

Politecnico di Bari

Repository Istituzionale dei Prodotti della Ricerca del Politecnico di Bari

New approaches in structural health monitoring: model updating using automated intelligent algorithms

This is a PhD Thesis

Original Citation:
New approaches in structural health monitoring: model updating using automated intelligent algorithms / Bruno, Gianluca. - ELETTRONICO. - (2026). [10.60576/poliba/iris/bruno-gianluca_phd2026]

Availability:
This version is available at http://hdl.handle.net/11589/295402 since: 2026-01-09

Published version
DOI:10.60576/poliba/iris/bruno-gianluca_phd2026

Publisher: Politecnico di Bari

Terms of use:

(Article begins on next page)

22 April 2026

Abstract
The growing need for safety, efficiency, and durability in civil infrastructure calls for advanced tools to monitor and manage structural performance. Structural Health Monitoring (SHM), combined with Operational Modal Analysis (OMA), enables the continuous assessment of structural conditions, but robust model calibration remains a major challenge. This PhD research proposes an innovative, fully automatic framework for model updating that integrates real monitoring data with advanced artificial intelligence techniques. The methodology, developed in two versions, DRUM-Av1.0 and DRUM-Av2.0, combines Deep Reinforcement Learning (DRL) and physics-based Reduced Order Models (ROMs) to achieve accurate, efficient, and robust calibration of complex structures. Applications to both numerical and real-world case studies demonstrate the framework's capability to reduce computational costs, handle uncertainty, and automate the updating process. This study lays the groundwork for the development of intelligent, self-adaptive digital twins for the predictive and resilient management of civil engineering structures.

DICATECh

2026

Gianluca Bruno

New approaches in Structural Health Monitoring: Model Updating using automated intelligent algorithms

01

D.R.S.A.T.E.

PhD in Risk and Environmental, Territorial and Building Development

Coordinator: Prof. Francesco Fiorito

XXXVIII CYCLE
Curriculum: CEAR-07/A

DICATECh
Department of Civil, Environmental, Building Engineering and Chemistry

POLITECNICO DI BARI

01
2026

Gianluca Bruno

New approaches in Structural Health Monitoring: Model Updating using automated intelligent algorithms

Prof. Giuseppina UVA
Dr. Fabio PARISI
Dr. Sergio RUGGIERI
DICATECh, Politecnico di Bari

Prof. Eleni CHATZI
DBAUG, ETH Zürich



Politecnico di Bari

LIBERATORIA PER L'ARCHIVIAZIONE DELLA TESI DI DOTTORATO

Al Magnifico Rettore
del Politecnico di Bari

Il/la sottoscritto/a Gianluca Bruno nato/a a Andria il 08/12/1996

residente a Andria in via Torino 67 e-mail bruno.gianluca.gb@gmail.com

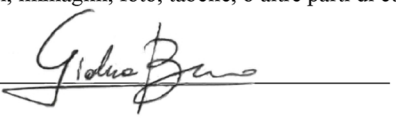
iscritto al 3° anno di Corso di Dottorato di Ricerca in Rischio,Sviluppo Ambientale,Territoriale ed Edilizio ciclo 38

ed essendo stato ammesso a sostenere l'esame finale con la prevista discussione della tesi dal titolo:

New approaches in Structural Health Monitoring: Model Updating using automated intelligent algorithms

DICHIARA

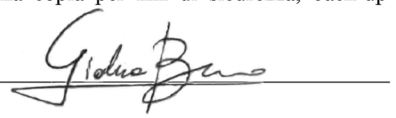
1) di essere consapevole che, ai sensi del D.P.R. n. 445 del 28.12.2000, le dichiarazioni mendaci, la falsità negli atti e l'uso di atti falsi sono puniti ai sensi del codice penale e delle Leggi speciali in materia, e che nel caso ricorressero dette ipotesi, decade fin dall'inizio e senza necessità di nessuna formalità dai benefici conseguenti al provvedimento emanato sulla base di tali dichiarazioni;
2) di essere iscritto al Corso di Dottorato di ricerca Rischio,Sviluppo Ambientale,Territoriale ed Edilizio ciclo 38, corso attivato ai sensi del "*Regolamento dei Corsi di Dottorato di ricerca del Politecnico di Bari"*, emanato con D.R. n.286 del 01.07.2013;
3) di essere pienamente a conoscenza delle disposizioni contenute nel predetto Regolamento in merito alla procedura di deposito, pubblicazione e autoarchiviazione della tesi di dottorato nell'Archivio Istituzionale ad accesso aperto alla letteratura scientifica;
4) di essere consapevole che attraverso l'autoarchiviazione delle tesi nell'Archivio Istituzionale ad accesso aperto alla letteratura scientifica del Politecnico di Bari (IRIS-POLIBA), l'Ateneo archivierà e renderà consultabile in rete (nel rispetto della Policy di Ateneo di cui al D.R. 642 del 13.11.2015) il testo completo della tesi di dottorato, fatta salva la possibilità di sottoscrizione di apposite licenze per le relative condizioni di utilizzo (di cui al sito http://www.creativecommons.it/Licenze), e fatte salve, altresì, le eventuali esigenze di "embargo", legate a strette considerazioni sulla tutelabilità e sfruttamento industriale/commerciale dei contenuti della tesi, da rappresentarsi mediante compilazione e sottoscrizione del modulo in calce (Richiesta di embargo);
5) che la tesi da depositare in IRIS-POLIBA, in formato digitale (PDF/A) sarà del tutto identica a quelle **consegnate**/inviate/da inviarsi ai componenti della commissione per l'esame finale e a qualsiasi altra copia depositata presso gli Uffici del Politecnico di Bari in forma cartacea o digitale, ovvero a quella da discutere in sede di esame finale, a quella da depositare, a cura dell'Ateneo, presso le Biblioteche Nazionali Centrali di Roma e Firenze e presso tutti gli Uffici competenti per legge al momento del deposito stesso, e che di conseguenza va esclusa qualsiasi responsabilità del Politecnico di Bari per quanto riguarda eventuali errori, imprecisioni o omissioni nei contenuti della tesi;
6) che il contenuto e l'organizzazione della tesi è opera originale realizzata dal sottoscritto e non compromette in alcun modo i diritti di terzi, ivi compresi quelli relativi alla sicurezza dei dati personali; che pertanto il Politecnico di Bari ed i suoi funzionari sono in ogni caso esenti da responsabilità di qualsivoglia natura: civile, amministrativa e penale e saranno dal sottoscritto tenuti indenni da qualsiasi richiesta o rivendicazione da parte di terzi;
7) che il contenuto della tesi non infrange in alcun modo il diritto d'Autore né gli obblighi connessi alla salvaguardia di diritti morali od economici di altri autori o di altri aventi diritto, sia per testi, immagini, foto, tabelle, o altre parti di cui la tesi è composta.

Luogo e data Bari 09/01/2026 Firma 

Il/La sottoscritto, con l'autoarchiviazione della propria tesi di dottorato nell'Archivio Istituzionale ad accesso aperto del Politecnico di Bari (POLIBA-IRIS), pur mantenendo su di essa tutti i diritti d'autore, morali ed economici, ai sensi della normativa vigente (Legge 633/1941 e ss.mm.ii.),

CONCEDE

- al Politecnico di Bari il permesso di trasferire l'opera su qualsiasi supporto e di convertirla in qualsiasi formato al fine di una corretta conservazione nel tempo. Il Politecnico di Bari garantisce che non verrà effettuata alcuna modifica al contenuto e alla struttura dell'opera.
- al Politecnico di Bari la possibilità di riprodurre l'opera in più di una copia per fini di sicurezza, back-up e conservazione.

Luogo e data Bari 09/01/2026 Firma 

D.R.S.A.T.E.

PhD in Risk and Environmental, Territorial and Building Development

Coordinator: Prof. Francesco Fiorito

XXXVIII CYCLE
Curriculum: CEAR-07/A

DICATECh
Department of Civil, Environmental, Land Building Engineering and Chemistry

POLITECNICO DI BARI

01
2026

Gianluca Bruno

New approaches in Structural Health Monitoring: Model Updating using automated intelligent algorithms

Prof. Giuseppina UVA
Dr. Fabio PARISI
Dr. Sergio RUGGIERI
DICATECh, Politecnico di Bari

Prof. Eleni CHATZI
DBAUG, ETH Zürich



D.R.S.A.T.E.	**POLITECNICO DI BARI**	**01** **2026**
Dottorato di Ricerca in Rischio e Sviluppo Ambientale, Territoriale ed Edilizio		
Coordinatore: Prof. Francesco Fiorito		
XXXVIII CICLO Curriculum: CEAR-07/A		
DICATECh Dipartimento di Ingegneria Civile, Ambientale, del Territorio, Edile e di Chimica		
	Gianluca Bruno	
	Nuovi approcci nel monitoraggio strutturale: Algoritmi automatici e intelligenti per la calibrazione di modelli	
	Prof. Giuseppina UVA Dr. Fabio PARISI Dr. Sergio RUGGIERI DICATECh, Politecnico di Bari Prof. Eleni CHATZI DBAUG, ETH Zürich	
		

EXTENDED ABSTRACT *(eng)*

The increasing necessity to ensure high levels of safety, efficiency, and durability in civil infrastructure and existing structures is one of the central and current topics in structural engineering. Buildings, bridges, and other strategic works, often built according to design criteria that are now obsolete, are currently exposed to degradation processes and a significant increase in uncertainty regarding knowledge of material characteristics, constraints, and load conditions. The management of these uncertainties is not only a technical challenge, but also a social and economic one, as the risk of damage or malfunction can have a considerable impact on communities and the environment. In this context, Structural Health Monitoring (SHM) has established itself as an essential tool for continuous and timely diagnosis of structural conditions. Modern monitoring campaigns, based on the systematic acquisition of dynamic responses and the application of Operational Modal Analysis (OMA), allow significant global dynamic parameters to be obtained, including frequencies and modal shapes. However, data acquisition alone is not sufficient to ensure a robust assessment of structural performance, especially in the presence of exceptional actions, changes in use, or degradation phenomena. In these cases, the calibration phase of numerical models, known as model updating (MU), is of fundamental importance. It is necessary to obtain the digital twin of the structure, reduce the uncertainty of the analyses, and support more efficient predictive maintenance strategies. Traditional MU strategies, both deterministic and probabilistic, are widely used in scientific practice, but they have evident limitations: they require strong a priori assumptions, are computationally expensive, and are

often difficult to apply to systems characterized by high dimensionality of uncertain parameters. Automation and intelligent efficiency of the updating process are now considered priority objectives by the international scientific community. The objective of this PhD research is to contribute to methodological advances in the field by integrating experimental data from real SHM campaigns with the latest generation of artificial intelligence techniques. The core of the proposal is to develop, validate, and compare an advanced framework for fully automatic model updating of complex structures, based on the combined use of physically based Reduced Order Modeling (ROM) and Deep Reinforcement Learning (DRL). The key idea is to manage calibration through an intelligent agent capable of interacting with the numerical model and real data, progressively learning (through reward mechanisms typical of DRL) the updating strategies that best reduce the discrepancy between the simulated response and the real response of the structure. The framework was designed and organised in five consecutive stages. Given the monitoring campaign data, the definition of the target variables of interest and a numerical model of the structure in an externally accessible environment, the proposed approach defines a DRL engine, in which a strategy for fine-tuning the structural control parameters was defined. At the same time, the training/testing phases of the intelligent agent were carried out to obtain the correct solution based on a statistical evaluation of the results. The methodology, divided into DRUM-Av1.0 and DRUM-Av2.0 versions, is applied and validated both on theoretical case studies and on real infrastructures, such as bridges and reinforced concrete frames. In the advanced version of the framework (DRUM-Av2.0), the introduction of ROMs, constructed using decomposition techniques from Full Order Models (FOMs), allows for a significant reduction in the calculation times required to train and validate the DRL agent, making the methodology applicable even to systems characterized by a high number of degrees of freedom and unknown parameters. The quantitative results obtained in the various case studies show that the approach: (a) guarantees high accuracy in the estimation of uncertain parameters; (b) provides higher computational efficiency than classical methods; (c) is robust with respect to interferences caused by incomplete information or experimental noise; (d) allows for almost complete automation of the calibration process.

Furthermore, in-depth analysis highlights the possibility of extending the target variables (e.g., by including modal shapes and data from in situ tests), as well as the future development of intelligent agents equipped with physics-based engineering knowledge, for an even more reliable and interpretable solution to structural updating problems. These prospects drive the methodology towards the achievement of an intelligent and self-adaptive digital twin, able to learn and optimise structural management throughout the entire life cycle of structures.

In conclusion, the work developed in this thesis defines and validates an innovative, efficient and scalable approach for the automatic model updating of structural models. The approach responds to the needs of modern research, proposing implementable tools for the predictive, resilient and adaptive management of civil engineering works, and provides a solid foundation for the evolution of structural digital twins in the field of structural engineering.

keywords

Structural Health Monitoring; Model Updating; Complex structures; Uncertainty; Full-Order Model; Reduce Order Model; Deep Reinforcement Learning; Digital Twin.

EXTENDED ABSTRACT *(ita)*

L'incremento della necessità di garantire livelli elevati di sicurezza, efficienza e durabilità alle infrastrutture civili e alle strutture esistenti rappresenta uno dei temi centrali ed attuali dell'ingegneria strutturale. Edifici, ponti e altre opere strategiche, spesso realizzati secondo criteri progettuali oggi superati, sono oggi esposti a processi di degrado ed ad un significativo incremento dell'incertezza riguardo alla conoscenza delle caratteristiche dei materiali, vincoli e condizioni di carico. La gestione di queste incertezze costituisce una sfida non solo tecnica, ma anche sociale ed economica, poiché il rischio di danni o malfunzionamenti può avere impatti considerevoli sulle comunità e sull'ambiente circostante. In questo contesto, il Structural Health Monitoring (SHM) si afferma come strumento imprescindibile per la diagnosi continua e tempestiva delle condizioni strutturali. Le moderne campagne di monitoraggio, basate sull'acquisizione sistematica delle risposte dinamiche e sull'applicazione dell'Operational Modal Analysis (OMA), consentono di ottenere parametri dinamici globali significativi, tra cui frequenze e forme modali. Tuttavia, la sola acquisizione di dati non è sufficiente a assicurare una valutazione robusta delle prestazioni strutturali, soprattutto in presenza di azioni eccezionali, variazioni d'uso o fenomeni di degrado. In questi casi, è di fondamentale importanza la fase di calibrazione dei modelli numerici, nota come model updating (MU), necessaria per ottenere il gemello digitale della struttura, ridurre l'incertezza delle analisi e supportare strategie di manutenzione predittiva più efficienti. Le strategie tradizionali di MU, sia di tipo deterministico che probabilistico, sono ampiamente utilizzate nella pratica scientifica, ma presentano limiti evidenti: richiedono forti ipotesi a priori, sono onerose dal punto di vista computazionale e spesso difficili da applicare a sistemi caratterizzati da elevata dimensionalità dei parametri incerto. L'automazione e l'efficientamento intelligente del processo di aggiornamento sono oggi considerati obiettivi prioritari dalla comunità scientifica internazionale. L'obiettivo della ricerca di questo dottorato

è di contribuire all'avanzamento metodologico del settore, integrando dati sperimentali provenienti da reali campagne di SHM con tecniche di intelligenza artificiale di più recente generazione. Il cuore della proposta risiede nello sviluppo, validazione e confronto di un framework avanzato per il model updating completamente automatico di strutture complesse, basato sull'impiego congiunto di Reduced Order Modelling (ROM) fisicamente basato e Deep Reinforcement Learning (DRL). L'idea cardine è gestire la calibrazione tramite un agente intelligente, in grado di interagire con il modello numerico e i dati reali, apprendendo progressivamente (attraverso meccanismi di ricompensa tipici del DRL) le strategie di aggiornamento che meglio riducono la discrepanza tra risposta simulata e risposta reale della struttura. Il framework è stato ideato e organizzato in cinque fasi consecutive. Dati i dati della campagna di monitoraggio, la definizione delle variabili target di interesse e un modello numerico della struttura in un ambiente accessibile esternamente, l'approccio proposto definisce un motore DRL, in cui è stata definita una strategia per la messa a punto dei parametri di controllo strutturale. Contestualmente, sono state eseguite le fasi di addestramento/test dell'agente intelligente, per ricavare la soluzione corretta in base a una valutazione statistica dei risultati. La metodologia, articolata nelle versioni DRUM-Av1.0 e DRUM-Av2.0, viene applicata e validata sia su casi studio teorici che su infrastrutture reali, come ponti e telai in calcestruzzo armato. Nella versione avanzata del framework (DRUM-Av2.0), l'introduzione del ROM, costruiti tramite tecniche di decomposizione dai Full Order Model (FOM), consente una significativa riduzione dei tempi di calcolo necessari per addestrare e validare l'agente DRL, rendendo la metodologia applicabile anche a sistemi caratterizzati da un numero elevato di gradi di libertà e parametri sconosciuti.

I risultati quantitativi ottenuti nei diversi casi studio mostrano che l'approccio: (a) garantisce accuratezza elevata nella stima dei parametri incerti; (b) fornisce efficienza computazionale superiore rispetto ai metodi classici; (c) risulta robusto rispetto ai disturbi dati da informazioni incomplete o rumore sperimentale; (d) permette un'automazione quasi completa del processo di calibrazione.

