field is an `array` of `numbers` defined by QuakeSeries.

Bibliography

[1] N. A. Abrahamson. "State Of The Practice Of Seismic Hazard Evaluation". en. In: *ISRM International Symposium Proceedings*. OnePetro, Nov. 2000. URL: https://onepetro.org/ISRMIS/proceedings-abstract/IS00/All-IS00/50863 (visited on 01/20/2023).

[2] A. Almutairi et al. "Analysis of Multi-span Bridges Using Opensees". en. In: (Oct. 2016).

[3] T. D. Ancheta et al. *PEER NGA-West2 Database*. 2013. URL: https://peer.berkeley.edu/publications/2013-03.

[4] Y. Arici and K. M. Mosalam. "Modal identification of bridge systems using state-space methods". en. In: *Structural Control and Health Monitoring* 12.3-4 (2005). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/stc.76, pp. 381–404. ISSN: 1545-2263. DOI: 10.1002/stc.76.

[5] Y. Arici and K. M. Mosalam. "System Identification of Instrumented Bridge Systems". In: *Earthquake Engineering & Structural Dynamics* 32.7 (2003), pp. 999–1020. ISSN: 1096-9845. DOI: 10/bgr7h6.

[6] M. A. Aschheim. *Misssion Gothic Bridge*. 1994. URL: https://nisee.berkeley.edu/elibrary/Image/NR917.

[7] T. T. Baber and Y.-K. Wen. "Random Vibration of Hysteretic, Degrading Systems". In: *Journal of the Engineering Mechanics Division* 107.6 (Dec. 1981), pp. 1069–1087. ISSN: 0044-7951, 2690-2427. DOI: 10.1061/JMCEA3.0002768.

[8] J. L. Beck. *Determining Models of Structures from Earthquake Records*. Report or Paper. California Institute of Technology, Jan. 1, 1978. URL: https://resolver.caltech.edu/CaltechEERL:1978.EERL-78-01 (visited on 07/17/2021).

[9] R. Betti and A. L. Hong. "Identification of the baseline modal parameters of the Carquinez suspension bridge using ambient vibration data". In: *SMIP08 seminar on utilization of strong-motion data*. 2008, pp. 63–82.

[10] R. Bouc. "Forced Vibrations of Mechanical Systems with Hysteresis". In: *Proc. of the Fourth Conference on Nonlinear Oscillations, Prague, 1967*. 1967. (Visited on 05/28/2022).

[11] B. M. Broderick and A. S. Elnashai. "Analysis of the failure of interstate 10 freeway ramp during the Northridge earthquake of 17 January 1994". en. In: *Earthquake Engineering & Structural Dynamics* 24.2 (1995). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/eqe.4290240205, pp. 189–208. ISSN: 1096-9845. DOI: 10.1002/eqe.4290240205.

[12] California Department of Transportation. *Seismic Design Criteria Version 2.0*. 2019, p. 250.

[13] Caltrans. *COLUMN ANALYSIS CONSIDERATIONS*. 6-1. Feb. 2009.

[14] Caltrans. *Memo to Designers 5-2: Diaphragm Abutments*. Letter. Oct. 1992. URL: https://dot.ca.gov/-/media/dot-media/programs/engineering/documents/memotodesigner/5-2-a11y.pdf (visited on 07/15/2021).

[15] S. D. C. Caltrans. "Seismic design criteria version 2.0". In: *Sacramento, CA: State of California, Department of Transportation* (2019).

[16] A. Cardona and M. Geradin. "A Beam Finite Element Non-Linear Theory with Finite Rotations". In: *International Journal for Numerical Methods in Engineering* 26.11 (Nov. 1988), pp. 2403–2438. ISSN: 0029-5981, 1097-0207. DOI: 10.1002/nme.1620261105.

[17] M. Çelebi, S. F. Ghahari, and E. Taciroglu. "Responses of the odd couple Carquinez, CA, suspension bridge during the Mw6.0 south Napa earthquake of August 24, 2014". en. In: *Journal of Civil Structural Health Monitoring* 9.5 (Nov. 2019), pp. 719–739. ISSN: 2190-5479. DOI: 10.1007/s13349-019-00363-6.

[18] N. Chopin and O. Papaspiliopoulos. *An Introduction to Sequential Monte Carlo*. en. Springer Series in Statistics. Cham: Springer International Publishing, 2020. ISBN: 978-3-030-47844-5 978-3-030-47845-2. DOI: 10.1007/978-3-030-47845-2.

[19] A. K. Chopra. *Dynamics of structures: theory and applications to earthquake engineering*. en. 4th ed. Upper Saddle River, N.J: Prentice Hall, 2012. ISBN: 978-0-13-285803-8.

[20] J. W. Cooley and J. W. Tukey. "An algorithm for the machine calculation of complex Fourier series". en. In: *Mathematics of Computation* 19.90 (1965), pp. 297–301. ISSN: 0025-5718, 1088-6842. DOI: 10.1090/S0025-5718-1965-0178586-1.

[21] R. Desroches and G. L. Fenves. "Evaluation of Recorded Earthquake Response of a Curved Highway Bridge". en. In: *Earthquake Spectra* 13.3 (Aug. 1997). Publisher: SAGE Publications Ltd STM, pp. 363–386. ISSN: 8755-2930. DOI: 10.1193/1.1585953.

[22] F. C. Filippou, E. P. Popov, and V. V. Bertero. *Effects of Bond Deterioration on Hysteretic Behavior of Reinforced Concrete Joints*. Tech. rep. EERC-83-19. Berkeley, CA: Earthquake Engineering Research Center, 1983.