Inoltre, l'analisi approfondita evidenzia la possibilità di estendere le variabili target (es. includendo forme modali e dati da prove in situ), così come lo sviluppo futuro

di agenti intelligenti dotati di conoscenza ingegneristica “physics-based”, per una soluzione ancora più affidabile e interpretabile dei problemi di aggiornamento strutturale. Tali prospettive orientano la metodologia verso il raggiungimento di un digital twin intelligente e auto-adattivo, capace di apprendere e ottimizzare la gestione strutturale lungo tutto il ciclo di vita delle strutture.

In conclusione, il lavoro sviluppato in questa tesi si definisce e valida un percorso innovativo, efficiente e scalabile per il model updating automatico di modelli strutturali. L’approccio risponde alle esigenze della ricerca moderna, proponendo strumenti implementabili per la gestione predittiva, resiliente e adattiva delle opere civili, e pone fondamenta solide per l’evoluzione dei digital twin strutturali nel panorama dell’ingegneria delle strutture.

keywords

Monitoraggio Strutturale, Calibrazione Modelli, Incertezza, Strutture complesse, Modelli di ordine completo, Modelli di ordine ridotto, Apprendimento per Rinforzo Profondo, Gemello Digitale.

INDEX

INTRODUCTION **1**
Motivations 1
Objectives 4
Outline of the thesis 5
1. Background of Structural Health Monitoring and Model Updating **9**
1.1 STRUCTURAL HEALTH MONITORING 9
1.1.1 Design of the system: sensors, placement and measurement 10
1.1.2 Data pre-processing 12
1.1.3 Dynamic Identification algorithms 13
1.1.4 Validation of identified modal forms 16
1.2 MODEL UPDATING 20
1.2.1 Deterministic methods 21
1.2.2 Probabilistic methods 23
1.2.3 Data driven and ML-based methods 25
2. Application of classical Structural Health Monitoring in a real case-study **27**
2.1 Description of the building 28
2.2 Monitoring campaign set up 32
2.2.1 Sensors 32
2.2.2 Placement 34
2.2.3 Measurements 37
2.3 Vibration levels analysis 38
2.4 Operational Modal Analysis 42
2.5 Model Updating phase 45
2.6 Discussion 49
3. Model Updating with use of Deep Reinforcement Learning **53**

3.1 DRL basics 54
3.2 DRUM-Av1.0 57
3.2.1 Step1: SHM and definition of the target variables 58
3.2.2 Step2: definition of the FOM 59
3.2.3 Step3: DRL engine 61
3.2.4 Step4: agent's training 64
3.2.5 Step5: model selection 67
3.3 Dummy case study 68
3.3.1 Surrogate SHM and NM 70
3.3.2 Definition of DRL-engine 72
3.3.3 Training and test of the agent 77
3.3.4 Comparative analysis with traditional MU process 84
3.4 Real case study 89
3.4.1 Surrogate SHM and NM 90
3.4.2 Definition of DRL-engine 93
3.4.3 Training and test of the agent 93
3.5 Discussion 97
4. Integration of ROM in the DRL framework 101
4.1 Basics of Reduced Order Modelling 102
4.2 Combining use of the FOM and ROM 104
4.2.1 Definition of the FOM and dataset creation 104
4.2.2 Definition of the ROM 106
4.3 DRUM-Av2.0: algorithm and difference 107
4.3.1 SHM campaign 108
4.3.2 Definition of the FOM and the ROM 109
4.3.3 Implementation of DRL engine 109
4.3.4 Agent's training phase 110
4.3.5 Validation step and model selection 112
4.4 First case study: one storey RC frame 113
4.4.1 Application of DRUM-Av2.0 114

4.4.2 Comparative analysis of DRUM-Av1.0 and DRUM-Av2.0 124
4.5 Second case study: two storey RC frame 125
4.5.1 Application of DRUM-Av2.0 128
4.5.2 Efficiency of the framework 134
4.6 Discussion 135
5. FINAL REMARKS 137
5.1 Summary 137
5.2 Future research 140
6. ACKNOWLEDGEMENTS Errore. Il segnalibro non è definito.
7. LIST OF FIGURES 143
8. LIST OF TABLES 150
9. LIST OF ACRONYMS 153
10. LIST OF SYMBOLS 155
11. CURRICULUM 179

INTRODUCTION

Motivations

The rapidly increasing need to ensure the safety, efficiency and durability of existing buildings and civil infrastructure is an open challenge for the engineering community. Buildings, bridges and strategic infrastructure, often built decades ago according to outdated criteria, are now exposed to both progressive deterioration and a significant increase in uncertainties regarding materials, constraints and load conditions. Managing these uncertainties is not only a technical challenge, but also a social and economic one, as the potential impact of damage or malfunction can be dramatic for communities and the surrounding environment. In this scenario, continuous structural health monitoring (SHM) is establishing itself as an essential tool for the timely diagnosis of the behaviour and health conditions of structures (Farrar and Worden, 2017; Giagopoulos et al., 2019). The discipline of SHM is based on the systematic recording of structural response data, in particular through dynamic monitoring and operational modal analysis (OMA), which enable significant dynamic parameters such as modal frequencies and shapes to be obtained (Brincker et al., 2020; Rainieri and Fabbrocino, 2014). However, in order to produce robust assessments of the structure's performance, e.g. under exceptional actions, changes in use or deterioration, the calibration phase, known as model updating (MU), of the numerical models is of fundamental importance in order to obtain the so-called structural digital twin. The integration of SHM systems allows a shift from maintenance policies based on fixed time intervals to predictive strategies conditioned by the real state of the structure, thus optimising interventions and reducing associated costs. Furthermore, MU allows for the continuous calibration and improvement of the numerical models used for structural assessment, ensuring a reduction in uncertainty and more reliable predictions of the health and residual life of the structure. Considering the trend in structural performance over time, it is possible to represent the decay curve, as shown in Figure 1. The decay trend is interrupted when the thresholds established by current regulations are reached

(Frangopol and Messervey, 2009). In addition, through routine or extraordinary maintenance, it is possible to modify the performance decay curve.

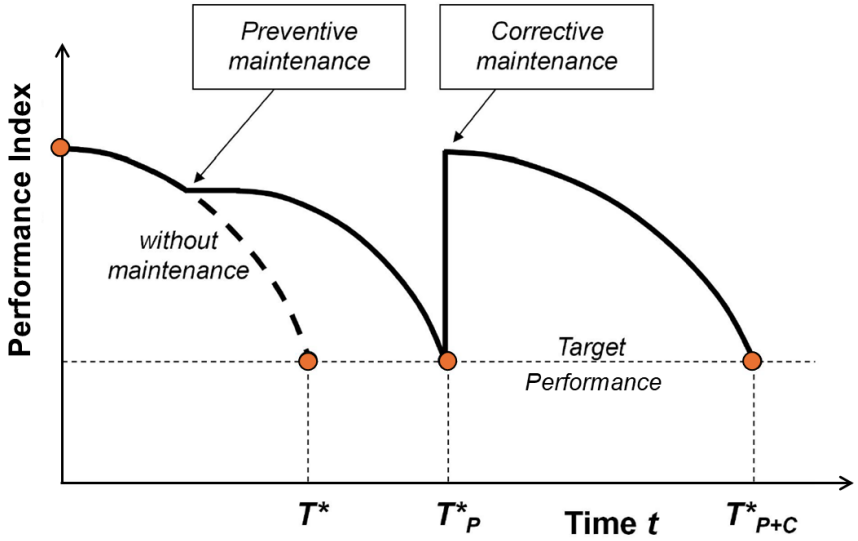


Figure 1. Performance of structural permanence in the case of preventive or corrective maintenance interventions

It is important to remember that the estimation of the decay curve of structural performance is linked to structural uncertainty values, which influence its characteristics. In particular, uncertainties may be related to:

- geometric characteristics of the individual element, for which it was not possible to carry out a detailed in situ survey;
- physical characteristics of the material (density) of the individual element or homogeneous family of elements, for which it was not possible to carry out in situ surveys;
- mechanical characteristics of the material (strength) of the individual element or homogeneous family of elements, for which it was not possible to carry out in situ investigations;
- degree of external constraint of the structure, for which the actual behaviour is not known;

- degree of internal constraint between individual elements, for which the actual behaviour is not known.

These become the parameters to be updated during the model updating process.

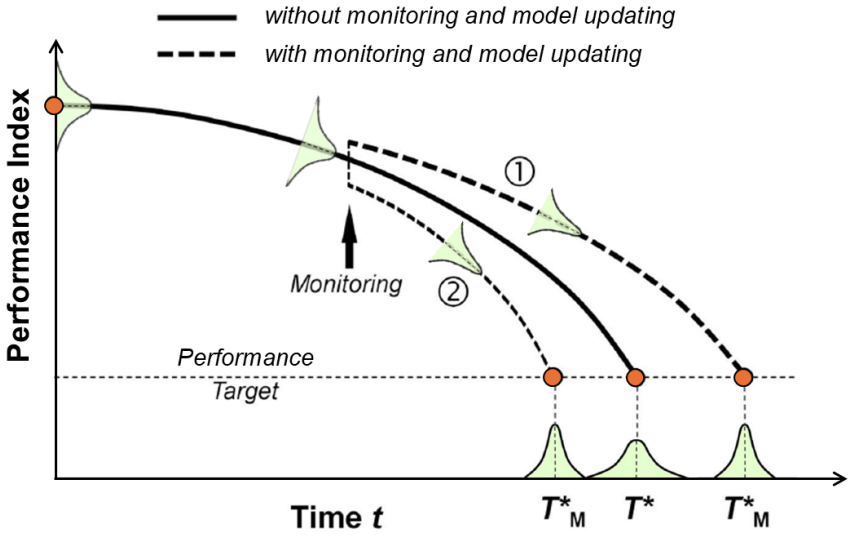


Figure 2. Structural performance trends and probability distribution of structural uncertainty in structures with and without monitoring + model updating

Considering structural performance, the SHM and MU processes can reduce the uncertainty of analyses, allowing for a more accurate estimation of performance loss until the threshold value is reached (Figure 2). In particular, the new estimated value T*m may be $<$ T*, in which case maintenance is required to slow down the decay process; or T*m may be $>$ T*, in which case it is possible to postpone the intervention, generating economic and computational savings.

Among the main MU methods, three broad families can be distinguished: (a) classical deterministic strategies, based essentially on trial-and-error analysis or sensitivity analysis (Aloisio et al., 2022; Avci et al., 2022; Aloisio et al., 2020; Kurent et al., 2021; Cabboi et al., 2017); (b) probabilistic strategies, which enable the calibration problem to be solved by calculating a probability value for the uncertain parameters of the model (Lye et al., 2021; Behmanesh et al., 2015; Rocchetta et al., 2017); (c) the latest techniques based on the use of Machine Learning (ML) technique (Yuan et al.,

2022) and deep learning (DL) techniques (Padil et al., 2016; Sabamehr et al., 2018; Naranjo-Pérez et al., 2020). The first two strategies, although very robust from a scientific point of view, have some practical limitations (described in more detail in Chapter 1). These include the need for strong a priori assumptions on the part of the analyst, which also affects accuracy, and increasing computational costs as the complexity and uncertainty of the structural parameters increase. In order to overcome these critical issues, recent research proposes the adoption of strategies based on the use of ML and DL. This field represents a real paradigm shift, allowing for the automation of the calibration process. However, these innovative MU strategies suffer from some critical issues, such as the high computational costs associated with the training phases of ML models. Solving the problem of MU for complex structures in an automated and continuous way remains a research task in which it is necessary to invest resources in order to achieve useful and easily implementable objectives in the field of structural engineering.

In this perspective, the motivation behind this thesis is to contribute to methodological advancement by integrating latest-generation artificial intelligence techniques with data derived from real SHM campaigns, in order to provide robust, rapid and efficient tools for model updating of complex structures and obtaining digital twins for their intelligent management throughout their life cycle.

Objectives

In the context of modern engineering challenges, the necessity to automate and improve the efficiency of structural model updating is becoming increasingly evident, especially given the growing availability of data from continuous monitoring systems. In order to meet this challenge, the main objective of this work is to develop an automatic framework for calibrating structural models based on the use of automatic algorithms, analysing the possibility of using modern Deep Reinforcement Learning (DRL) techniques, ensuring good accuracy, efficiency and automation in the model updating of complex structures.

Concretely, the work aims to explore and validate a new methodological paradigm, in which the model calibration phase is managed by an intelligent agent capable

of interacting with the numerical model and experimental data, progressively learning, through the reward mechanism typical of DRL, which update strategies are most effective in reducing the error between the simulated and real responses of the structure. This solution, inspired by recent applications of DRL in engineering (Fu et al., 2024; Bleker et al., 2024; Morato et al., 2023; Arcieri et al., 2024), aims to overcome the limitations of traditional procedures, which are often characterised by high computational times, heavy reliance on the analyst's experience and difficulties in managing large spaces of uncertain parameters. The proposed automation therefore hopes to significantly reduce human intervention, provide more accurate and reproducible results, and enable continuous calibration with a view to structural digital twins. The specific objectives driving the work can be summarised as follows:

- Analyse the main stages of a continuous real-time monitoring campaign and subsequent model updating phase, in order to understand the key aspects of the entire process.
- Develop an end-to-end framework for model updating, integrating DRL and real SHM data, capable of operating completely automatically and without prior knowledge of the parameters of the initial model.
- Demonstrate, through representative case studies and benchmarking against classical methods and the latest machine learning approaches, that the proposed framework is capable of achieving greater accuracy and robustness in calibration, reducing computational time and costs.
- Validate the approach in realistic environments, addressing typical issues such as noise, incompleteness and uncertainty in experimental data.

Through these objectives, the thesis aims to make a concrete contribution to the progress of the sector, providing an advanced and implementable method that can be adopted for the intelligent, predictive and resilient management of civil infrastructure.

Outline of the thesis

The thesis is structured as follows, starting with the main characteristics of monitoring techniques and model updating, followed by the proposed methodology for solving the problem of numerical model calibration. Chapter 1 is divided into two parts.

The first part introduces and analyses the main characteristics of the SHM process. This section explains all the theoretical bases behind the data acquisition and processing process, the main dynamic identification algorithms known in the literature, and the techniques for checking and verifying the results. The second part describes the main model updating methods used by the scientific community. Analysing the main advantages and disadvantages of the methodologies and explaining the reasons behind the proposed work. Chapter 2 describes the main characteristics of a real continuous monitoring campaign, through the analysis of a practical case study of monitoring on a complex building (Ruggieri et al., 2024). This phase is extremely useful for learning about the main difficulties and potentialities of a real monitoring campaign. The techniques used for data acquisition, processing and dynamic identification are described in detail, in the specific case of a building subject to ordinary and extraordinary actions. Finally, the results obtained are analysed both in terms of vibrations induced on the building and in terms of vibrations induced on humans according to the limit values set by legislation (UNI 9614, 2017; ISO 2631/1, 1997; ISO 2631/2, 2003; UNI 9916, 2014; ISO 4866, 2010; DIN 4150-3, 2016), highlighting the advantages of using continuous dynamic monitoring techniques. This is in order to resolve the problem of numerical model calibration, reduce structural uncertainty values and obtain a model that is useful for predictive analysis. Given the motivations and objectives set out, Chapter 3 presents the model calibration framework through the use of DRL techniques using SHM data. The main characteristics of DRL problems and the different steps of the proposed procedure are reported (Bruno et al., 2025). Subsequently, the proposed procedure was applied to a dummy case study to assess its advantages and limitations compared to traditional methods. Instead, it was tested on a real case, a three-span reinforced concrete bridge, subject to continuous monitoring to assess its applicability in the case of real structures. From the first applications of the proposed methodology, it was observed that the main limitations were due to the direct use of the high-fidelity numerical model in the training and validation process of the DRL agent. This resulted in considerable computational time and effort required for the agent training phases. For these reasons, Chapter 4 presents an updated version of the proposed model

updating framework, based on the integrated use of Reduced-Order Models (ROM) and DRL techniques for the calibration of the high-fidelity numerical model. The theoretical aspects behind model order reduction processes and the techniques used are described. Then, its implementation in the calibration framework through DRL is described. The new framework was tested on two case studies to evaluate its advantages and disadvantages and main changes compared to the previous version. Chapter 5 discusses the main results of the entire thesis and the proposed framework, attempting to describe the possible future developments of this work.

1. Background of Structural Health Monitoring and Model Updating

This chapter introduces the fundamental concepts behind Structural Health Monitoring and Model Updating techniques, with the aim of outlining the theoretical and methodological framework of reference. SHM is a rapidly growing discipline focused on the continuous monitoring of structures in order to assess their condition over time, identify any failures and ensure safety and reliability. At the same time, model updating techniques help to reduce the gap between numerical simulation models and the real behaviour of structures, thus improving the predictive capability of analyses.

The first part presents the principles of SHM, with particular attention to the design of monitoring systems and dynamic identification procedures for structures. Subsequently, the focus shifts to model updating methodologies, illustrating both deterministic and probabilistic approaches. Finally, the most recent developments are discussed, including the use of machine learning techniques and surrogate models, such as artificial neural networks (ANN), Gaussian Process Regression (GPR) and other data-driven strategies, which are increasingly common in complex engineering applications. This overview thus provides the theoretical basis necessary to tackle the advanced analyses developed in the following chapters, focusing on the link between theory, numerical modelling and practical applications.

1.1 STRUCTURAL HEALTH MONITORING

Structural monitoring is a fundamental discipline in civil and mechanical engineering, aimed at continuously monitoring the condition of structures by tracking their dynamic responses over time. The introduction of SHM systems responds to the need to overcome the limitations of traditional visual inspection: by using sensor networks, these technologies enable the continuous acquisition of significant data on the condition of structures, providing timely alerts in the event of anomalies and supporting the diagnosis of overall health. Increased seismic vulnerability, the need to reduce maintenance costs and the drive to extend the useful life of infrastructure have led to the rapid evolution of SHM tools and techniques Barsocchi et al., 2020; Zonno et al., 2018; Kita

et al., 2021; Betti et al., 2022). Over the years, through various studies in the literature, a fixed structure for the entire monitoring process has been established, as described in Figure 3. The first phase, called data acquisition, is linked to the design of the monitoring system, in which the type of sensors to be used, their position and the characteristics of the measured signals must be defined. This is followed by a very important phase, which is the pre-processing of the signals to be used as input in the processing phase through dynamic identification algorithms. The output values from the identification algorithms must be saved in a suitable storage unit. Finally, there is the phase of analysis and verification of the results obtained and subsequent model updating.

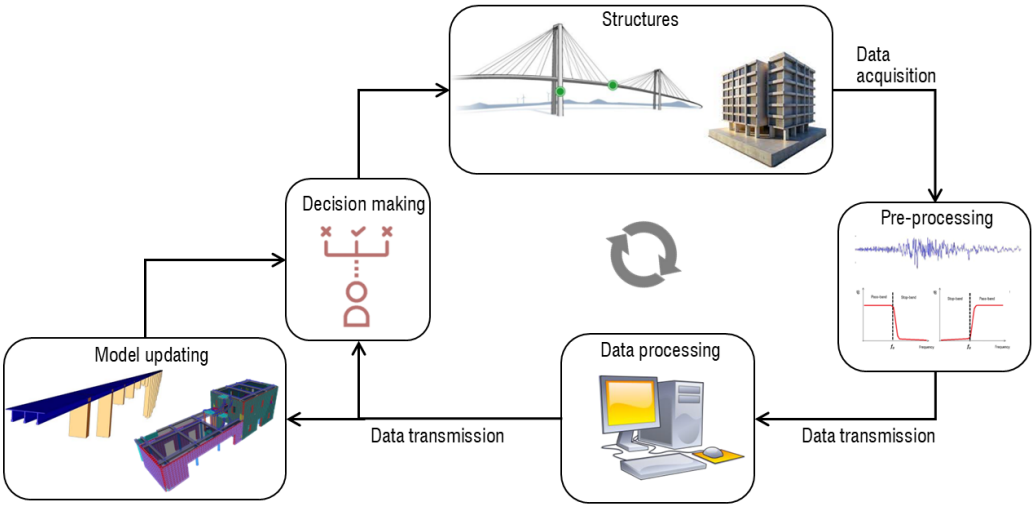


Figure 3. Structural Health Monitoring phases.

With modern technologies, the trend is towards easy-to-install monitoring systems and signal acquisition and processing units installed on structures (Morello et al., 2010; Jornet-Monteverde et al., 2021; Sindhuja and Kevildon, 2015). This allows for systems that aim to be autonomous, efficient and low cost (Sivasuriyan et al., 2021; Riggio and Dilmaghani, 2019).

1.1.1 Design of the system: sensors, placement and measurement

In order to carry out a good monitoring campaign, it is important to properly execute the initial system design phase. This consists of choosing three fundamental

components: (a) type of sensors; (b) placement; (c) measurement characteristics. (a) Sensors are divided into two families wired and wireless, among the wired ones the more traditional piezoelectric accelerometers (Johnson et al., 2003) and the latest fibre-optic sensors (Uva et al., 2014; Bado and Casas, 2021). Both have the advantage of high efficiency at the expense of ease of installation in the facility. In this case, each sensor must be connected to an acquisition unit that processes and saves the measurements. The second family of sensors is the wireless and micro-electromechanical systems (MEMS) (Sabato et al., 2016), to the more recent use of laser interferometer sensors and smartphones (Matarazzo et al., 2018; Guzman-Acevedo et al., 2019). The choice of the type of sensor to be used is very important and depends on the measurement to be obtained. (b) Depending on the desired output and the characteristics of the structure, it's essential to choose the correct sensor position. In the case of vibration-based SHM, one must identify those points that are representative of the global dynamic behaviour of the structure. In recent years this aspect has been the subject of interest of the scientific community, because the position and number of sensors is fundamental for a correct monitoring campaign (Sun & Büyüköztürk, 2015; Ostachowicz et al., 2019). As described in W. Ostachowicz et al. the Optimisation of Sensor Placement (OSP) problem is described by a linear framework that must take into account the type of sensors, the parameters measured and the costs of the monitoring campaign. Once these aspects have been defined, the problem can be solved by minimising an optimisation function (Kammer, 1991; Papadimitriou, 2004; Castro-Triguero et al., 2013; Meo & Zumpano, 2005). Other interesting results on this topic show the use of a finite element model to determine the number and positioning of sensors in a structure, such as the study presented in (Zini et al., 2022), where a method for assessing this problem is proposed. This method is based on the relationship between observed displacements and modal responses using Fisher's information matrix, where an appropriate sensor configuration maximises the norm of this matrix. (c) To be representative of the phenomenon to be analysed, measurements must be consistent and coherent. This is because the actual signals of civil structures are analogue and continuous, while to be processed and saved they must be converted into digital signals. The converted digital

signal must retain the same information as the original analogue signal. For this reason, the fundamental principles of sampling, quantisation and encoding must be observed. During the acquisition phase, it is important to consider the Nyquist–Shannon theorem, which plays a fundamental role in defining the correct sampling frequency. For a more detailed description, please refer to the studies in the literature on signal theory and data processing (Bendat and Allan, 2010; Rainieri and Fabbrocino, 2014; Brincker and Larsen, 2007).

1.1.2 Data pre-processing

In dynamic SHM, the pre-processing phase of accelerometer data is crucial to ensure the usability of accelerograms in modal and dynamic identification algorithms. The raw signals collected by accelerometers are frequently contaminated by instrumental or environmental noise, low-order drift and unwanted components, which can severely compromise the accuracy of subsequent processing (Gharehbaghi et al., 2022). Pre-processing consists of several key steps. The first is to remove noise from the signal using a mean value detrending filter. This is necessary and always present due to the characteristics of the digital signal. Next, digital filters are applied to eliminate undesirable frequency components. These filters are classified into three categories: i) low-pass filters, which attenuate high frequencies caused by environmental disturbances and electronic noise, while preserving the dynamic response; ii) high-pass filters, which eliminate very low frequency components due to sensor drift or insignificant movements, improving sensitivity to frequencies of interest; iii) band-pass filters, which allow a specific frequency range to be selected, typically that of the first natural frequencies, improving the overall cleanliness of the signal (Figure 4).

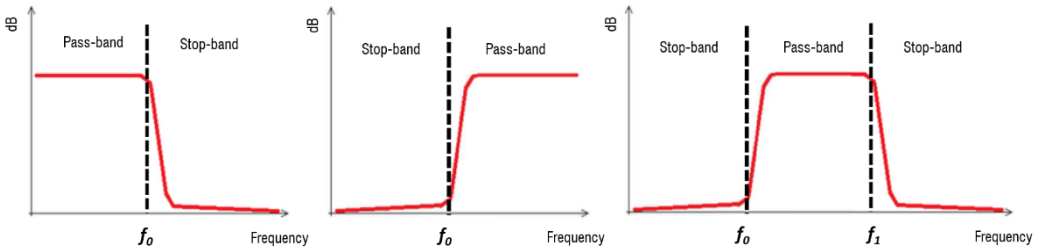


Figure 4. Categories of band filters for signals: (from left to right) i) low-pass filter; ii) high-pass filter; iii) band-pass filter.

After filtering, it is important to apply downsampling, i.e. reducing the sampling frequency after appropriate anti-aliasing filtering. This allows the amount of data to be reduced without losing useful frequency information, while still complying with Nyquist's criterion with respect to the filtered band. The processing often also includes normalisation and temporal segmentation techniques to optimise dynamic identification Zonta et al., 2014). All these operations allow the production of clean, reliable and optimal accelerograms as input for the main dynamic structural identification algorithms, avoiding limitations due to noise and drift and ensuring accurate extraction of modal parameters.

1.1.3 Dynamic Identification algorithms

With the data obtained during the previous phases, dynamic identification algorithms are applied. This is one of the most advanced aspects of structural monitoring, as it allows natural frequencies, modal shapes and damping ratios to be extracted directly from responses to environmental inputs. In the field of OMA, two families of algorithms can be distinguished according to the domain of implementation of the data used. The first family defines methods based on the Fourier transform of signals and indicates methods in the frequency domain (frequency-based approaches). Parameter estimation methods based directly on the processing of signal time histories, on the other hand, define the family of approaches in the time domain (time-based approaches). Time-domain approaches focus on the direct analysis of time series acquired by sensors (e.g., accelerations), studying how the signal evolves moment by moment. These strategies include methods such as Time Domain Decomposition

(TDD) (Kim et al., 2004) and the Stochastic Subspace Identification (SSI) (Van Overschee & De Moor, 1996). TDD aims to isolate elementary modal behaviours in each time series, from which natural frequencies and damping parameters can be determined, while SSI is based on the construction of stochastic dynamic models derived from experimental data and allows for the robust identification of modal parameters even in the presence of noise. In the context of the Stochastic Subspace Identification (SSI) algorithm, there are two main variants (Peeters and De Roeck, 1999): the data-driven version (SSI-DATA) and the covariance-driven version (SSI-COV). The distinctive feature between the two lies in the method of calculating modal parameters: in the data-driven approach, these parameters are estimated directly using experimental data acquired from the structure; on the contrary, with the covariance-driven version, data covariance matrices are used as a basis for extracting dynamic information. Although they are somewhat complex in terms of theory and implementation, SSI methods have established themselves due to their high precision in modal estimates and their robustness against noise and real operating conditions. Recent literature highlights numerous advances and optimisations of these variants, which make them suitable even for complex case studies and large structures (Qin et al., 2016; Reynders et al., 2015; Li et al., 2017). Other operational tools in the time domain include autoregressive and autoregressive-moving average models (Figueiredo et al., 2010; Gul & Catbas, 2009), in which the output is described as the result of a linear combination of past values and a stochastic component, making these methods very effective on large data sets but computationally intensive.

Methods developed in the frequency domain, on the other hand, aim to decompose the signal into its harmonic components and analyse the distribution of energy over specific frequency ranges. Here, techniques such as Peak Picking (Zahid et al., 2020), the Frequency Domain Decomposition (FDD) (Brincker and Zhang, 2009) and its developments allow the identification of natural frequencies, modal shapes and damping estimates starting from the power spectrum or Power Spectral Density matrix (PSD) of the response, exploiting peak localisation and singular value decomposition. The limitation of FDD is the lack of damping estimation. Several studies have developed

an updated version, called Enhanced FDD (EFDD), to estimate damping (Brincker et al., 2021). This involves the use of temporal autocorrelation, obtained by means of the inverse Fourier transform of the band around the modal peak, and the logarithmic decrease in the signal envelope. These strategies provide an immediate view of changes in the modal response and are particularly effective in diagnosing frequency variations due to damage, especially in structures mainly subjected to environmental excitation. Mathematically, the procedure for applying FDD starts from the estimation of the PSD matrix of the multichannel responses:

$$G_{yy}(\varpi) = H^{*}(\varpi)G_{xx}(\varpi)H^{T}(\varpi) \tag{1}$$

where $G_{yy}(\varpi)$ is the PSD matrix of the output measurements, $H(\varpi)$ is the matrix of Frequency Response Functions (FRF), while for environments with random inputs (white noise), $G_{xx}(\varpi)$ can be considered constant (Brincker & Zhang, 2009). The PSD matrix is evaluated for each frequency and subjected to Singular Value Decomposition (SVD):

$$G_{yy}(\varpi) = N(\varpi)\Sigma(\varpi)N^{H}(\varpi) \tag{2}$$

Where $N(\varpi)$ is the matrix of singular vectors (estimation of modal shapes) and $\Sigma(\varpi)$ is the matrix of singular values (modal autospectra). Near the modal resonance peak, the first singular vector provides a reliable estimate of the dominant modal shape, while the position of the peak of the first singular value identifies the natural frequency. In the presence of closely spaced modes, the successive singular vectors are also significant for reconstructing the corresponding modal shapes. In summary, FDD exploits the statistical properties of the environmental signal and SVD to robustly and automatically isolate, visualise and estimate the main dynamic parameters (frequency, modal shape, damping) of the monitored structure, without the need for controlled stresses.

1.1.4 Validation of identified modal forms

Through techniques in the time and frequency domains, both real modes, i.e. those that actually represent the structure, and spurious modes, which are representative of noise, are identified. Therefore, before moving on to the analysis phase of the results obtained through dynamic identification, it is important to validate the identified modes. To verify the consistency of the identified modes, certain techniques can be used, such as the Modal Assurance Criterion (MAC) (Allemang and Brown, 1982), defined by the equation:

$$MAC_{ij} = \frac{\left|\phi_i{}^T \phi_j\right|^2}{\left(\phi_i{}^T \phi_i\right)\left(\phi_j{}^T \phi_j\right)} \tag{3}$$

This expression provides a numerical index between 0 and 1, where values close to 1 indicate a high similarity between modal shapes, while values close to 0 indicate a low correlation. Verification using MAC is essential to validate the correct identification of modes and to compare the results between different identification techniques or with respect to finite element model predictions. The use of MAC allows reliable modes to be selected and any errors or biases in the estimation of modal shapes to be identified. Another method for evaluating the identified mode is the characterisation of the complex part, since the mode shape estimates are in the form of complex vectors. For these reasons, a distinction must be made between normal modes, characterised by mode shape vectors with real values, and complex modes. In the case of normal (real) modes, the displacements at the various degrees of freedom (DOF) reach their maximum at the same time and pass through the equilibrium position at the same time. This is not the case for complex modes, where both the maximum and zero values of the modal displacements are reached at different times for the various DOFs (Ewins, 2009). Consequently, while phase angles are all 0 or 180 for normal modes, both amplitude and phase characterise the movement of different DOFs in the case of complex modes. Complex modes can arise for a number of reasons, such as gyroscopic effects, aerodynamic effects, non-linearities and non-proportional damping (Ewins, 2009). The simplest method for assessing modal complexity consists in plotting the components

of the i-th eigenvector in the complex plane, thus obtaining the so-called 'Complexity Plot' (Figure 5). This allows the degree of complexity to be assessed by simple visual inspection. A method based on complexity plots for the quantitative assessment of the degree of complexity of the estimated mode shapes is described elsewhere. Other indices that measure the complexity of modal shapes are Modal Phase Collinearity (MPC) (Pappa et al., 1993) and the Mean Phase Deviation (MPD) (Heylen & Sas, 2005).

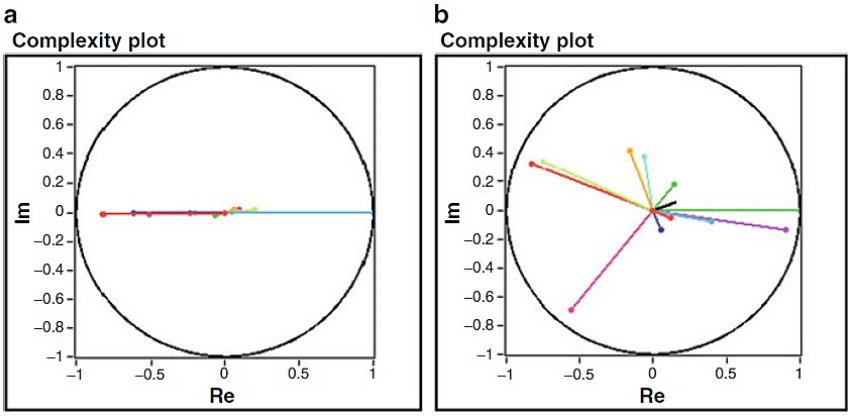


Figure 5. Complexity plot: a) real mode, b) complex mode (Rainieri and Fabbrocino, 2014).

Furthermore, through the graphical visualisation of the identified modal shapes, it is possible to distinguish between real and complex modes in a simple and preliminary manner. The tracking of animated modal shapes overcomes the limitations of static visualisation (shown in Figure 6). Animated graphs simulate the oscillation of the structure according to the selected mode shape, so they can also effectively represent complex modes. Therefore, the information obtained from the extraction of dynamic parameters must be configured so that the modal shapes obtained can be visualised. To do this, a few steps are necessary:

- *Creation of a wireframe spatial model,* that schematically represents the monitored structure. This model does not contain any information on the characteristics of the materials, applied loads or stiffness of the elements, but is simply a set of rods connecting points in space.

- *Identification in the wireframe model of the monitored points*, which correspond to the points where the sensors have been positioned. Each of these points is associated with the direction and orientation corresponding to the installed sensors.
- *Allocating the information obtained from identification to each monitored point,* so that it is possible to visualise how these points move in relation to each other in space, in a specific manner of vibration.

At this point, by assigning the information obtained from the identification process to each measured point, it is possible to visualise how the various points move at a given vibration mode, i.e. whether the measured point is moving in phase (+) or in opposition to phase (-) with respect to the reference direction of the sensor. One of the main disadvantages of visual inspection of the identified mode shapes is that the identified model is incomplete, as measurements are taken at only a few points. The number of measurement points is typically much lower (by several orders of magnitude) than the number of DOFs adopted, for example, in numerical models. Since there are several DOFs that are not measured during the tests, in the grid of points adopted to represent the geometry of the structure, some of them will always be characterised by zero modal displacements. This effect, caused by the finite number of measurement channels, must be taken into account in the visual inspection of modal shapes. In fact, these points probably show some displacements that are not represented in the graphs. Neglecting the effect of incomplete measurements can lead to false results in the interpretation of experimental modal shapes. In some cases, the interpolation of modal displacements or the consideration of certain constraints (e.g., the presence of rigid diaphragms) can compensate for the missing information on some modal displacements in experimental estimates. This solution is applicable in software through the creation of interpolating functions between nodes (Rainieri and Fabbrocino, 2014).