[23] M. Fraino et al. "Seismic Soil-Structure Interaction Effects in Instrumented Bridges". en. In: (2012). URL: https://www.iitk.ac.in/nicee/wcee/article/WCEE2012_5307.pdf.

[24] M. Fraino et al. "Evaluation of soil-structure interaction effects on seismic response using recorded motions from instrumented bridges". In: Jan. 2014.

[25] A. Giuffre and P. E. Pinto. "Il Comportamento Del Cemento Armato per Sollecitazioni Cicliche Di Forte Intensità". In: (1970).

[26] R. K. Goel and A. K. Chopra. "Evaluation of Bridge Abutment Capacity and Stiffness during Earthquakes". In: *Earthquake Spectra* 13.1 (Feb. 1997), pp. 1–23. ISSN: 8755-2930, 1944-8201. DOI: 10/fdkchm.

[27] J. E. Goldberg and R. M. Richard. "Analysis of Nonlinear Structures". In: *Journal of the Structural Division* 89.4 (Apr. 1963), pp. 333–351. ISSN: 0044-8001, 2690-3377. DOI: 10.1061/JSDEAG.0000948.

[28] X. He et al. “System Identification of Alfred Zampa Memorial Bridge Using Dynamic Field Test Data”. EN. In: *Journal of Structural Engineering* 135.1 (Jan. 2009). Publisher: American Society of Civil Engineers, pp. 54–66. ISSN: 0733-9445. DOI: 10.1061/(ASCE)0733-9445(2009)135:1(54).

[29] B. L. Ho and R. E. Kalman. “Editorial: Effective construction of linear state-variable models from input/output functions: Die Konstruktion von linearen Modeilen in der Darstellung durch Zustandsvariable aus den Beziehungen für Ein- und AusgangsgröSSen”. de. In: *Automatisierungstechnik* 14.1-12 (Dec. 1966). Publisher: De Gruyter (O), pp. 545–548. ISSN: 2196-677X. DOI: 10.1524/auto.1966.14.112.545.

[30] E. Hognestad. “Study of Combined Bending and Axial Load in Reinforced Concrete Members”. PhD thesis. June 1951.

[31] M. J. Huang and A. F. Shakal. “CSMIP Strong-Motion Instrumentation and Records from the I10/215 Interchange Bridge near San Bernardino”. In: *Earthquake Spectra* 11.2 (May 1, 1995), pp. 193–215. ISSN: 8755-2930. DOI: 10/bc64bd.

[32] T. J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Mineola, NY: Dover Publications, 1987. ISBN: 978-0-486-41181-1.

[33] T. J. Ingham, S. Rodriguez, and M. Nader. “Nonlinear analysis of the Vincent Thomas Bridge for seismic retrofit”. en. In: *Computers & Structures*. Proceedings of the 11th ADINA Conference 64.5 (Sept. 1997), pp. 1221–1238. ISSN: 0045-7949. DOI: 10.1016/S0045-7949(97)00031-X.

[34] N. P. Jones and R. H. Scanlan. “Theory and Full-Bridge Modeling of Wind Response of Cable-Supported Bridges”. EN. In: *Journal of Bridge Engineering* 6.6 (Dec. 2001). Publisher: American Society of Civil Engineers, pp. 365–375. ISSN: 1084-0702. DOI: 10.1061/(ASCE)1084-0702(2001)6:6(365).

[35] J.-N. Juang. “System Realization Using Information Matrix”. en. In: *Journal of Guidance, Control, and Dynamics* 20.3 (May 1997), pp. 492–500. ISSN: 0731-5090, 1533-3884. DOI: 10.2514/2.4068.

[36] J.-N. Juang, J. E. Cooper, and J. R. Wright. “An eigensystem realization algorithm using data correlations (ERA/DC) for modal parameter identification”. In: NTRS Author Affiliations: NASA Langley Research Center, Queen Mary College NTRS Document ID: 19870035963 NTRS Research Center: Legacy CDMS (CDMS). Apr. 1987. URL: https://ntrs.nasa.gov/citations/19870035963 (visited on 12/28/2023).

[37] J.-N. Juang and R. S. Pappa. “An eigensystem realization algorithm for modal parameter identification and model reduction”. en. In: *Journal of Guidance, Control, and Dynamics* 8.5 (Sept. 1985), pp. 620–627. ISSN: 0731-5090, 1533-3884. DOI: 10.2514/3.20031.

[38] J.-N. Juang and M. Q. Phan. *Identification and control of mechanical systems*. en. Cambridge, England ; New York: Cambridge University Press, 2001. ISBN: 978-0-521-78355-2.

[39] J.-N. Juang et al. “Identification of observer/Kalman filter Markov parameters - Theory and experiments”. In: *Journal of Guidance, Control, and Dynamics* 16.2 (1993). Publisher: American Institute of Aeronautics and Astronautics _eprint: https://doi.org/10.2514/3.21006, pp. 320–329. ISSN: 0731-5090. DOI: 10.2514/3.21006.

[40] R. E. Kalman. “Mathematical Description of Linear Dynamical Systems”. In: *Journal of the Society for Industrial and Applied Mathematics Series A Control* 1.2 (Jan. 1963). Publisher: Society for Industrial and Applied Mathematics, pp. 152–192. ISSN: 0887-4603. DOI: 10.1137/0301010.

[41] D. Karmakar, S. Ray-Chaudhuri, and M. Shinozuka. “Finite element model development, validation and probabilistic seismic performance evaluation of Vincent Thomas suspension bridge”. In: *Structure and Infrastructure Engineering* 11.2 (Feb. 2015). Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/15732479.2013.863360, pp. 223–237. ISSN: 1573-2479. DOI: 10.1080/15732479.2013.863360.