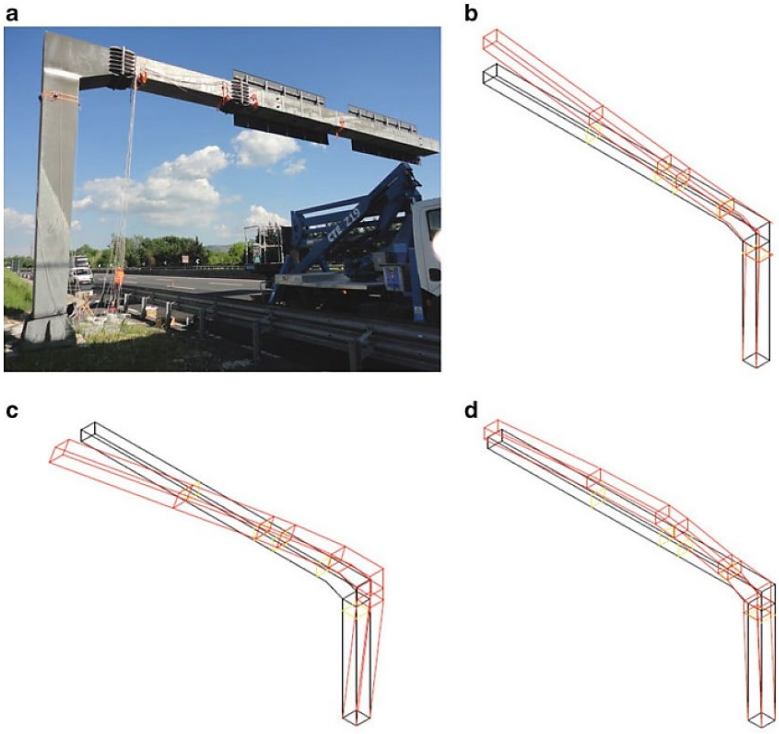


Figure 6. Steel structure for road sign (a) and static display of some of its experimentally estimated mode shapes (b–d) (Rainieri and Fabbrocino, 2014).

The data obtained from the SHM process, suitably pre-processed and validated using the procedures described, represent a fundamental resource for advanced infrastructure management. These data not only enable real-time monitoring and timely detection of potential anomalies or structural damage, but also form the basis for the development of advanced predictive models and the continuous updating of numerical reference models. By integrating the results from dynamic identification algorithms with machine learning techniques or probabilistic analysis, it is possible to predict the future behaviour of the structure with greater reliability and optimise preventive maintenance and risk assessment strategies.

1.2 MODEL UPDATING

Updating the numerical model is one of the most important steps in modern structural management based on dynamic monitoring data. This process consists of calibrating the uncertain parameters of a numerical model based on information obtained from SHM campaigns, such as natural frequencies, modal shapes, damping and any other dynamic parameters observed experimentally. The goal is to obtain a model (the structural digital twin) that corresponds as closely as possible to the real behaviour of the structure, which is a fundamental requirement for reliable predictions and accurate risk analysis.

Model updating requires the analyst to make a preliminary decision on the complexity and fidelity of the numerical model to be used, identifying two main extremes: high-fidelity models and low-fidelity models. High-fidelity models (full-order models, FOMs) are distinguished by the detail with which they represent every aspect of the structure, such as geometry, materials, boundary conditions, and loads, and are able to capture both global behaviour and local, non-linear, or evolutionary phenomena. These models tend to have high computational costs, especially in the iterative phases of updating, but are indispensable when it is necessary to simulate complex effects or study the propagation of localised damage (Svendsen et al., 2021; Lee et al., 2023). Low-fidelity models (such as Reduced-Order Models, ROMs), on the other hand, use coarser meshes, large timesteps and simplify certain physical mechanisms (Fernández-Godino, 2023; Conti et al., 2024). This feature makes them suitable for representing global behaviour, while optimising computational time and resources. They can be formalised using simplified analytical models, in which the structure is schematised as a mass-spring system with a few degrees of freedom (Agathos et al., 2021; Vlachas et al., 2025), or through surrogate models that directly describe the input-output relationship learned from the data (Deshmukh and Allison, 2017; Torzoni et al., 2023). In many practical cases, hybrid strategies are adopted where ROMs are used for initial global calibration, while FOMs are used to investigate particularly critical parts in detail or to assess the propagation of local damage. Furthermore, with the increasing availability

of data and machine learning techniques, surrogate models and data-driven solutions are becoming an increasingly integral part of advanced model updating strategies.

The second fundamental decision that the analyst must make concerns the methodology adopted for calibrating uncertain parameters. This choice is not marginal, as it deeply affects both the final result in terms of accuracy and reliability, and the computational sustainability of the process itself (Ereiz et al., 2022). The techniques that can be used can be divided into three main 'families': (a) deterministic methods, (b) probabilistic methods, (c) data-driven methods, based on ML and DL techniques. Each of these categories offers different solutions depending on the uncertainty in the data, the objective of the analysis (speed, robustness, interpretability), the number of parameters to be updated, and the resources available for simulation.

The selection of the most suitable MU technique is therefore related to the characteristics of the structure to be calibrated, the quantity and quality of experimental data available through SHM, as well as the need to manage the uncertainty resulting from both the physical-mechanical parameters and the modelling itself in an informed manner. The following subchapters will describe these different methodologies in detail, highlighting their operating principles, advantages and limitations, as well as the potential offered by the integration of data-driven techniques and latest-generation artificial intelligence.

1.2.1 Deterministic methods

MU's deterministic approaches are based on a clear and intuitive principle: minimising the deviation between experimental data and data obtained from the numerical model, solving an optimisation-type objective function (Hofmeister et al., 2019). Formally, the vector of optimal parameters $\hat{\theta}$ is defined as:

$$\hat{\theta} = \arg\min_{\theta} F\left(\bar{d}, d(\theta)\right) \tag{4}$$

where d represents the measured data and $d(\theta)$ the output simulated by the model parameterised by θ. The aim is to minimise the residue:

$$res(\theta) = \bar{d} - d(\theta) \tag{5}$$

which quantifies the error between model and reality.

These methods are widely reported in the literature and are often used in contexts where the number of uncertain parameters is limited and well defined, in particular the mechanical properties of structural materials or certain geometric characteristics (Aloisio et al., 2022; Avci et al., 2022; Kurent et al., 2021; Cabboi et al., 2017). The analyst plays a crucial role in appropriately defining engineering assumptions, such as external or internal constraints, thus directing the model towards a solution consistent with the observed reality. In some cases, the process is supported by complementary data, such as mechanical tests on materials taken directly from the structure, which allow the calibration confirmation or restriction, as highlighted in the study by Aloisio et al. (2021) on the elastic modules of concrete in a monitored beam. From a computational point of view, deterministic methods are well adapted to use with high-fidelity models (full-order models), as their calculation structure integrates naturally with iterative optimisation on detailed model parameters, ensuring a high degree of accuracy. Among the established methodological variants are the sensitivity technique (Svendsen et al., 2021; Z. Yuan et al., 2018; Zhu et al., 2017), which performs an iterative search driven by the sensitivity of the outputs with respect to uncertain parameters, and regularisation methods (Titurus and Friswell, 2007), which integrate penalty terms into the objective function to improve the stability and robustness of the solution, especially in the presence of noisy or insufficient data. Recently, innovative methods such as the use of inverse surrogate models based on machine learning algorithms (e.g., random forest regression) have been proposed to improve the efficiency and accuracy of deterministic updates, mitigating typical problems of incorrect conditioning and multiple identification of structural properties (Kamali et al., 2024).

However, one of the main limitations of deterministic methods is in the setting of initial design assumptions. The assumptions made prior to calibration (e.g., on geometry, boundary conditions, or material properties) can heavily influence the final solutions obtained. In practice, even if the residual $res(\theta)$ can be reduced significantly

through optimisation, there is no guarantee that the resulting model accurately represents the actual behaviour of the structure. There is a risk of a calibrated but incorrect model if the basic assumptions are incorrect or overly simplified (Ereiz et al., 2022). Another critical issue is related to the iterative trial-and-error approach, in which the search for a sub-optimal solution can be very complex, especially if the calibration involves several parameters at the same time. This process can lead to convergence on local minima of the objective function, and can also make it difficult to find the right compromise between the different variables involved. In some cases, an exhaustive evaluation of all possible combinations and the use of automated algorithms are employed to solve the problem (offline MU) (Bruno et al., 2024). Finally, the increase in the number of parameters to be updated significantly increases these problems, increasing the bias of the sensitivity matrix and making the entire updating process more sensitive to identification errors and stability issues, especially in the presence of noisy or poorly informative experimental data. For this reason, deterministic methods are more reliable when applied to models with few significant parameters, while complex systems may require alternative methods or methods that combine other techniques to obtain a more accurate solution.

1.2.2 Probabilistic methods

The probabilistic approach to model updating focuses on the explicit management of uncertainty about the parameters that characterise the numerical model of the structure. In this perspective, each uncertain parameter is associated with a probability distribution (e.g., normal or uniform), and the objective becomes to update these distributions according to the experimental data collected through monitoring. From an operational point of view, the probabilistic model updating process is very often formalised in the literature according to the Bayesian paradigm (Lye et al., 2021b; Behmanesh et al., 2015; Rocchetta et al., 2017; Jia et al., 2022), whereby prior knowledge about unknown variables (prior probability) is modified through the so-called likelihood function, obtaining posterior distributions as output:

$$p(\theta|\bar{d},\mathcal{M}) = \frac{p(\bar{d}|\theta,\mathcal{M})\,p(\theta|\mathcal{M})}{p(\bar{d}|\mathcal{M})} \tag{6}$$

where θ is the vector of model parameters, d are the experimental data, M the class of models considered, $p(\theta \mid M)$ the a priori distribution of parameters (Probability Density Function), $p(d \mid \theta, M)$ the likelihood function and $p(d \mid M)$ is normalisation. The latter involves the integration of parameters across the entire space and is particularly important for model selection (Beck and Yuen, 2004).

$$p(\bar{d}|\mathcal{M}) = \int_D p(\bar{d}|\theta)p(\theta)\,d\theta \tag{7}$$

with *D* is the space of the model's numerical parameters. Assuming that the prior probability is independent of the experimental data, it is common to calculate the likelihood function as the convolution of the PDFs for modelling and measurement errors, defining the prior probability with the same distribution as the posterior. In practice, especially with high-fidelity models involving numerous parameters, the calculation of the posterior can become difficult to compute due to the increasing dimensionality of the problem. For this reason, sampling methods such as Monte Carlo Markov Chains (MCMC) are often used (Ierimonti et al., 2020; Beck & Au, 2002; Jang and Smyth, 2017; Monchetti et al., 2023), which allow the solution space to be explored efficiently, although at a high computational cost when the numerical model is very detailed. Other probabilistic approaches include fuzzy logic techniques (Chowdhury and Adhikari, 2012; Haag et al., 2011), principal component analysis (PCA) (Sakiyama et al., 2022), genetic algorithm (Tran-Ngoc et al., 2021) and Kriging strategies (Y. Yuan et al., 2022), which, by expanding the set of tools available, allow for more flexible management of extremely uncertain scenarios or qualitative information. Recently, the integration of machine learning-based methods, such as the use of Artificial Neural Networks or multitask learning models (Bayesian multitask learning), has made it possible to strengthen the ability to manage uncertainty and improve the robustness of updates even on large complex models. The use of surrogate models, combined with Bayesian frameworks, allows for more accurate and computationally sustainable estimates, especially in the

case of intensive and real numerical simulations. The main advantage of probabilistic approaches is the possibility of quantifying the residual uncertainty on the estimated parameters, providing an accurate and robust solution. However, the high mathematical complexity due to the need to deal with probability distributions and solve multidimensional integrals (typical of classical Bayesian inference) often makes these approaches unintuitive and difficult to manage by analysts without a solid background in probability and statistics. Furthermore, updating many parameters simultaneously, which is common in high-fidelity models, multiplies the dimensionality of the problem and can require very high calculations, running times and memory resources, effectively confining the application of probabilistic strategies to offline contexts or case studies with a limited number of parameters.

1.2.3 Data driven and ML-based methods

In recent years, Machine Learning and Artificial Intelligence (AI) techniques have established as powerful tools in the field of structural model updating, thanks to their ability to manage the complexity, uncertainty and volume of data typical of advanced SHM systems. The most popular approach today involves the construction of surrogate models, such as ANNs (Torzoni et al., 2023) or Gaussian Processes (Malekloo et al., 2021), which are trained on large datasets obtained by simulating thousands of scenarios using Finite Element Model FEM (high-fidelity or full-order) numerical models. These surrogate models, once trained, are able to provide quick estimates of the structural state or directly formulate probabilistic predictions of structural parameters, using only dynamic monitoring data as input, without having to solve the entire numerical model each time. This strategy allows for a significant reduction in computational time and costs, which is a fundamental advantage, especially when updating complex and large models. The literature reports cases of great interest, such as the methodology proposed by Y. Lee et al. (2023), which involves generating a data space by simulating multiple damage scenarios on the reference FEM model, followed by training the ANN on the simulated dynamic responses. Other research, such as Zeng et al. (2025), introduce probabilistic parametric estimators that learn directly from the data the mapping between structural response and model parameters, enabling real-time

probabilistic estimates. The integration of data from high-fidelity models, model reduction strategies and ML now allows data-driven algorithms to be implemented effectively even in large-scale problems, supporting the development of intelligent SHM systems capable of diagnosis, prediction and updating in near real time.

However, AI and ML techniques have some crucial limitations that affect their robustness and real applicability in structural engineering. First, their effectiveness critically depends on the availability of training datasets that are large, representative, and physically reliable, often generated through costly and intensive numerical simulations (high-fidelity/FEM models or experimental campaigns). The collection, selection and validation of such data involves a significant computational and organisational effort, which can become a real limiting factor in cases where all possible structural and evolutionary conditions need to be represented in a comprehensive manner. Several studies have shown that the quality, variety and physical reality of the datasets used for training play a significant role in the performance of ML models applied to structural updating (Ghahramani, 2015; Esteghamati et al., 2025). The lack of representativeness or the presence of data that is too similar risks limiting the generalisation capacity and robustness of algorithms, making them less effective outside the domains already seen during training. Recent studies explore the construction, selection and optimisation of datasets (as in Vrtač et al. (2023)), illustrating the importance of advanced data design strategies to ensure optimal coverage of variables of interest. Furthermore, the robustness and interpretability of the results produced by ML/AI models can be significantly affected by unexpected changes in the conditions of the structure or operating environment (e.g., new damage states, climate/exceptional changes, load conditions never simulated). Models not trained on similar scenarios may therefore fail to correctly recognise phenomena outside the domain of the original dataset, with risks of diagnostic or predictive error. Finally, the black-box nature of many deep architectures (deep learning and complex models) can reduce the transparency of decision-making processes, making it more complex for structural engineers to analyse the reasons behind the results and diagnose any anomalies or uncertainties in the predictive models themselves.

2. Application of classical Structural Health Monitoring in a real case-study

This Section aims to explore and analyse in detail the operational procedure of a real structural monitoring campaign, with the aim of understanding its distinctive characteristics, application potential and technical-scientific content. In particular, it analyses the main characteristics of dynamic monitoring in a case study of a complex building in the port of Bari. Through this study, it will provide an overview of the methodologies, tools, and procedures necessary to carry out an effective and reliable SHM investigation. The case study presented concerns the assessment of ordinary and extraordinary vibration effects induced by dredging activities on the surrounding structures in the port of Bari (Ruggieri et al., 2024), representing a significant example of how structural monitoring can be used to assess the impact of unconventional vibration sources on existing structures. This scenario provides an opportunity to examine:

- Data acquisition methodologies through distributed sensor systems.
- Analysis techniques for assessing vibration levels.
- The application of OMA for the recognition of any structural damage.
- Regulatory criteria for assessing human comfort and structural safety.
- Model Updating procedures for the calibration of numerical models.

The Section covers all the technical aspects of the monitoring campaign, from the description of the building to the setup of the data acquisition system, the analysis of the results, and the final thoughts. This structure provides the reader with a practical and comprehensive guide for understanding and possibly replicating similar procedures in similar contexts, highlighting the critical issues, methodological choices and technical solutions adopted. The approach adopted also highlights the potential of structural monitoring as a tool to support engineering decisions. Furthermore, it explains the importance of using monitoring data for the correct calibration of numerical models useful for performing predictive analyses for the assessment of structural integrity and the prevention of potential damage.

2.1 Description of the building

The building selected for structural monitoring represents a significant case study for the application of SHM techniques in complex urban contexts. The structure under analysis is currently the headquarters of ARPA Puglia (Regional Agency for Environmental Prevention and Protection), a regional agency for environmental prevention and protection. It was strategically chosen as a "sentinel building" due to its proximity to the port dredging work area in the city of Bari. The building was constructed in the early decades of the 20th century and for many years housed a railway station serving the port system. From a planimetric point of view, the structure has a rectangular shape with a total surface area of approximately 250 square metres. The distinctive feature of the building is the presence of two separate structures of different heights connected by a glass spiral staircase, which creates a significant difference in height visible in the perspective views (Figure 7). This particular geometric configuration represents an element of complexity both from a structural point of view and in terms of overall dynamic characteristics.



Figure 7. Phots showing rear elevation views of the sentry building.

The structural configuration, on the other hand, is the result of maintenance and consolidation work carried out about twenty years ago, which radically transformed the original load-bearing system. The structure consists of load-bearing masonry and vertical and horizontal reinforced concrete elements within the perimeter walls, creating a

mixed masonry-reinforced concrete structure. For simplicity of description, specific nomenclature is used below for the different structural parts: Body A represents the lowest part of the building with a height of 3.75 m; Body B consists of the upper part, with the first floor at 3.65 m and the second floor at 3.50 m. The two structural bodies are structurally connected by a r.c. spiral staircase, enclosed in a circular glass-aluminium structure that represents an additional element of structural complexity (Figure 8).



Figure 8. Phots showing front of the sentry building and indication of Body A, Body B, and staircase.

A detailed geometric survey was conducted on the building to determine the dimensions of the structural elements, including masonry, reinforced concrete elements, floors and stairs. The results of this survey are particularly significant for understanding the structural behaviour and are reported in Table 1.

Table 1.

Building part	Floor	Structural element	Dimensions (cm) /shape
Body A	Ground floor	Masonry walls	55
		RC columns	40 / circular
		RC walls	25 / rectangular
		Slab	50
Body B	Ground floor	Masonry walls	55
		RC columns	40 / circular
		RC walls	25 / L-shape

Body B	First floor	Slab	50
		Masonry walls	45
		RC columns	40 / circular
		RC walls	25 / L-shape
		Slab	35
Staircase	Ground-First floors	RC columns	20 x 25
		Stair slab	15
		Landing slab	35

An interesting aspect of the building concerns the variety of structural elements in reinforced concrete (shown in Figures 9 and 10). In particular, the columns have different geometric configurations, including circular section columns, which are the most common type and were subject to consolidation work carried out after the building was constructed (Figure 10(a)). L-shaped columns (Figure 10(b)) and rectangular columns to which the spiral staircase is connected (Figure 10(c)).



Figure 9. Internal photos of the building showing structural elements (both RC and masonry) at the ground floor (left) and at the first floor (right).



(a) (b) (c) (d)

Figure 10. Photos of RC structural elements: (a) circular column; (b) L-shape column; (c) rectangular column supporting the helicoidal staircase; (d) external RC wall.

From the point of view of environmental stresses, the building is subject to various sources of vibration under normal operating conditions, which constitute the "environmental noise" caused by ordinary pedestrian and vehicular traffic on the road in front of the building (Figure 8), port traffic on the road behind the building (Figure 7), and the internal machinery necessary for the activities of the regional agency housed there. In addition, the structure is subject to vibrations induced by extraordinary dredging works for port remediation. This work, carried out from May 2022 to August 2022, involved the use of the Willem Van Rubroeck dredger, a latest-generation vessel with a load capacity of 6,900 tonnes, a nominal power of 8,500 kW and a total length of 151.3 metres. The dredging used the "Cutter Suction Dredger" method, which involves breaking up the rock and simultaneously pumping the crushed material directly into the reclamation areas through floating steel pipes with a diameter of approximately one metre. Considering that the dredging was carried out on a compact rock bank, one of the main concerns was the possible damage to the existing buildings surrounding the work area (as shown in Figure 11). This characterisation of environmental conditions is essential for distinguishing the effects of vibrations induced by dredging from the normal baseline

vibration level of the building and for understanding the possible causes of structural dynamic characteristics..

2.2 Monitoring campaign set up

The design and implementation of the structural monitoring campaign required careful planning to ensure the acquisition of reliable data representative of the dynamic behaviour of the building during the dredging period. The experimental setup was developed with two main objectives in mind: i) the assessment of vibration levels in accordance with current regulatory requirements; ii) performing OMA for continuous dynamic characterisation and recognition of any structural damage. For these reasons, the monitoring strategy involved the use of two distinct and complementary monitoring systems, both optimised for specific operational purposes. This methodological choice made it possible to obtain separate information and mutually validate the results, increasing the overall reliability of the experimental campaign.

2.2.1 Sensors

The monitoring system was implemented using two types of accelerometer sensors, characterised by different technical specifications and operating modes.

Intelligent vibrometric monitoring system. The first system consists of an intelligent vibrometric monitoring terminal, specifically designed for the continuous assessment of vibration levels in accordance with Italian normative requirements. The device incorporates an internal triaxial accelerometer characterised by low intrinsic noise and is a fully integrated system that includes sensors, a 3G modem, Wi-Fi connectivity and GPS. The main technical features of the device include seven highly robust and waterproof measurement channels, a storage capacity of over ten thousand readings per day, and a sampling frequency of 200 Hz with acquisition intervals of 5 seconds. The system also incorporates a filtering system consisting of bandpass filters set at 1/3 octave followed by weighting filters. Specifically, two specific filters are applied to isolate the most significant frequency bands for different applications. For the assessment of human comfort, the frequency band between 0.5 Hz and 80 Hz is

considered, while for structural damage, the frequency band is between 1 Hz and 300 Hz. The operating principle of the device is based on the requirements of the relevant Italian standards. UNI 9614:2017 (2017) provides guidance on vibration limits for human comfort, based on ISO 2631-1 (1997) and ISO 2631-2 (2003), while the UNI 9916:2014 establishes vibration limits for structural damage, based on ISO 4866 (2010) and DIN 4150-3 (2016). Data processing is performed in the time domain by calculating the RMS (Root Mean Square) value, defined according to the equation:

$$RMS = \sqrt{\frac{1}{T}\int_0^T x^2(t)dt} \tag{8}$$

where T is the signal duration and $x(t)$ is the amplitude value as a function of time. RMS values are then expressed on a logarithmic scale (decibels, dB) as a function of the ratio between the measured value and a reference value.

Distributed accelerometer system. The second system consists of a set of biaxial and triaxial accelerometers (Figure 11) distributed throughout the building and connected to a control unit for data acquisition via a wired network. The control unit is the central element of the system (Figure 11) and features a network connection via Ethernet port with TCP/IP protocol, 256 MB internal memory and 64 GB SD storage, external USB support for data backup and dual power supply system via standard internal battery with 24-hour autonomy and traditional external power supply. The accelerometers operate according to the force-balance principle, where accelerations are recorded by monitoring the position of an inertial mass positioned with specific degrees of freedom on dedicated suspensions. The control circuit keeps the inertial mass stationary with respect to the reference system, while the recorded acceleration is proportional to that applied to the sensor frame. The maximum sensitivity of the detection accelerograms is 4g. Each device connected to the network has a self-powered channel, allowing all channels to operate simultaneously at a maximum frequency of 1 kHz. A trigger threshold of 0.005 m/s^2 has been set for each channel which, once activated, allows the control unit to send SMS/email notifications to users and start a new recording. The data acquired by the data logger is stored in CSV (ASCII/text) format in the

available memory and then transmitted to an FTP/SFTP server via the established network connection at pre-set time intervals.

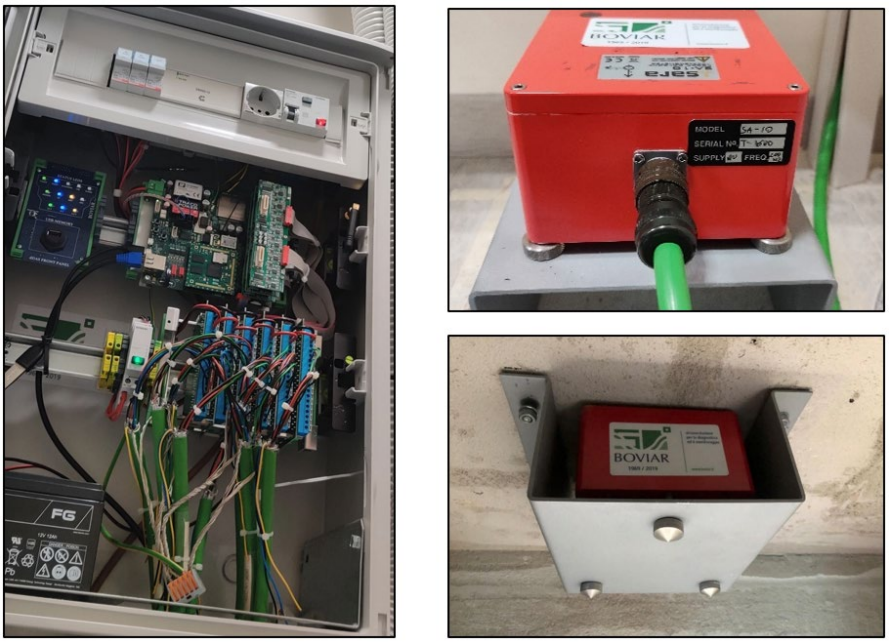


Figure 11. Photos of the data acquisition control unit (left) and of the installed accelerometers (right).

2.2.2 Placement

The strategic positioning of the sensors was designed to maximise monitoring effectiveness, taking into account both the specific objectives of each system and the geometric and structural characteristics of the building.

The vibrometric monitoring terminal was positioned in a special pit built on the roof of Body B (Figure 12), with the primary objective of capturing the most intense vibrations that physically occur in the highest part of the building. This position was chosen due to the need to monitor the point of the structure theoretically subject to maximum dynamic amplification.



Figure 12. Photo of the position of the smart vibration monitoring terminal (left) and photo of the instrument (right).

It is important to note that the optimal position would have been the centre of gravity of the top floor, but the presence of a skylight made it necessary to identify an off-centre position. Therefore, a dedicated support structure (well) was built to house the device safely and protect it from the elements.

The layout of the distributed accelerometers was designed according to scientifically established criteria (Acunzo et al., 2017) and provided for the installation of nine accelerometers distributed across the three levels of the building according to a specific configuration, which allows for the correct identification of the overall dynamic characteristics of the structure. Three triaxial accelerometers were installed on the ground floor, including one external accelerometer positioned outside the building and two internal accelerometers located inside the building to monitor the foundations. On the first floor, four biaxial accelerometers were placed in opposite corners near the structural elements. The four devices were used to compare possible modal differences between Body A and Body B. On the second floor, two biaxial accelerometers were installed exclusively in Body B, mounted on the ceiling using a dedicated fixing method (shown in Figure 11-bottom right). The decision to use triaxial devices on the ground floor was justified by the need to evaluate possible vertical vibration components, while an external accelerometer was used to verify the effects on the building's external constraints. Four devices were used on the first floor to compare possible modal differences

between the two structural parts, considering a probable internal constraint on the first floor. Figure 13 shows the configuration of the acquisition unit and the various sensors, differentiated for the individual floors of the structure.

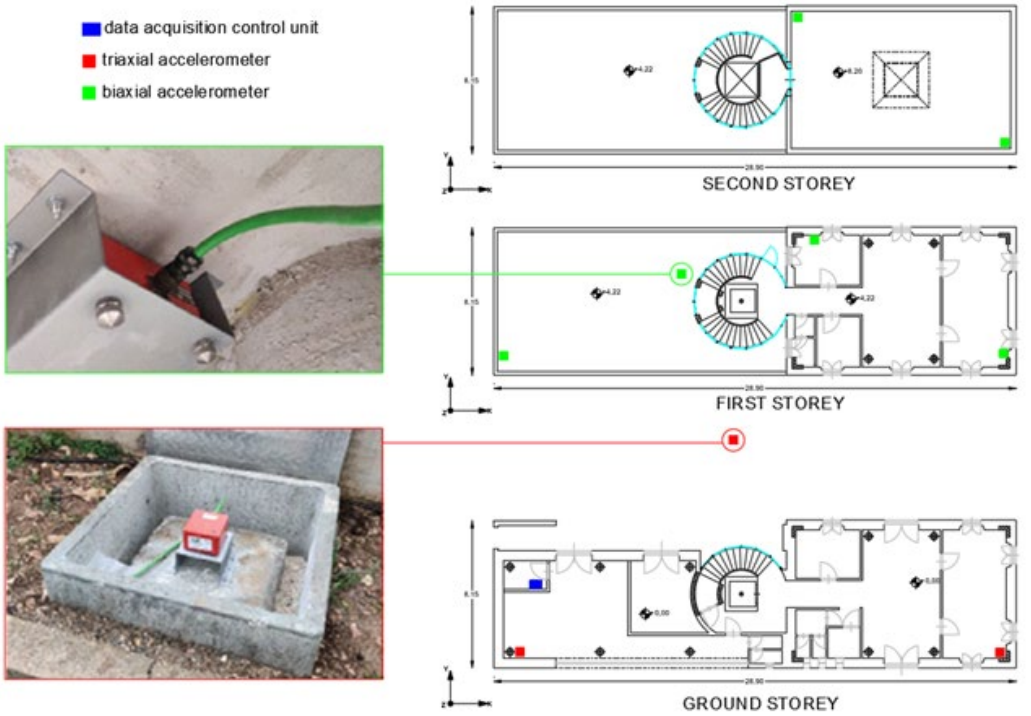


Figure 13. Disposition of all accelerometers and of the data acquisition control unit throughout the building, about a biaxial accelerometer at the first storey and placed in the ceiling and photos about the triaxial accelerometer at the ground storey outside the building (placed in a specific manhole).

The overall configuration involved setting up a total of twenty-one acquisition channels, considering the number of accelerometers and the presence of biaxial and triaxial devices.

2.2.3 Measurements

The data acquisition protocol was designed to ensure continuous monitoring throughout the period of interest, while optimising the use of available storage and transmission resources.

The intelligent vibrometric system operated in continuous monitoring mode for the entire duration of the campaign, with automatic data acquisition and processing according to precise specifications. The system maintained continuous acquisition 24 hours a day throughout the period of interest, with automatic processing according to filters for human comfort and structural damage. An automatic comparison with Italian normative thresholds was also implemented, with the storage of over ten thousand daily readings and real-time data transmission for remote monitoring.

For the distributed accelerometer system, an intermittent acquisition protocol optimised for OMA requirements was adopted. Recordings last 10 minutes every half hour with a sampling frequency of 200 Hz and sampling intervals of 0.05 seconds. Each session generates a data volume of approximately twelve thousand acceleration readings expressed in m/s^2, stored in CSV format for compatibility with analysis software. Automatic transmission involves sending the data to the server at the end of each acquisition session.

The monitoring campaign was structured to cover three distinct periods: a pre-dredging period to establish the building's reference vibration conditions, the dredging period for monitoring during extraordinary activities (May-August 2022), and a post-dredging period to verify the return to normal conditions. This methodological choice made it possible to assess not only whether the regulatory limits had been exceeded, but also whether the dredging activity had actually increased the normal operating vibration levels of the building. It is important to specify that the first monitoring system was used exclusively to verify whether the vibration thresholds set by current Italian rules for human comfort and structural damage had been exceeded, while the second system was adopted for the purposes of Operational Modal Analysis and Model Updating, to assess the possible onset of structural damage through the identification of variations in modal parameters.

2.3 Vibration levels analysis

The vibration level analysis was conducted by comparing the data measured during the three monitoring periods, i.e. pre-dredging, during dredging and post-dredging, with the Italian regulatory vibration limits defined for human comfort and structural damage. Continuous monitoring by the intelligent vibrometric system returned RMS acceleration and velocity values expressed on a logarithmic scale (dB), calculated according to Equation 8 and using bandpass filters dedicated to the frequency bands for human comfort (0.5-80 Hz) and structural damage (1-300 Hz). Tables 2 and 3 show the vibration limits set by the regulations. For human comfort, the UNI 9614 norm (2017), establishes two different values for planar vibrations (in X and Y) and jolting vibrations (in the Z direction), for the different intended uses of buildings. With regard to the vibration limit levels for structural damage, the UNI 9916 (2014), establishes two different speed limits depending on the duration of the action itself (i.e. short term and permanent). Considering short-term actions that do not induce resonance or structural fatigue phenomena.

Table 2. Vibration thresholds in acceleration for human comfort, expressed in mm/s^2, according to UNI 9614-17 (2017). Values are reported for different use destination and different period of the day

Use destination	Period of the day	Limits for human comfort, X and Y directions (mm/s^2)	Limits for human comfort, Z direction (mm/s^2)
Residential	Daytime period	3.6	10
Residential	Night period	5	7
Offices	Always	14.4	20
Industries	Always	28.8	40
Critical Area	Always	3.6	5

Considering a destination of use for offices and industrial buildings, the vibration limit value for horizontal actions is 83 dB and 86 dB in the vertical direction for human comfort and 105 dB for structural damage (in all directions).

Table 3. Vibration thresholds in velocity for building damages (at the last storey), expressed in mm/s, according to UNI 9916 (2014) and DIN 4150-3 (2016). Values are reported for different use destination and for short-term and permanent duration of the vibrations.