[42] D. Karmakar, S. Ray-Chaudhuri, and M. Shinozuka. “Seismic response evaluation of retrofitted Vincent Thomas bridge under spatially variable ground motions”. en. In: *Soil Dynamics and Earthquake Engineering* 42 (Nov. 2012), pp. 119–127. ISSN: 0267-7261. DOI: 10.1016/j.soildyn.2012.06.008.

[43] P. Kaviani, F. Zareian, and E. Taciroglu. *Performance-Based Seismic Assessment of Skewed Bridges*. en. Tech. rep. Pacific Earthquake Engineering Research CenterP, Jan. 2014. URL: https://peer.berkeley.edu/sites/default/files/webpeer-2014-01-peyman_kaviani_farzin_zareian_and_ertugrul_taciroglu.pdf.

[44] D. C. Kent and R. Park. “Cyclic Load Behaviour of Reinforcing Steel”. In: *Strain* 9.3 (July 1973), pp. 98–103. ISSN: 00392103, 14751305. DOI: 10.1111/j.1475-1305.1973.tb01816.x.

[45] K. Kim et al. “Seismic Response of a Large-Scale Highway Interchange System”. en. In: *Earthquake Geotechnical Engineering Design*. Ed. by M. Maugeri and C. Soccodato. Geotechnical, Geological and Earthquake Engineering. Cham: Springer International Publishing, 2014, pp. 223–240. ISBN: 978-3-319-03182-8. DOI: 10.1007/978-3-319-03182-8_9.

[46] M. Kurata et al. “Internet-Enabled Wireless Structural Monitoring Systems: Development and Permanent Deployment at the New Carquinez Suspension Bridge”. EN. In: *Journal of Structural Engineering* 139.10 (Oct. 2013). Publisher: American Society of Civil Engineers, pp. 1688–1702. ISSN: 1943-541X. DOI: 10.1061/(ASCE)ST.1943-541X.0000609.

[47] M. B. Levine and R. F. Scott. “Dynamic Response Verification of Simplified BridgeFoundation Model”. In: *Journal of Geotechnical Engineering* 115.2 (Feb. 1989), pp. 246–260. ISSN: 0733-9410, 1944-8368. DOI: 10/dxh5kd.

[48] H. Lu, R. Betti, and R. W. Longman. “Identification of linear structural systems using earthquake-induced vibration data”. en. In: *Earthquake Engineering & Structural Dynamics* 28.11 (1999). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/%28SICI%291096-9845%28199911%2928%3A11%3C1449%3A%3AAID-EQE881%3E3.0.CO%3B2-5, pp. 1449–1467. ISSN: 1096-9845. DOI: 10.1002/(SICI)1096-9845(199911)28:11<1449::AID-EQE881>3.0.CO;2-5.

[49] J. Lysmer and R. L. Kuhlemeyer. “Finite dynamic model for infinite media”. In: *Journal of the engineering mechanics division* 95.4 (1969), pp. 859–877.

[50] J. B. Mander, M. J. N. Priestley, and R. Park. “Theoretical StressStrain Model for Confined Concrete”. en. In: *Journal of Structural Engineering* 114.8 (Aug. 1988). Publisher: American Society of Civil Engineers, pp. 1804–1826. DOI: 10.1061/(ASCE)0733-9445(1988)114:8(1804).

[51] B. Maroney, K. Romstad, and M. Chajes. “Interpretation of Rio Dell freeway response during six recorded earthquake events”. In: *Procs. of the 4th US National Conference on Earthquake Eng., Palm springs, CA*. Vol. 1. 1990, pp. 1007–1016.

[52] B. H. Maroney and Y. H. Chai. “Bridge abutment stiffness and strength under earthquake loadings”. In: *2nd International Workshop on the Seismic Design of Bridges*. 1994.

[53] P. Matinrad and F. Petrone. “ASCE/SEI 7-compliant site-specific evaluation of the seismic demand posed to reinforced concrete buildings with real and simulated ground motions”. In: *Earthquake Engineering & Structural Dynamics* 52.15 (2023), pp. 4987–5009.

[54] S. Mazzoni et al. “OpenSees command language manual”. In: *Pacific Earthquake Engineering Research (PEER) Center* 264.1 (2006). Publisher: Berkeley, California, United States, pp. 137–158.

[55] D. B. McCallen and K. M. Romstad. “Dynamic Analyses of a Skewed Short-Span, Box-Girder Overpass”. In: *Earthquake Spectra* 10.4 (Nov. 1994), pp. 729–755. ISSN: 8755-2930. DOI: 10.1193/1.1585795.

[56] F. McKenna, M. H. Scott, and G. L. Fenves. “Nonlinear Finite-Element Analysis Software Architecture Using Object Composition”. en. In: *Journal of Computing in Civil Engineering* 24.1 (Jan. 2010). Publisher: American Society of Civil Engineers, pp. 95–107. DOI: 10.1061/(ASCE)CP.1943-5487.0000002.

[57] S. H. Megally, P. F. Silva, and F. Seible. *SEISMIC RESPONSE OF SACRIFICIAL SHEAR KEYS IN BRIDGE ABUTMENTS*. en. Structural Systems Research Project SSRR-2001/23. California Department of Transportation, University of California, San Diego, May 2002.

[58] M. Millen. *eng-tools/eqsig: First production release*. Version v1.0.0. June 2019. DOI: 10.5281/zenodo.3263949.