Use destination	Short-term (mm/s)				Permanent (mm/s)
	1 – 10 Hz	10 – 50 Hz	50 – 100 Hz	All freq.	All frequencies
Residential	5	5 – 15	15 – 20	15	5
Industrial / Commercial	20	20 – 40	40 – 50	40	10
Historical / Sensible	3	3 – 8	8 – 10	8	2.5

Figures 14 and 15 show the vibration levels divided into three time bands during the day (0 to 7 a.m., 7 a.m. to 10 p.m. and 10 p.m. to 12 p.m.) and in the three monitoring phases (pre-dredging reference time, during dredging and post-dredging), for the two horizontal directions (i.e. X and Y). During the pre-dredging period, the average daily vibration levels remained significantly below the alarm thresholds for the "office/industrial-commercial" use class, showing a background intensity of between 40 dB and 55 dB in the three spatial directions. When dredging activities began, there was an increase in vibrations with maximum daily peaks exceeding even 10 dB compared to the reference period; however, these values never exceeded the limits established for structural damage or human comfort (see Figure 14 and Figure 15). In the months following the end of dredging, the levels gradually returned to their initial background values, confirming the temporary nature of the increase in vibration induced by the works. The directional analysis showed that the horizontal component along the transverse Y axis showed the greatest relative increase, due to the preferential propagation of vibration waves along the masonry exposed to the side of the excavation area. The X and Z components, on the other hand, maintained variations of less than 6 dB compared to the pre-dredging period, indicating a homogeneous dynamic response in the building. Furthermore, Figure 16 shows the peak vibration values obtained during the dredging works (from 25/05/2022 to 08/08/2022) in the X, Y and Z directions. The maximum vibration values always remained below the regulatory limits, both for human comfort and structural damage.

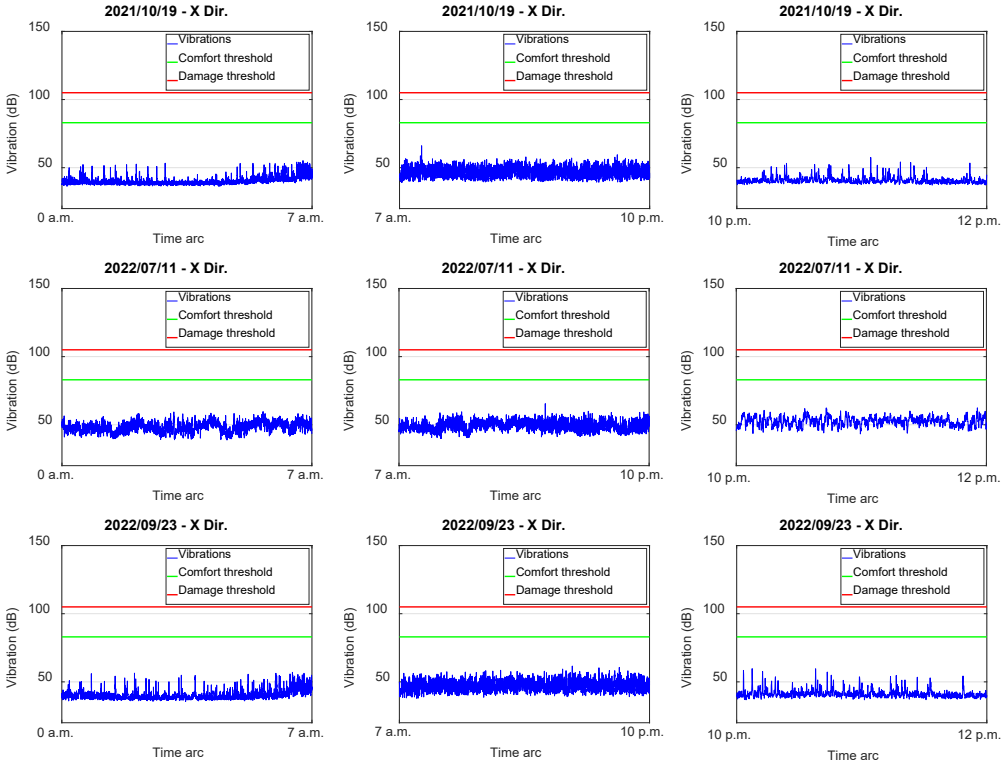


Figure 14. Comparison in terms of vibration (logarithmic scale) among recorded vibrations at the last storey of sentry building and threshold values for human comfort and building damages, X direction. In the Figure are reported the recordings before the dredging (2021/10/19), during the dredging (2022/07/11) and after the dredging (2022/09/23).

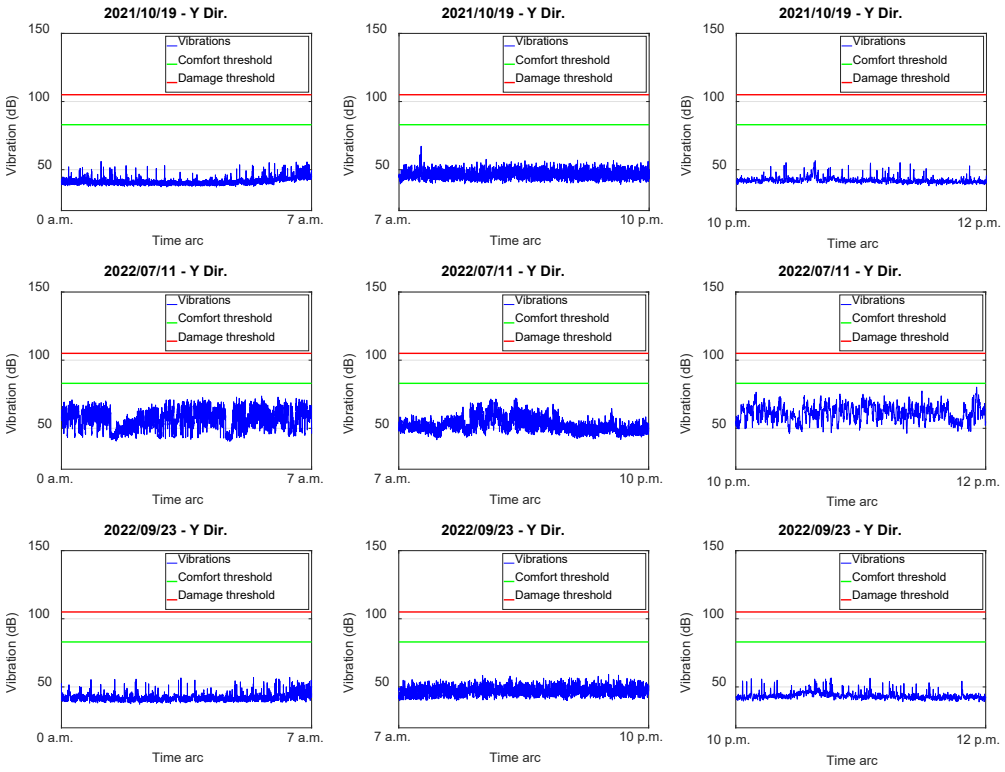


Figure 15. Comparison in terms of vibration (logarithmic scale) among recorded vibrations at the last storey of sentry building and threshold values for human comfort and building damages, Y direction. In the Figure are reported the recordings before the dredging (2021/10/19), during the dredging (2022/07/11) and after the dredging (2022/09/23).

The results, presented in histograms of maximum daily values for each spatial direction (Figure 16), confirm that the dredging activities, although potentially generating significant vibrations, did not compromise either the comfort of the occupants or the structural integrity of the sentinel building. This evidence supports the validity of the monitoring technique used and the correct application of regulatory protocols for the management of environmental vibrations in complex construction sites.

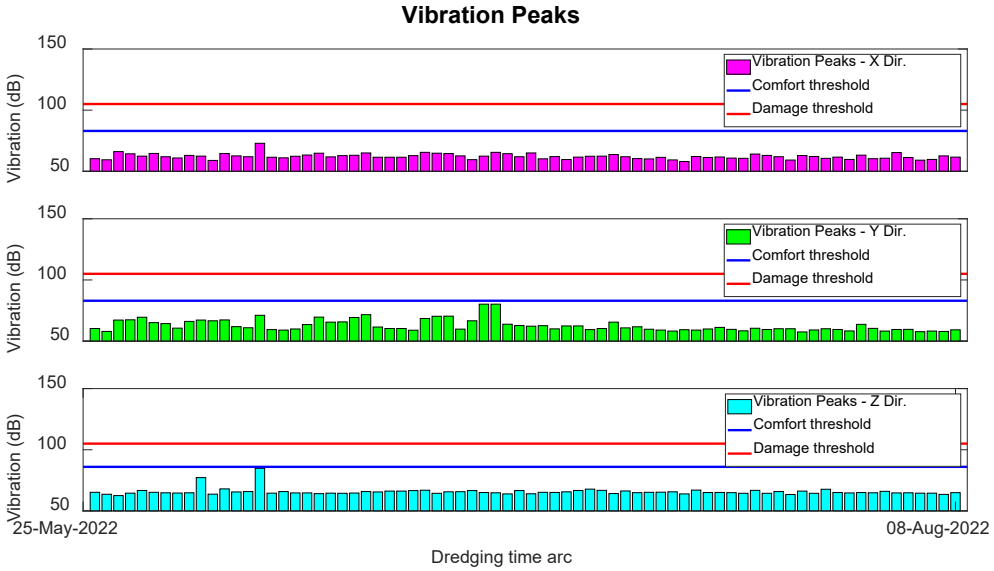


Figure 16. Maximum daily values of vibration (i.e., peaks), accounting for all main directions (i.e., X, Y, and Z) and compared with the thresholds for human comfort and building damages.

2.4 Operational Modal Analysis

With the second monitoring system installed on the structure, OMA was carried out to verify the possible onset of structural damage due to extraordinary dredging activities, analysing the dynamic characteristics (i.e. modal frequencies) in the period before, during and after the works. The experimental methodology was based on the acquisition of environmental data provided by distributed accelerometers, suitably filtered to exclude high-frequency and low-frequency components using a bandpass filter between 1 Hz and 40 Hz. In addition, a detrend filter was applied to clean the noise from the signal. The pre-processed data were processed using ARTeMIS Modal software (Structural Vibration Solutions, 2023), which enabled the reconstruction of the building's geometry, the modelling of boundary conditions and the management of twenty-one measurement channels distributed between the ground floor, first floor and second floor. The FDD algorithm was used for data processing, applied in the frequency domain to select the prevailing vibration modes in the most robust way. The pre-dredging analysis highlighted five main vibration modes, selected through peak picking in the SVD graph (Figure 17). The first mode is translational in the Y direction (weak axis), the

second is translational in the X direction (strong axis), the third is rotational, while the fourth and fifth are translational with inversion in the Y and X directions respectively (shown through the static graphical representation of the ARTeMIS Modal software (Structural Vibration Solutions, 2023) in Figure 18).

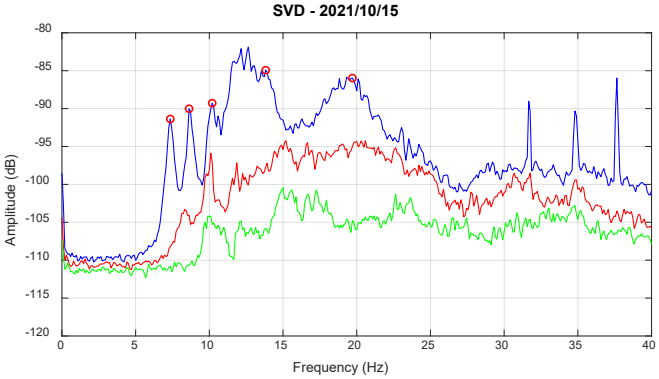


Figure 17. SVD graph using FDD algorithm, by elaborating acceleration data at 2021/10/15 for the sentry building. Red circles indicate the estimated modes of vibration.

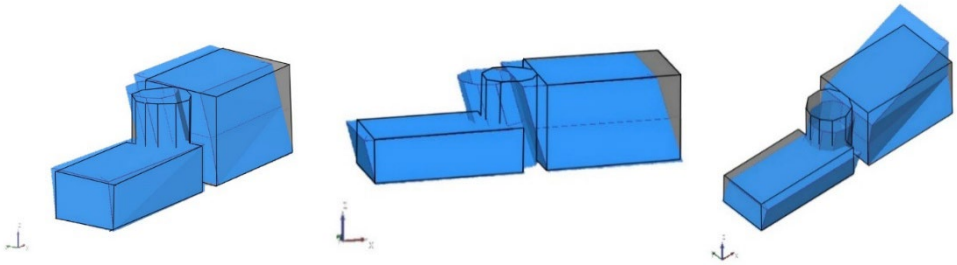


Figure 18. Graphical rapresentation of modal shapes of the first three modes of vibration for the sentry building

The natural frequencies range between 7.3 Hz and 19.2 Hz, with damping values varying from 2.4% to 6.6% (as shown in Table 4 for the day of 15/10/21, pre-dredging period). To validate the selected modal shapes, the classic techniques described in Chapter 1.1.4 were used, namely the visualisation of the complexity plots and the value shown in Table 4, the MAC trend to visualise the degree of correlation between the identified modes, and the dynamic visualisation of the modal shapes. These confirm the robustness of the modes identified as characteristic of the structure.

Table 4. Frequencies, damping, and complexity for the first five vibration modes of the sentry building, using environmental excitation of 2021/10/15.

Mode	Frequency (Hz)	Damping (%)	Complexity (%)
1	7.324	2.434	0.234
2	8.594	5.545	0.262
3	10.156	6.587	2.621
4	14.258	6.589	3.613
5	19.238	5.214	1.373

During dredging, continuous monitoring of modal parameters was carried out on a daily basis in order to detect any changes in dynamic characteristics indicative of possible structural damage. The results show that the variations observed on the first three frequencies identified are limited to 2%, 2.5% and 8% respectively (Figure 19). These deviations are fully compatible with experimental uncertainty and operational variability related to work and environmental conditions. In no case were variations above the thresholds for damage diagnosis detected.

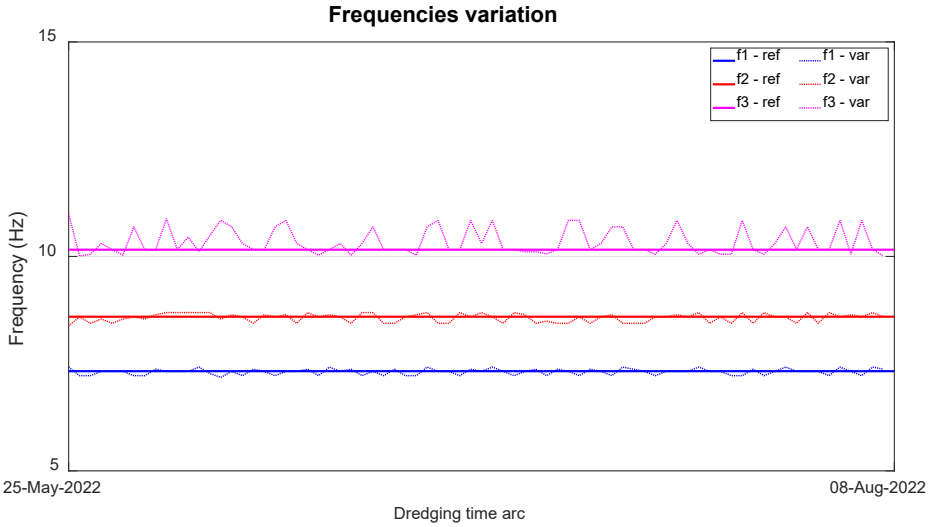


Figure 19. Variation (indicated as “var”) of the main first three frequencies (indicated as “f1”, “f2”, and “f3”) during the dredging period, with respect to the output in the reference day (indicated as “ref”).

In conclusion, the OMA demonstrated that the dredging activities did not cause any significant structural alterations to the building, confirming the effectiveness of the monitoring and the methodological robustness of the procedure adopted. The stability of the frequencies identified throughout the period of interest provides a transparent guarantee of the building's health, in line with best SHM practices in the civil engineering field.

2.5 Model Updating phase

Finally, the results of the SHM campaign were used to calibrate a FEM. This made it possible to overcome uncertainties regarding the geometric and mechanical structural parameters that cannot be determined directly through non-invasive experimental tests. The reason for calibration lies in the structural complexity of the building, which is characterised by a mixed structure consisting of masonry and reinforced concrete elements; different types of constraints between the structural bodies (Body A, Body B and Staircase) and a geometry with significant irregularities in elevation. The inability to carry out in-depth investigations on site (destructive and non-destructive tests) means that these structural characteristics are subject to a high degree of uncertainty. Through MU, it was possible to bridge the gap in knowledge of these parameters. For the building in question, it was decided to perform a calibration using deterministic MU techniques (Section 1.2.1), manually comparing the modal frequencies (and their shapes) obtained through OMA and those obtained numerically from FEM. The model was created in SAP2000 structural analysis software (Computer and Structures, Inc, 2024), respecting the characteristics of the geometric survey and maintaining an adequate level of detail with the overall dynamic characteristics obtained through the monitoring campaign (see Figure 20). The perimeter masonry was modelled using two-dimensional shell elements, while the reinforced concrete elements were represented with beam elements for columns and slender walls and with shell elements for larger walls and diaphragms; the floors were mainly modelled as horizontal diaphragms. The spiral staircase was idealised as a system composed of curved beams and plates of variable thickness, with parametric connections to Body A and Body B to simulate configurations ranging from more rigid to more flexible, consistent with the uncertainties

regarding the actual constraints. The structure is considered to be fixed at the base. For calibration, three parameters were varied according to structural uncertainty, namely: (i) the equivalent elastic modulus of the masonry, E_m; (ii) the elastic modulus of concrete E_c for reinforced concrete pillars/partitions/floors; (iii) the internal constraint between the different structural parts (i.e. connection between Body A, Body B and staircase).