[59] R. Mohammadi and R. Motamed. “Assessing Soil-Shaft Interaction at an Instrumented Bridge in California in Terms of Effects of Soil-Shaft Nonlinear Properties in Various Earthquake Scenarios”. Unpublished yet. 2024.

[60] R. Mohammadi and R. Motamed. “NUMERICAL MODELING OF SOIL-SHAFT INTERACTION VALIDATED USING THE INSTRUMENTED HAYWARD BRIDGE IN CALIFORNIA”. In: ().

[61] V. Mosquera, A. W. Smyth, and R. Betti. “UTILIZATION OF STRONG-MOTION DATA FOR ASSESSMENT OF STRUCTURAL INTEGRITY IN INSTRUMENTED HIGHWAY BRIDGES”. In: (2009), p. 20.

[62] V. Mosquera, A. W. Smyth, and R. Betti. “Rapid Evaluation and Damage Assessment of Instrumented Highway Bridges: DAMAGE ASSESSMENT OF INSTRUMENTED HIGHWAY BRIDGES”. In: *Earthquake Engineering & Structural Dynamics* 41.4 (Apr. 10, 2012), pp. 755–774. ISSN: 00988847. DOI: 10/bww9tz.

[63] V. Mosquera, A. W. Smyth, and R. Betti. “Utilization of strong-motion data for assessment of structural integrity in instrumented highway bridges”. In: *SMIP09* 65 (2009). Publisher: Citeseer.

[64] S. Muthukumar and R. DesRoches. “A Hertz contact model with non-linear damping for pounding simulation”. en. In: *Earthquake Engineering & Structural Dynamics* 35.7 (June 2006), pp. 811–828. ISSN: 0098-8847, 1096-9845. DOI: 10.1002/eqe.557.

[65] R. D. Nayeri et al. “Study of Time-Domain Techniques for Modal Parameter Identification of a Long Suspension Bridge with Dense Sensor Arrays”. EN. In: *Journal of Engineering Mechanics* 135.7 (July 2009). Publisher: American Society of Civil Engineers, pp. 669–683. ISSN: 0733-9399. DOI: 10.1061/(ASCE)0733-9399(2009)135:7(669).

[66] A. Neuenhofer and F. C. Filippou. “Evaluation of Nonlinear Frame Finite-Element Models”. In: *Journal of Structural Engineering* 123.7 (July 1997), pp. 958–966. ISSN: 0733-9445, 1943-541X. DOI: 10.1061/(asce)0733-9445(1997)123:7(958).

[67] A.-S. M. Niazy. “Seismic performance evaluation of suspension bridges”. PhD Thesis. University of Southern California, 1991.

[68] R. S. Pappa, K. B. Elliott, and A. Schenk. “Consistent-mode indicator for the eigensystem realization algorithm”. en. In: *Journal of Guidance, Control, and Dynamics* 16.5 (Sept. 1993), pp. 852–858. ISSN: 0731-5090, 1533-3884. DOI: 10.2514/3.21092.

[69] *PEER Simulated Ground Motion Database*. 2024. URL: https://sgmd.peer.berkeley.edu/.

[70] E. P. Popov and H. Petersson. “Cyclic Metal Plasticity: Experiments and Theory”. In: *Journal of the Engineering Mechanics Division* 104.6 (Dec. 1978), pp. 1371–1388. ISSN: 0044-7951, 2690-2427. DOI: 10.1061/JMCEA3.0002434.

[71] W. Ramberg and W. R. Osgood. *Description of Stress-Strain Curves by Three Parameters*. Tech. rep. Technical Note No. 902. National Advisory Committee for Aeronautics, July 1943.

[72] L. Reese and K. Nyman. “Field Load Tests of Instrumented Drilled Shafts at Islamorada, Florida”. In: *Report to the Girdler Foundation and Exploration Corporation, Clearwater, FL, February* (1978).

[73] I. Romero. “The Interpolation of Rotations and Its Application to Finite Element Models of Geometrically Exact Rods”. In: *Computational Mechanics* 34.2 (July 2004). ISSN: 0178-7675, 1432-0924. DOI: 10.1007/s00466-004-0559-z.

[74] E. Safak. “Use of structural response data from small earthquakes and aftershocks”. In: *NIST SPECIAL PUBLICATION SP* (1994). Publisher: NATIONAL INSTIUTE OF STANDARDS & TECHNOLOGY, pp. 613–613.

[75] A. Saini and M. Saiidi. *Probabilistic damage control approach for seismic design of bridge columns*. California Department of Transportation, 2014.

[76] A. V. Sanchez, F. Seible, and M. N. Priestley. *Seismic Performance of Flared Bridge Columns*. SSRP - 97/06. Oct. 1997.

[77] B. D. Scott, R. Park, and M. J. N. Priestley. “Stress-Strain Behavior of Concrete Confined by Overlapping Hoops at Low and High Strain Rates”. In: *ACI Journal Proceedings* 79.1 (Jan. 1982). DOI: 10.14359/10875.

[78] A. Shakal and M. Huang. *Standard Tape Format for CSMIP Strong-Motion Data Tapes*. Dec. 1985. URL: https://www.conservation.ca.gov/cgs/documents/program-smi/reports/other/OSMS_85-03.pdf.

[79] A. Shamsabadi and E. Taciroglu. “A Frequency-Time Domain Handshake Method For Seismic Soil-Foundation-Structure Interaction Analysis of Long-Span Bridges”. en. In: (2013).

[80] J. C. Simo and T. J. R. Hughes. *Computational Inelasticity*. Interdisciplinary Applied Mathematics v. 7. New York: Springer, 1998. ISBN: 978-0-387-97520-7.

[81] J. Simo, D. Fox, and M. Rifai. “On a Stress Resultant Geometrically Exact Shell Model. Part III: Computational Aspects of the Nonlinear Theory”. In: *Computer Methods in Applied Mechanics and Engineering* 79.1 (Mar. 1990), pp. 21–70. ISSN: 00457825. DOI: 10.1016/0045-7825(90)90094-3.

[82] *SMC- and BBF-format Data Files*. 2001. URL: https://escweb.wr.usgs.gov/nsmp-data/smcfmt.html.

[83] A. W. Smyth, J.-S. Pei, and S. F. Masri. “System identification of the Vincent Thomas suspension bridge using earthquake records”. en. In: *Earthquake Engineering & Structural Dynamics* 32.3 (2003). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/eqe.226, pp. 339–367. ISSN: 1096-9845. DOI: 10.1002/eqe.226.

[84] F. Soleimani, S. Mangalathu, and R. DesRoches. “Seismic Resilience of Concrete Bridges with Flared Columns”. In: *Procedia Engineering* 199 (2017), pp. 3065–3070. ISSN: 18777058. DOI: 10.1016/j.proeng.2017.09.417.

[85] E. Spacone, V. Ciampi, and F. Filippou. “Mixed formulation of nonlinear beam finite element”. In: *Computers & Structures* 58.1 (1996), pp. 71–83. ISSN: 0045-7949. DOI: https://doi.org/10.1016/0045-7949(95)00103-N.

[86] E. Spacone, V. Ciampi, and F. C. Filippou. *A Beam Element for Seismic Damage Analysis*. Tech. rep. EERC-92-07. Aug. 1992.

[87] D. Swensen. *CSMIP Pilot Project on Feasibility of Using Lower Cost Strong-Motion Instrumentation to Fill the Gaps*. Conference Presentation. UCLA Mong Learning Center, Engineering VI, 404 Westwood Plaza, Los Angeles, CA, Nov. 2022.

[88] E. Taciroglu et al. “Comparative Study of Model Predictions and Data from Caltrans/CSMIP Bridge Instrumentation Program: A Case study on the Eureka-Samoa Channel Bridge”. In: (Mar. 2014). Number: CA14-2418. URL: https://trid.trb.org/view/1370061 (visited on 07/01/2023).

[89] M. Turek et al. “Ambient Vibration Testing of the Painter St. Overpass”. en. In: *Dynamics of Civil Structures, Volume 4*. Ed. by F. N. Catbas. Conference Proceedings of the Society for Experimental Mechanics Series. Cham: Springer International Publishing, 2014, pp. 411–417. ISBN: 978-3-319-04546-7. DOI: 10.1007/978-3-319-04546-7_44.

[90] P. Van Overschee and B. De Moor. "N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems". en. In: *Automatica*. Special issue on statistical signal processing and control 30.1 (Jan. 1994), pp. 75–93. ISSN: 0005-1098. DOI: 10.1016/0005-1098(94)90230-5.

[91] A. Venkittaraman. "Seismic resilience of highway bridges". In: (2013).

[92] C. E. Ventura, R. Brincker, and P. Andersen. "Dynamic properties of the Painter Street Overpass at different levels of vibration". en. In: ().

[93] C. E. Ventura et al. "Ambient vibration studies of three short-span reinforced concrete bridges". In: *Proceedings of the Elenventh World Conference on Earthqauke Engineering*. 1996.

[94] N. Wang, A. Elgamal, and J. Lu. "Assessment of the Samoa Channel Bridge-foundation seismic response". en. In: *Soil Dynamics and Earthquake Engineering* 108 (May 2018), pp. 150–159. ISSN: 0267-7261. DOI: 10.1016/j.soildyn.2018.01.042.

[95] N. Wang, A. Elgamal, and J. Lu. "Seismic response of the Eureka Channel Bridge-Foundation system". en. In: *Soil Dynamics and Earthquake Engineering* 152 (Jan. 2022), p. 107015. ISSN: 0267-7261. DOI: 10.1016/j.soildyn.2021.107015.

[96] N. Wang, A. Elgamal, and T. Shantz. "Recorded seismic response of the Samoa Channel Bridge-foundation system and adjacent downhole array". en. In: *Soil Dynamics and Earthquake Engineering* 92 (Jan. 2017), pp. 358–376. ISSN: 0267-7261. DOI: 10.1016/j.soildyn.2016.09.034.

[97] Y.-K. Wen and M. Asce. "Method for Random Vibration of Hysteretic Systems". In: *RANDOM VIBRATION* (1976).

[98] S. D. Werner, J. L. Beck, and M. B. Levine. "Seismic Response Evaluation of Meloland Road Overpass Using 1979 Imperial Valley Earthquake Records". In: *Earthquake Engineering & Structural Dynamics* 15.2 (1987), pp. 249–274. ISSN: 00988847, 10969845. DOI: 10/b9jxpx.

[99] J. C. Wilson. "Stiffness of Non-Skew Monolithic Bridge Abutments for Seismic Analysis". In: *Earthquake Engineering & Structural Dynamics* 16.6 (1988), pp. 867–883. ISSN: 1096-9845. DOI: 10/cgqnx9.

[100] J. C. Wilson and B. S. Tan. "Bridge Abutments: Assessing Their Influence on Earthquake Response of Meloland Road Overpass". In: *Journal of Engineering Mechanics* 116.8 (Aug. 1990), pp. 1838–1856. ISSN: 0733-9399, 1943-7889. DOI: 10/bz9ttn.