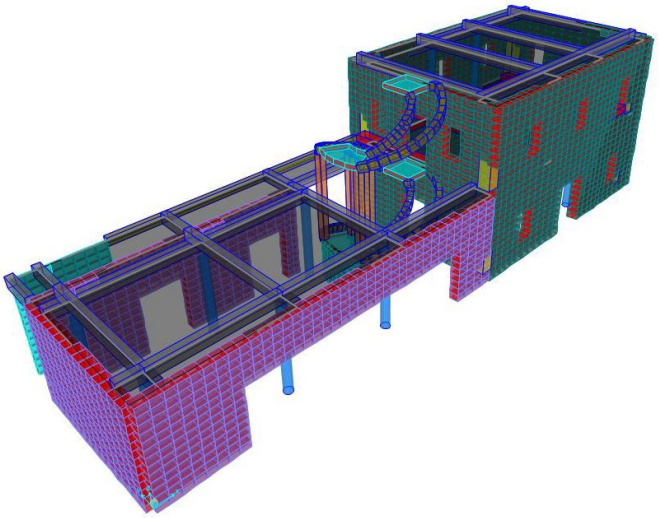


Figure 20. FE model of the building.

To limit the possible combinations of parameters and in accordance with current italian technical code for building materials (MIT, 2018), it was decided to set engineering ranges. For the elastic modulus of masonry E_m considering a lower limit of 1200MPa and an upper limit of 1620 MPa, in accordance with the regulations for the type of masonry present in the structure; and for concrete, varying E_c in the range between 29,900 MPa and 33,050 MPa. For internal constraint conditions, setting two possible scenarios depending on the constraint ratio between the first deck of the two factory buildings, namely:

- Model with a single diaphragm in the foreground, defined as a Constrained model, “C”.
- Model with two diaphragms for the first deck, separated for Body A and Body B, defined as an Unconstrained Model, “U”.

And by setting three different scenarios for the degree of constraint of the stairwell to the two buildings:

- Staircase constrained to Body A (Test 1, “T1”).
- Staircase constrained to Body B (Test 2, “T2”).
- Staircase constrained at the first storey to Body A and at the second storey to Body B (Test 3, “T3”).

For all possible scenarios, the values of the two elastic modules were calibrated manually, obtaining the configuration that minimises the difference between the frequencies obtained from the OMA and the modal frequencies obtained from the model. Tables 5 and 6 show the values of the first three frequencies for each individual scenario and the percentage differences with the results obtained experimentally.

Table 5. Comparison in percentage terms among experimental and numerical outputs for constrained models (best solution by varying Em and Ec), first three modes of vibration, using as reference the environmental excitation of 2021/10/15.

Mode	Experimental Frequencies (Hz)	C-T1		C-T2		C-T3	
		Frequency (Hz)	Percentage difference (%)	Frequency (Hz)	Percentage difference (%)	Frequency (Hz)	Percentage difference (%)
1	7.324	5.19	-29.14	6.05	-17.39	5.20	-29.00
2	8.594	7.32	-14.82	8.54	-0.65	7.32	-14.82
3	10.156	9.78	-3.70	11.03	8.60	9.78	-3.70

Table 6. Comparison in percentage terms among experimental and numerical outputs for unconstrained models (best solution by varying Em and Ec), first three modes of vibration, using as reference the environmental excitation of 2021/10/15.

Mode	Experimental Frequencies (Hz)	U-T1		U-T2		U-T3	
		Frequency (Hz)	Percentage difference (%)	Frequency (Hz)	Percentage difference (%)	Frequency (Hz)	Frequency (Hz)
1	7.324	6.97	-4.83	6.95	1	7.324	6.97
2	8.594	8.94	4.02	8.75	2	8.594	8.94
3	10.156	11.90	17.17	11.70	3	10.156	11.90

In particular, the results identified as optimal the configuration with independent diaphragms between Body A and Body B and a staircase connected to both bodies according to a flexible model ('U-T3'), with Em equal to 1200 MPa and Ec equal to 30550 MPa, obtaining deviations always less than 10% for the main frequencies. All constrained models (C-T1, C-T2, C-T3) were discarded because they had a percentage error equal to 30% and 15% for the first two modal frequencies. Figure 21 reports the modal shape of the first three modes of vibration given by the calibrated numerical model, showing a similar behaviour to those shown in Figure 18.

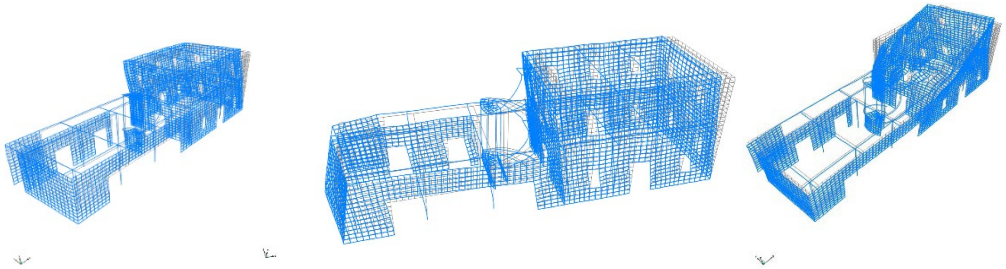


Figure 21. Modal shapes of the first three modes of vibration for the calibrated FE model of the building.

Finally, to validate the calibrated model obtained and verify possible damage during the dredging period, the percentage differences between the modal frequencies were analysed on a day-by-day basis. Hence, Figure 22 presents the percentage differences for the first three modes of vibration. As observed, the maximum percentage differences amounted to -2.28%, 3.12%, 9.41% for the first three modes of vibration, respectively, while the minimum percentage differences amounted to -5.70%, -0.46%, and 0.58%, for the first three modes of vibration, respectively. Although there is natural uncertainty in the OMA outputs, the proposed calibrated numerical model offers a sub-optimal solution for further investigations.

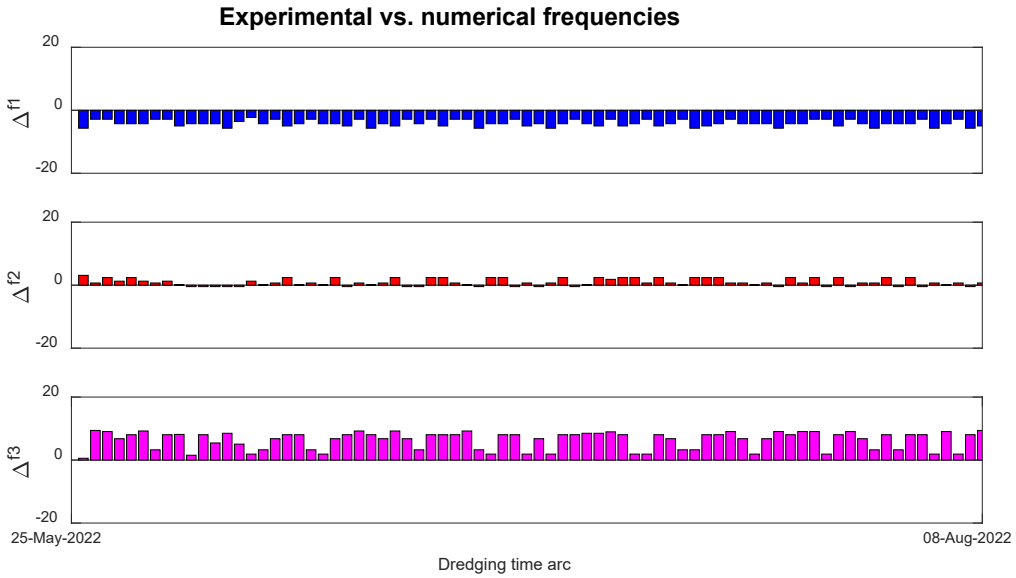


Figure 22. Percentage differences among experimental and numerical frequencies for the main first three frequencies (indicated as "Δf1", "Δf2", and "Δf3") during the dredging period.

2.6 Discussion

The results of the monitoring campaign carried out on the building during the dredging activities in the port of Bari offer extremely relevant food for thought, both in terms of application and methodology. This is consistent with what has been described in the previous sections regarding the importance of integrated SHM systems in structural risk management. The experience analysed demonstrates that monitoring is a useful tool not only for real-time assessment of environmental effects, but also for the calibration and validation of high-fidelity numerical models.

Firstly, the systematic comparison of the vibration levels recorded at different times of the day before, during and after dredging, with respect to the current regulatory limits (UNI 9614 for human comfort and UNI 9916 for damage to buildings), showed that the regulatory thresholds were never exceeded. This data takes on further significance when reinforced by the OMA, as the daily results confirmed the invariance of the fundamental modal parameters (frequencies, modal shapes, damping), leading to the exclusion of structurally damaging effects on the general condition of the monitored building. The validity of the experimental assessment is fundamental in the subsequent

MU phase, in which the FE model was constantly calibrated on the dynamic data collected. The uncertainty regarding the internal constraint conditions between the structural bodies and the staircase, typical of buildings with high geometric and constructional irregularity, was addressed by exploring alternative scenarios, allowing the configuration most representative of the actual behaviour to be identified. This experience, in addition to confirming the robustness of SHM techniques for emergency management or extraordinary actions (such as vibrations induced by dredging), highlights two fundamental implications for engineering research:

- Structural monitoring should not be viewed only as a tool for alerts or ex-post diagnostics, but as a key element in the construction and validation of high-fidelity predictive models. Only by collecting high-resolution data relating to actual operating conditions is it possible to identify parameters that cannot be acquired through traditional investigations, calibrate model assumptions and reduce uncertainty in predictive simulations.
- The integration of monitoring and modelling enables an appropriate risk and safety management strategy. The method adopted for the continuous monitoring campaign has shown that even in the presence of events characterised by high uncertainty (precise location and power of the source, propagation paths, geometric eccentricity), it is possible to prevent critical conditions by relying on models that are constantly verified and updated on the basis of the actual evolution of stresses.

However, the traditional MU approach analysed in this Section also highlights intrinsic limitations that restrict its applicability and efficiency in more complex contexts or in the presence of large structural systems. The procedure adopted is based on iterative manual or semi-automatic search strategies, which require the a priori definition of discrete scenarios and significant engineering experience to guide convergence towards the optimal solution. This approach is computationally expensive when the number of parameters that need to be calibrated increases and when it is necessary to consider multiple alternative structural configurations simultaneously. Furthermore, the convergence of the procedure is not guaranteed in terms of solution uniqueness, and

the risk of local minima in the objective function can lead to calibrated parameters that are not physically meaningful or representative of actual structural conditions.

A further critical issue appears in the management of parametric uncertainty and in the model's ability to dynamically adapt to operating conditions that vary over time. Although the updated model was validated on a daily basis during the dredging period, extending this strategy to long-term scenarios, with progressive structural evolutions or more complex sources of uncertainty (e.g., material deterioration, changes in boundary conditions, complex interactions with non-linear systems), would require continuous manual revision of the model, with costs and timeframes that are unsustainable in large-scale operational applications.

These limitations highlight the need to explore advanced and automated methodologies for MU, capable of managing multidimensional parametric spaces, learning autonomously from observed structural dynamics, and ensuring robust convergence and predictive reliability even in the presence of high complexity and uncertainty. This removes the need for stringent initial assumptions about structural characteristics, simplifying the entire process. In this perspective, the integration of artificial intelligence techniques and classical calibration approaches represents a promising frontier for SHM data management.

3. Model Updating with use of Deep Reinforcement Learning

As highlighted in the previous Sections, MU still represents a critical phase in structural engineering, as it consists of calibrating a numerical model based on SHM data in order to accurately reproduce actual structural behaviour. In particular, Section 2 analysed a real monitoring campaign and examined the main aspects, characteristics and advantages of correct MU of a high-fidelity model of the real structure. This context motivates the development of advanced MU techniques capable of combining the accuracy of probabilistic estimates with the automation and efficiency of deterministic methods. DRL is proposed as an ideal tool, consisting of an agent immersed in a fully accessible simulation environment that intelligently explores possible combinations of values for uncertain parameters, guided by a reward function, learning to quickly converge towards the configuration that best reproduces the output data from the SHM (Bruno et al., 2025). The framework, described in detail in this Section and called DRUM-Av1.0 (Deep Reinforcement Learning for Updating Model Approach version 1.0), represents a new paradigm for the calibration process of numerical models with multiple uncertain parameters and a high degree of fidelity and description of reality (FOM). The following sections of this chapter describe the theoretical basis of DRL techniques (Section 3.1); the architecture of the framework with its five key steps (Section 3.2); its application to a fictitious case study, with a comparison with more traditional MU techniques (Section 3.3); and finally, an application to a real case study (Section 3.4), useful for highlighting the potential and current limitations that led to the subsequent development of the thesis work (Section 3.5). This work demonstrates how DRUM-A represents a significant step forward in MU techniques, enabling online, automated and scalable updates as structural complexity increases.

3.1 DRL basics

Deep Reinforcement Learning is a branch of machine learning that exploits a combination of Reinforcement Learning (RL) and Deep Learning. The idea of RL consists of enabling an artificial agent to take decisions through a *trial-and-error* approach, interacting with a specific environment. The role of the artificial agent is to use the previous experience to optimize an objective function, expressed in terms of a cumulative reward. During its training phase, the agent aims to learn the best behaviour for the problem investigated, by gradually improving its performance. In addition, the agent is not constrained to have a complete knowledge of the environment but only needs to interact with the environment and collect information for improving its ability (François-Lavet et al., 2018). The role of DL is to improve the capacity of the agent when operating in complex environments with high-dimensional inputs. Specifically, DL enhances the generalization capability of value and policy functions in large and continuous spaces for maximizing the reward over time (as later specified).

Focusing on the RL part (DL part is not clearly disseminated here, but its role in DRL can be found in (François-Lavet et al., 2018) and is mentioned later), it can be defined as a discrete-time stochastic control process in which the agent interacts with the environment in a discrete Markov decision problem. According to the above definition, the future of the process depends only on the current state and is not related to the values of the entire set of past states. The discrete Markov problem is defined by 3-tuples, that is, (i) **S**, the state space of the environment; (ii) **A**, the set of actions available to the agent; (iii) R, the reward function. The environment, and its related state space **S**, represents the space in which the agent interacts, by applying actions and observing states (i.e., solution). Depending on the problem, **S** can be discrete or continuous: smaller and simpler environment can be schematized with finite and countable states (discrete), while high-dimensional problems require infinite and uncountable states (continuous). The action space, **A**, characterizes the set of available actions that the agent can perform in the interaction with the environment. Also in this case, depending on the problem, **A** can be discrete or continuous. At each interaction between the agent and the environment, the actions represent an output for the agent and, at

same time, an input for the environment, which is updated continuously. For each interaction, the agent receives a reward, which is quantitative feedback from the environment, based on the errors made by the agent after performing the actions. The reward is a scalar value ruled by the reward function, *R*, which encodes the desired behaviour of the agent and then, represents a pivotal component of the process because it drives the learning phase. Along with the reward, for each step, the agent observes information about the evolution of the state of the environment, called observations, which belong to the observation space $\boldsymbol{\Omega}$. In the case of fully observable environments, $\boldsymbol{\Omega}$ and $\boldsymbol{S}$ coincide.

The interaction between each component characterizing the RL framework can be described starting from the initial timestep, t_0. The environment is in the state $s_{(t0)} \in \mathbf{S}$, for which the agent obtains the observation $\omega_{(t0)} \in \boldsymbol{\Omega}$. In the subsequent operations, at the time generically indicated as $t = t_{0+i}$ (for *i* ranging from 1 to the total number of timesteps), the agent performs actions $a_{(t)} \in \mathbf{A}$ to obtain a reward $r_{(t)} \in R$. As shown in Figure 23, the algorithm performs iterative steps for each value of t, as following reported (Sutton and Barto, 2018):

- The environment is in the environment state $s_{(t)} \in \mathbf{S}$.
- The agent collects the observation $\omega_{(t)} \in \boldsymbol{\Omega}$ related to the state $s_{(t)} \in \mathbf{S}$ of the environment.
- From the set of possible actions, the agent selects and performs an action $a_{(t)} \in \mathbf{A}$.
- According to the performed actions, the agent obtains a reward $r_{(t)} \in R$.