[101] M. H. M. Yassin. "Nonlinear Analysis of Prestressed Concrete Structures under Monotonic and Cyclic Loads". PhD thesis. University of California, Berkeley, 1994.

[102] H. Yun et al. "Monitoring the collision of a cargo ship with the Vincent Thomas Bridge". en. In: *Structural Control and Health Monitoring* 15.2 (2008). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/stc.213, pp. 183–206. ISSN: 1545-2263. DOI: 10.1002/stc.213.

[103] J. Zhang and N. Makris. "Seismic Response Analysis of Highway Overcrossings Including Soil-Structure Interaction". In: *Earthquake Engineering & Structural Dynamics* 31.11 (Nov. 2002), pp. 1967–1991. ISSN: 0098-8847, 1096-9845. DOI: 10/bdj3ps.

[104] J. Zhang and N. Makris. “Seismic response analysis of highway overcrossings including soilstructure interaction”. en. In: *Earthquake Engineering & Structural Dynamics* 31.11 (2002). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/eqe.197, pp. 1967–1991. ISSN: 1096-9845. DOI: 10.1002/eqe.197.

[105] Y. Zhang and J. P. Lynch. “Long-Term Modal Analysis of the New Carquinez Long-Span Suspension Bridge”. en. In: *Topics in Dynamics of Bridges, Volume 3*. Ed. by A. Cunha. Conference Proceedings of the Society for Experimental Mechanics Series. New York, NY: Springer, 2013, pp. 73–82. ISBN: 978-1-4614-6519-5. DOI: 10.1007/978-1-4614-6519-5_8.

[106] Y. Zhang et al. “Automated Modal Parameter Extraction and Statistical Analysis of the New Carquinez Bridge Response to Ambient Excitations”. en. In: *Model Validation and Uncertainty Quantification, Volume 3*. Ed. by H. S. Atamturktur et al. Conference Proceedings of the Society for Experimental Mechanics Series. Cham: Springer International Publishing, 2014, pp. 161–170. ISBN: 978-3-319-04552-8. DOI: 10.1007/978-3-319-04552-8_15.

[107] Y. Zhang et al. “Two-Dimensional Nonlinear Earthquake Response Analysis of a Bridge-Foundation-Ground System”. en. In: *Earthquake Spectra* 24.2 (May 2008). Publisher: SAGE Publications Ltd STM, pp. 343–386. ISSN: 8755-2930. DOI: 10.1193/1.2923925.

Appendix A: Sample Tables of Bridge Inventory

23-0015R Vallejo - Carquinez/I80 East Bridge

Site Details

23-0015R Table 1

Highway Agency District	04 - District 4
County Name	095 - Solano County
Place Code	0
Structure Number	23 0015R
Location	04-SOL-080-0.01
Year Built	1958
Approach Roadway Width	51.8
Bridge Median Code	0 - No Median
Skew Angle	0
Historical Significance Code	5 - Not National Register eligible
Type of Service on Bridge Code	1 - Highway
Type Of Service Under Bridge Code	8 - Highway-waterway-railroad
Length of Maximum Span	1100.1
Structure Length	5209
Left Curb/Sidewalk Width	1.6
Right Curb/Sidewalk Width	1.6
Bridge Roadway Width Curb to Curb	51.8
Deck Width - Out to Out	57.7
Underwater Inspection	Y60
Federal Lands Highways Code	0 - N/A
Metropolitan Planning Organization (MPO)	6197001 - Metropolitan Transportation Commission, CA
U.S. Congressional District	05 - Congressional District 5
State Senate District	003 - State Senate District 3
State House District	014 - State House District 14
City (InfoBridge Place Code-Name)	17274 - Crockett CDP

23-0015R Table 2

No. of Spans	5
Plan Shape	Straight.
Total Length	3350' (1021.1m). Bridge spans range from 150' (45.7m) to 1100' (335.3m).
Width of Deck	52' (15.8m)

23-0015R Table 3

Superstructure Type	Main span is a steel truss structure typically consisting of built-up steel welded box members with perforated plates on two sides. Connections are high-strength bolted. Abutment 1 is a seat abutment.
Substructure Type	Steel truss towers.
Foundation Type	Abutment 1 is supported on a spread footing. Piers 2 and 3 are supported on concrete caissons. Pier 4 is supported on steel H-piles. Pier 5 is supported on a hollow concrete shaft.

23-0015R Table 4

Base Highway Network	1 - On Base Network
Toll Status	1 - Toll Bridge
Maintenance Responsibility	1 - State Highway Agency
Design Load Descriptor	5 - MS 18 / HS 20
Structure Flared	1 - Flared
Bridge Railings	1 - Meets currently acceptable standards
Transitions	1 - Meets currently acceptable standards
Approach Guardrail	1 - Meets currently acceptable standards
Bridge Guardrail Ends	1 - Meets currently acceptable standards
Deck Condition Rating	5 - Fair Condition
Superstructure Condition Rating	7 - Good Condition
Channel and Channel Protection Condition Rating	6 - Channel remediation in fair condition
Culverts Condition Rating	N - Not a culvert
Operating Rating Method Code	1 - Load Factor(LF)

Operating Rating	46.4
Inventory Rating Method Code	1 - Load Factor(LF)
Inventory Rating	27.900000
Structural Evaluation Appraisal	6 - Equal to Present Minimum Criteria
Waterway Adequacy Appraisal	8 - Equal to Present Desirable Criteria
Approach Alignment Appraisal	8 - Equal to Present Desirable Criteria
Scour Critical Bridge Value	8 - Foundations stable; Scour above top of footing
Bridge Condition	F - Fair
Condition Code	5 - Fair Condition
Underclearance Appraisal Vertical and Horizontal	3 - Intolerable; High Priority Corrective Action
STRAHNET Highway Designation	1 - Interstate STRAHNET
Main Span Material	4 - Steel Continuous
Approach Spans Material	3 - Steel
Approach Spans Design	2 - Stringer/Multi-beam or Girder
Deck Geometry Appraisal	4 - Minimum Tolerable
Substructure Condition Rating	7 - Good Condition

23-0015R Table 5

Route Signing Prefix Code	1 - Interstate Highway
Designated Level of Service Code	1 - Mainline
Route Number	00080
Directional Suffix Code	0 - Not Applicable
Features Intersected	CARQUINEZ STRAIT,UP RR
Facility Carried By Structure	INTERSTATE 80 EB
Inventory Route - Minimum Vertical Clearance	20.5000000
Mile Point	0.006
LRS Inventory Route	0000000080
Subroute Number	1
Bypass or Detour Length	29.800000
Functional Class Of Inventory Route	11 - Urban Principal Arterial - Interstate
Lanes On the Structure	4
Lanes Under the Structure	14
Average Daily Traffic	128000
Year of Average Daily Traffic	2019
Navigation Control Code	1 - Permit Required
Navigation Vertical Clearance	130.9
Navigation Horizontal Clearance	998
Inventory Route Total Horizontal Clearance	51.8000000
Minimum Vertical Clearance Over Bridge Roadway	20.5000000
Minimum Vertical Underclearance Reference Feature	H - Highway underneath
Minimum Vertical Underclearance	16.0000000
Minimum Lateral Underclearance Reference Feature	H - Highway underneath
Minimum Lateral Underclearance on Right	1.6000000
Minimum Lateral Underclearance on Left	"000"
Parallel Structure Designation Code	R - Right of set in direction of the inventory
Direction of Traffic Code	1 - 1 - way traffic
Inventory Route NHS Code	1 - On NHS
Average Daily Truck Traffic (Percent ADT)	6
Designated National Truck Network Code	1 - On National Truck Network
Future Average Daily Traffic	192000
Year of Future Average Daily Traffic	2040
Average Daily Truck Traffic (Volume)	7680

23-0015R Table 6

Structure Operational Status Code	A - Open

23-0015R Table 7

Main Span Design	10 - Truss - Thru
Number of Spans in Main Unit	5
Number of Approach Spans	18
Deck Structure Type Code	1 - Concrete Cast-in-Place

Wearing Surface Type Code	2 - Integral Concrete
Membrane Type Code	0 - None
Deck Protection Code	0 - None
Deck Area	300781.3

23-0015R Table 8

Pier Abutment Protection Code	2 - In place; functioning

23-0015R Table 9

Inspection Date	921
Designated Inspection Frequency	24
Fracture Critical Details	Y24
Other Special Inspection	N
Fracture Critical Detail Date	820
Underwater Inspection Date	418

33-0214L Hayward Hwy 580-238 Interchange

Site Details

33-0214L Table 1

Highway Agency District	04 - District 4
County Name	001 - Alameda County
Place Code	0
Structure Number	33 0214L
Location	04-ALA-580-R30.59
Year Built	1988
Approach Roadway Width	50.9
Bridge Median Code	0 - No Median
Skew Angle	99 - 99 - Typical Skew not Defined
Historical Significance Code	5 - Not National Register eligible
Type of Service on Bridge Code	1 - Highway
Type Of Service Under Bridge Code	1 - Highway, with or without pedestrian
Length of Maximum Span	196.9
Structure Length	2029.9
Left Curb/Sidewalk Width	0
Right Curb/Sidewalk Width	0
Bridge Roadway Width Curb to Curb	50.9
Deck Width - Out to Out	54.5
Waterway Adequacy Appraisal	N - N/A
Underwater Inspection	N - Not needed
Federal Lands Highways Code	0 - N/A
Scour Critical Bridge Value	N - Bridge not over waterway
Metropolitan Planning Organization (MPO)	6197001 - Metropolitan Transportation Commission, CA
U.S. Congressional District	15 - Congressional District 15
State Senate District	010 - State Senate District 10
State House District	020 - State House District 20
City (InfoBridge Place Code-Name)	11964 - Castro Valley CDP

33-0214L Table 2

No. of Spans	14
Plan Shape	Curved
Total Length	2030' (618.7m). Bridge spans range from 92' (28.0m) to 197' (60.0m) long.
Width of Deck	54' (16.5m)

33-0214L Table 3

Superstructure Type	Concrete box girders.
Substructure Type	Octagonal concrete columns. 1 to 3 columns per bent.
Foundation Type	Bents are supported by concrete CIDH piles or spread footings. Abutments are supported by concrete CIDH piles or steel piles.

33-0214L Table 4

Base Highway Network	1 - On Base Network
Toll Status	3 - On Free Road
Maintenance Responsibility	1 - State Highway Agency
Design Load Descriptor	6 - MS 18+Mod / HS 20+Mod
Structure Flared	1 - Flared
Bridge Railings	1 - Meets currently acceptable standards
Transitions	1 - Meets currently acceptable standards
Approach Guardrail	1 - Meets currently acceptable standards
Bridge Guardrail Ends	1 - Meets currently acceptable standards
Deck Condition Rating	7 - Good Condition
Superstructure Condition Rating	7 - Good Condition
Channel and Channel Protection Condition Rating	N - Not over waterway
Culverts Condition Rating	N - Not a culvert
Operating Rating Method Code	A - Assigned rating based on Load Factor Design (LFD) reported in metric tons

Operating Rating	59.6
Inventory Rating Method Code	A - Assigned rating based on Load Factor Design (LFD) reported in metric tons
Inventory Rating	35.700000
Structural Evaluation Appraisal	5 - Better Than Minimum Adequacy
Approach Alignment Appraisal	8 - Equal to Present Desirable Criteria
Bridge Condition	F - Fair
Condition Code	5 - Fair Condition
Underclearance Appraisal Vertical and Horizontal	3 - Intolerable; High Priority Corrective Action
STRAHNET Highway Designation	1 - Interstate STRAHNET
Main Span Material	6 - Prestressed Concrete Continuous
Main Span Design	5 - Box Beam or Girders - Multiple
Approach Spans Material	0 - Other Material Main or N/A (No Other Span)
Deck Geometry Appraisal	5 - Better Than Minimum Adequacy
Substructure Condition Rating	5 - Fair Condition

33-0214L Table 5

Route Signing Prefix Code	1 - Interstate Highway
Designated Level of Service Code	1 - Mainline
Route Number	00580
Directional Suffix Code	0 - Not Applicable
Features Intersected	I 238 NB,CONNS,CASTRO VL
Facility Carried By Structure	I 580 EB
Inventory Route - Minimum Vertical Clearance	"9999"
Mile Point	19.008
LRS Inventory Route	0000000580
Subroute Number	1
Bypass or Detour Length	3.100000
Functional Class Of Inventory Route	11 - Urban Principal Arterial - Interstate
Lanes On the Structure	3
Lanes Under the Structure	33
Average Daily Traffic	102000
Year of Average Daily Traffic	2017
Navigation Control Code	N - No waterway
Navigation Vertical Clearance	0
Navigation Horizontal Clearance	0
Inventory Route Total Horizontal Clearance	50.9000000
Minimum Vertical Clearance Over Bridge Roadway	"9999"
Minimum Vertical Underclearance Reference Feature	H - Highway underneath
Minimum Vertical Underclearance	17.3000000
Minimum Lateral Underclearance Reference Feature	H - Highway underneath
Minimum Lateral Underclearance on Right	2.6000000
Minimum Lateral Underclearance on Left	5.5000000
Parallel Structure Designation Code	L - Left of set in opposite direction of inventory
Direction of Traffic Code	1 - 1 - way traffic
Inventory Route NHS Code	1 - On NHS
Average Daily Truck Traffic (Percent ADT)	6
Designated National Truck Network Code	1 - On National Truck Network
Future Average Daily Traffic	110289
Year of Future Average Daily Traffic	2037
Average Daily Truck Traffic (Volume)	6120

33-0214L Table 6

Structure Operational Status Code	A - Open

33-0214L Table 7

Approach Spans Design	0
Number of Spans in Main Unit	14
Number of Approach Spans	0
Deck Structure Type Code	1 - Concrete Cast-in-Place
Wearing Surface Type Code	0 - None
Membrane Type Code	0 - None

Deck Protection Code	0 - None
Deck Area	110549.8

	33-0214L Table 8
Inspection Date	420
Designated Inspection Frequency	24
Fracture Critical Details	N - Not needed
Other Special Inspection	N

Appendix B: Glossaries

Acronyms

BART Bay Area Rapid Transit. 58

Caltrans California Department of Transportation. v, 49, 58, 59, 79, 171

CESMD Center for Engineering Strong Motion Data. 49

CGS California Geological Survey. v, 49, 61, 86

CSMIP California Strong Motion Instrumentation Program. v, 49

CSN Community Seismic Network. 49

DOC California Department of Conservation. 58

DOF Degree Of Freedom. 67, 105, 106, 134

ERA Eigensystem Realization Algorithm. vi, 112, 113, 118

LTI Linear Time-Invariant. 104, 107, 112, 118

MDOF Multiple Degree Of Freedom. v, vi, 104–107, 124, 134

NRHA Nonlinear Response History Analysis. 26, 66, 67, 73, 76

OKID Observer Kalman Filter Identification. vi, 118

PGM Probabilistic Graphical Model. vi, 136, 137

SDC Caltrans Seismic Design Criteria. 79

SDOF Single Degree Of Freedom. v, 104, 123, 134

SRIM System Realization using Information Matrix. 119

SVD Singular Value Decomposition. 114, 119, 121

BRACE^2 General Terms

asset A unique physical object for which there is interest in its structural health. iv, 4–6, 8, 10, 11, 89, 92

digital twin A computational model of a unique physical asset that persistently reflects the asset's state, context and behavior. 1, 10, 47

evaluation A report on the health of an asset, generated by predictors, in response to an event. 8, 10

event A situation that could place the health of a structure in jeopardy, and for which an evaluation is desired. iv, 4, 8

health monitoring platform The platform that hosts the *BRACE*^{2} structural health monitoring framework and web application. 1, 3, 4, 6–9, 20, 22, 26, 27, 40, 87, 94

metric Computed quantities that can inform judgment calls. iv, 4, 7, 14

predictor An analysis technique that an engineer can configure beforehand to process data when events are triggered. iv, 1, 4, 5, 8, 10, 15, 19, 89, 93

Type I predictor A predictor that simulates estimated responses based on modeled behavior. iv, 1, 6, 10, 15, 16, 19–21, 27, 29

Type II predictor A predictor that performs data-driven identification of structural properties. vii, 1, 6, 10, 11, 15, 19, 40, 45