Once agent obtains $r_{(t)}$, the environment state is updated in a new form, i.e., $s_{(t+1)} \in \mathbf{S}$ and in turn, the agent can repeat the process at the next timestep (t+1), by collecting the new observation $\omega_{(t+1)} \in \boldsymbol{\Omega}$ (in fully observable environment, $\omega_{(t+1)} = s_{(t+1)}$) and by performing the new action, $a_{(t+1)} \in \mathbf{A}$. According to the iterative process described, the agent aims to maximises the cumulative reward, expressed by the Q-value function $Q(s, a)$, by finding a policy $\pi(s, a) \in \boldsymbol{\Pi}$, with $\boldsymbol{\Pi}$ indicating the set of all possible policies. Different formulations of the cumulative reward collected by the agent according to a policy $\pi(s, a)$ exist, e.g., the V-value function (or State-Value

function) reported in Equation (9), and the Q-value function (or Action-value function) reported in Equation (10)

$$V^{\pi}(s) = E\left[\sum_{i=1}^{\infty} \gamma^{i}\, r_{(t_0+i)} \middle| s_t = s, \pi\right] \tag{9}$$

$$Q^{\pi}(s,a) = E\left[\sum_{i=1}^{\infty} \gamma^{i}\, r_{(t_0+i)} \middle| s_t = s, a_t = a, \pi\right] \tag{10}$$

where $E[\cdot]$ represents the expectation, $\gamma \in [0,1)$ is the discount factor (which emphasizes the importance of the nearest rewards over future rewards in the further iterations). Optimal V-value function $V^{*}(s)$ and Q-Value function $Q^{*}(s,a)$ are usually defined by maximizing the V-value function and the Q-value function, respectively, over all the possible policies $\pi(s,a) \in \mathbf{\Pi}$:

$$\mathrm{V}^{*}(\mathrm{s}) = \max_{\pi \in \Pi} V^{\pi}(s) \tag{11}$$

$$\mathrm{Q}^{*}(\mathrm{s},\mathrm{a}) = \max_{\pi \in \Pi} Q^{\pi}(s,a) \tag{12}$$

According to the main aim of the agent, which needs to learn a policy that maximizes the cumulative reward, the optimal policy is defined as $\pi^{*}(s,a) \in \mathbf{\Pi}$. In the case of V-Value function, the optimal $V^{*}\left(s = s_{(t)}\right)$ is defined as the expected discounted reward for a given state *s* while following the optimal policy π^{*}. Instead, in the case of Q-Value function, the optimal Q-value $Q^{*}\left(s = s_{(t)}, a = a_{(t)}\right)$ is defined as the expected discount return of the agent performing the action $a = a_{(t)}$ in the state $s = s_{(t)}$ while following the optimal policy π^{*}. For the latter, the knowledge of optimal Q-Value function $Q^{*}(s,a)$ in Equation (10) allows to obtain the optimal policy $\pi^{*}(s,a)$ directly from $Q^{*}(s,a)$ (François-Lavet et al., 2018):

$$\pi^{*}(\mathrm{s}) = \arg\max_{a}\left(\mathrm{Q}^{*}(\mathrm{s},\mathrm{a})\right) \tag{13}$$

Generally, RL algorithms can be classified in two main categories depending on the modality to derive the policy, i.e., value-based and policy-based algorithms. Value-based algorithms are not defined to learn explicitly a policy, but for defining it from the

value function. Policy-based algorithms define the policy by working on explicit representations. Among policy-based methods, the policy gradient methods are widely used, in which the algorithm optimizes an objective function (such as the expected cumulative reward) by finding a good policy (approximated with neural networks) via stochastic gradient ascent. To this scope, let the parametrized policy $\pi_{\psi}(a, s)$ according to ψ, an objective function $J(\Psi)$ is defined to estimate the expected return of actions and update the policy:

$$J(\Psi) = E_{\pi\Psi}\left[\sum_{i=1}^{\infty} \gamma^{i} r_{(t_0+i)}\right] \tag{14}$$

where $E\pi_{\Psi}[\cdot]$ represents the expectation with respect to $\pi_{\Psi}(a, s)$. This latter is usually the employed approach for DRL, which allow continuous updates during the interaction of the agent with the environment.

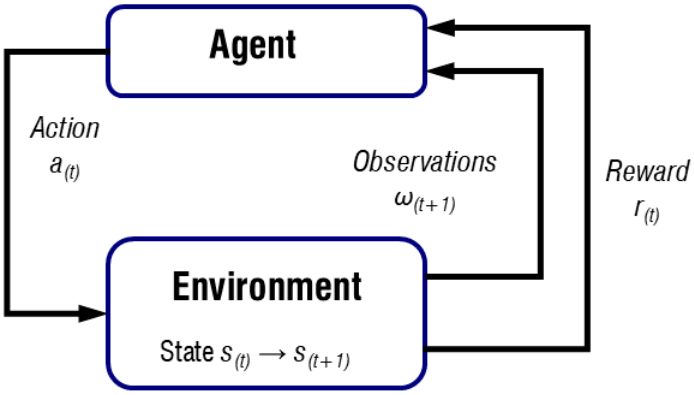


Figure 23. Interaction between Agent and Environment in DRL

3.2 DRUM-Av1.0

This Section presents DRUM-Av1.0, describing a detailed overview of each step characterizing the framework, interacting as in Figure 24.

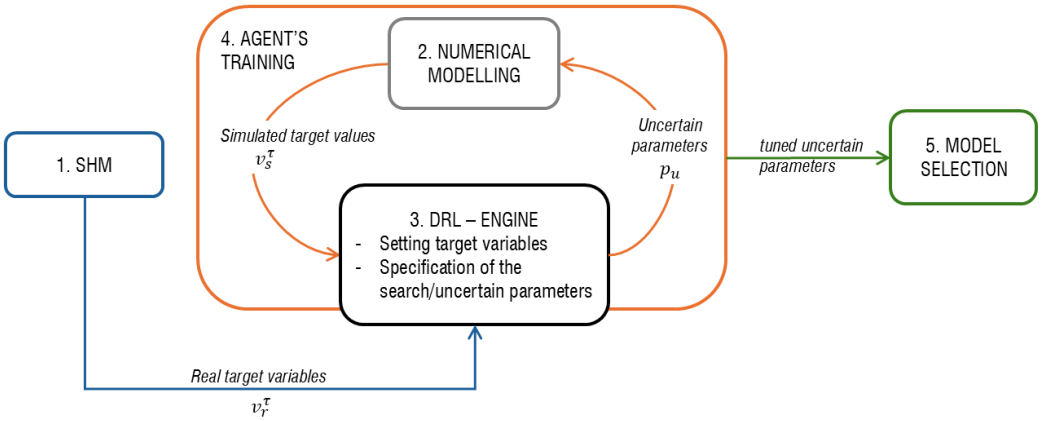


Figure 24. General framework of DRUM-Av1.0.

3.2.1 Step1: SHM and definition of the target variables

The first step of DRUM-Av1.0 consists of defining the best strategy to enhance the knowledge of the real structure, by collecting experimental data to exploit for characterizing the structural behaviour. A SHM campaign must be designed and performed on the structure to investigate. As described in Section 2, the SHM is defined as a strategy for a continuous damage identification, i.e., the variation of geometrical, mechanical, and connectivity (e.g., boundary conditions) properties respect to the original configuration (Fuenteset al., 2020). The SHM system consists of three fundamental elements: (H. Li et al., 2014): (i) a sensor system; (ii) a data pre-processing system (i.e., for acquisition, transmission, and storage); (iii) a data post-processing system (i.e., identification algorithms). Regarding points (i) and (ii), different aspects should be defined, such as the type, the number, and the placement of sensor. Wired sensors ensure continuous monitoring that feeds acquisition units for subsequent processing, while wireless sensors allow for timely monitoring with fewer time constraints. The choice and distribution of sensors must optimise structural representativeness, also considering costs and parameters to be measured. Number and location of sensors depends on the structural representativeness to achieve, and on the type of sensors, the parameters to monitor, the costs of the experimental campaign (Sun & Büyüköztürk, 2015; Ostachowicz et al., 2019). From an analytical point of view, vibrational methods (developed in both the time and frequency domains) are mainly used to extract dynamic

characteristics of the structure such as natural frequencies, modal shapes and damping ratios. Alternatively, deformation-based methods allow localised phenomena to be evaluated using representative measurements.

The selection of the most suitable SHM methodology depends on the type of structure and the specific aspects to be monitored, and therefore it is not possible to provide unambiguous guidelines. On the other hand, the output of the desired SHM campaign represents the input of DRUM-Av1.0. In fact, according to the monitored features and the related values, the real target variables (v_r^τ) are defined. Hence, DRUM-A is set to updating a numerical model according to v_r^τ, which are considered the ground truth for further steps. In the end, two main aspects are following highlighted. The development and testing of DRUM-Av1.0 mainly used natural frequencies derived from operational modal analysis (OMA), which efficiently monitor dynamic structural characterisation. However, the framework can be extended to update numerical models based on other types of target variables, such as modal shapes or global and local deformations, or combinations thereof. It should also be noted that the current version of DRUM-Av1.0 operates through offline updating based on specific data obtained from the SHM campaign and has not yet been implemented for real-time updating, a feature that could be developed in future advances. In brief, step 1 of DRUM-Av1.0 utilises experimental data acquired with appropriately calibrated SHM methods to define target variables, which will serve as a “ground truth” reference for the calibration and optimisation of the numerical model of the structure.

3.2.2 Step2: definition of the FOM

The second step within the proposed framework involves the construction of a high-fidelity numerical model, e.g. the FOM, which forms the basis for the entire updating process. This phase is crucial, as the quality of the model will determine the system's ability to learn and correctly reproduce the actual behaviour of the structure. The FOM aims to accurately simulate the mechanical and dynamic response of the monitored structure, taking into account its actual geometric, material and constraint characteristics. To achieve this, the analyst can adopt different modelling strategies, chosen according to the complexity of the structure, the availability and quality of experimental

data and the degree of detail required. Generally, the Finite Element Method (Bathe, 2008) is the most widely used approach for modelling buildings and civil infrastructure, thanks to its versatility and ability to handle a large number of degrees of freedom. However, in specific cases, the Boundary Element Method (BEM) (Aliabadi, 2020) or the Discrete Element Method (DEM) (Bićanić, 2004) may also be used, which are particularly suitable for surface interaction problems and structures with fragmented behaviour, respectively. A fundamental aspect for the integration of FOM into the DRUM-Av1.0 method is that the Numerical Software Environment (NSE) must be open and accessible externally, allowing automated interaction during the iterations of the agent learning process. This requirement ensures that automatic simulations are performed, allowing the agent to progressively vary the values of the uncertain parameters and update the model until the optimal configuration is achieved. Within the FOM, all elements that significantly influence the dynamic response of the structure are defined as uncertain parameters (p_u). These parameters may include mechanical properties of materials (such as elastic modulus or density), external or internal constraint conditions, and local geometric characteristics that are not perfectly known. Each variation in these parameters produces a set of simulated target variables (v_s^τ), which must be compared with the actual target variables (v_r^τ), previously obtained from the SHM campaign. In the initial configuration of the model, the two quantities do not coincide: the objective of the updating process is precisely to progressively reduce the difference between v_s^τ and v_r^τ. When constructing the FOM, certain fundamental principles must be observed:

- Definition of the level of accuracy. The analyst must establish the desired level of detail for the model from the outset, as the initial configuration cannot be modified during the DRUM-A process.
- Reduction of preliminary assumptions. Unlike traditional updating methods, it is not necessary to define the values of uncertain parameters a priori, as these are manipulated and refined by the agent during training.

- Model elasticity. Simulations conducted to calculate target variables are limited to the elastic range, consistent with the linear nature of experimental data derived from dynamic tests.

The FOM thus defined therefore represents the high-fidelity numerical reference on which the learning system operates. It represents an "artificial environment" in which the DRL agent explores, experiments and converges towards the optimal solution that best reproduces the observed structural behaviour. The care with which this model is set up directly affects the effectiveness of the entire updating process, making its definition one of the most delicate moments in the implementation of the DRUM-A method.

3.2.3 Step3: DRL engine

The core of DRUM-A consists of formalizing the DRL framework for tuning the NM. The main aim is to search the optimal values of p_u to minimize the difference between v_s^τ and v_r^τ. According to the theoretical background in Section 2, four steps must be detailed (described in the next subsections):

i) The environment, the state space *S*, and the observation space Ω.
ii) The available actions for the agent, enclosed in *A*.
iii) The reward r_t and the selected function to maximize it.
iv) The overall agent learning strategy.

3.2.3.1 Environment, state space, and observation space

DRUM-Av1.0 operates within the NSE where the state space S represents all possible states of the system. The observed state at timestep t, denoted as $\omega_{(t)}$, is a vector combining the simulated target variables v_s^τ and uncertain parameters p_u of the NM, and considering a fully observable space for NSE, $\mathbf{\Omega} = \mathbf{S}$, $s_{(t)} = \omega_{(t)}$. The observed vector space is thus $\omega_{(t)} = \{v_{s(t)}^\tau, p_{c(t)}\}$. The simulated target variables may include dynamic characteristics such as frequencies, modal shapes, damping ratios, or sensor-measured accelerations, velocities, and displacements, $v_{s(t)}^\tau = \left[v_{s(t)}^{\tau 1}, \dots, v_{s(t)}^{\tau k}, \dots, v_{s(t)}^{\tau l}\right]$. The uncertain parameters, $p_{u(t)} = \left[p_{u(t)}^1, \dots, p_{u(t)}^j, \dots, p_{u(t)}^n\right]$, correspond to unknown structural properties such as:

- Geometric features not fully obtainable from in-situ surveys.
- Physical material properties (e.g., densities) unmeasurable on site.
- Mechanical properties (e.g., elastic moduli) lacking detailed test data.
- External constraints (e.g., soil-structure interaction).
- Internal constraints among structural elements (e.g., diaphragm effects).

While the uncertain parameters p_u do not require strict engineering assumptions. On the other hand, it is suggested to specify realistic validity ranges to constrain the observation space and improve learning efficiency by excluding non-plausible values (e.g., unrealistic elastic moduli).

3.2.3.2 Action space

At each timestep t, the agent performs actions $a_{(t)}$ within the action space $\mathbf{A}$. Since the uncertain parameters form a vector, the actions are also vector-valued with each component corresponding to a parameter adjustment, $a_{(t)} = \left[a^1_{(t)}, \dots, a^j_{(t)}, \dots, a^n_{(t)}\right]$. Actions may be continuous or discrete (Aradi, 2020). In the first case upper and lower bounds define the range of possible adjustments, as $a(p^j_{u(t)}) \in [a(p^j_u)^{min}, a(p^j_u)^{max}]$; while in the discrete spaces, a finite set of possible changes is predefined, i.e., $a(p^j_{c(t)}) \in \{a(p^j_c)^{min}, \dots, a(p^j_c), \dots, a(p^j_c)^{max}\}$. Actions alter the uncertain parameters additively or multiplicatively with positive or negative signs relative to their values from the previous timestep. As for example, if the action is additive with positive sign, and the agent establish to add the maximum value of the available actions, the j^{th} structural control parameters at the timestep t is expressed as $p^j_{u(t)} = p^j_{u(t-1)} + a(p^j_u)^{max}$.

3.2.3.3 Reward

Besides of observing the $\omega_{(t)}$, the agent receives from the environment NSE the reward, $r_{(t)}$, at each timestep t. According to Section 3.1, DRUM-Av1.0 relies on the cumulative reward at the end of the episodes, by employing a policy-based method, that is, proximal policy optimization, PPO (see next Section). Note that the iterations for

all timesteps characterize an episode *e*, from which a solution is obtained. Running more episodes, the agent is able to update the policy for each episode *e* and maximize the expected cumulative reward while minimizing the difference between $v_{s(t)}^{\tau}$and v_r^{τ}. In the case in which, at the end of the timestep, the obtained solution is worse than the previous one, the agent is penalized, and the value of the cumulative reward reduces. Let *m* the number of iterations for episode *e* (i.e., timesteps in each episode *e*), and $\omega_{(t)}$ and $a_{(t)}$ dependent on $v_{s(t)}^{\tau}, v_r^{\tau}$, and $p_{u(t)}$, the reward in DRUM-Av1.0 is (François-Lavet et al., 2018):

$$r_{(t)} = R(v_{s(t)}^{\tau}, v_r^{\tau}, p_{u(t)}, m) \tag{15}$$

The reward function *R* and the related value of the reward $r_{(t)}$ must be specified case-by-case, according to the specific variables and parameters characterizing $\omega_{(t)}$. In addition, different versions of $r_{(t)}$ could be used, according to the analyst's preferences. As for example, in the application of DRUM-Av1.0 on the case study reported in Section 4, a detailed function for $r_{(t)}$ was defined by penalizing the agent with negative values was elaborated for the problem investigated.

3.2.3.4 Agent learning strategy

The learning strategy depends on the nature of state $\mathbf{S}$ and action $\mathbf{A}$ spaces, if they are discrete or continuous. Algorithms suitable for discrete domains include Monte Carlo Learning (Andrieu et al., 2003), Temporal Difference Learning (Abdi & Moshiri, 2013), SARSA (Wen et al., 2022), and PPO (Schulman et al., 2017), while continuous or large spaces use PPO (Schulman et al., 2017), Deep Q-Learning (Mnih et al., 2015), Deep Deterministic Policy Gradient (Lillicrap et al., 2015), or Soft Actor-Critic (Haarnoja et al., 2018). The agent iteratively updates its policy π to select the best set of actions, $a_{(t)}$, maximizing expected cumulative reward, gradually refining parameter estimates. For performing the latter, DRUM-Av1.0 employed a V-value function and, according to Equation (9), it can be defined as

$$V^{\pi}(s) = E\left[\sum_{i=1}^{m} \gamma^{i}\, r_{(t)} \middle| v^{\tau}_{s(t)}, v^{\tau}_{r}, p_{u(t)}, \pi\right] \quad (16)$$

The algorithm to train the agent is reported in the next Section, accounting for all above information, while the agent learning strategy algorithm used in the application of DRUM-A on the case study reported in Section 3.3, is the PPO, for its capability to be used in both continuous and discrete problems.

3.2.4 Step4: agent's training

After defining the algorithms governing the problem and the sets of uncertain parameters, variables, and actions, this section outlines the proposed approach for training the agent. An agent is considered intelligent if it can interact effectively with the NSE to adjust the uncertain parameters of the numerical model, thereby modifying the simulated target variables $v^{\tau}_{s(t)}$ to increase its reward $r_{(t)}$. The training process is iterative and structured into discrete sessions, divided into timesteps and episodes. The practical aim during training is to find the best policy over a given number of episodes ε, maximizing the cumulative reward governed by the policy function π. The analyst specifies the number of timesteps m per episode and the number of episodes ε for the agent to perform the iterative learning steps. If by chance the agent finds the correct solution at the first timestep, it receives the maximum immediate reward $r_{(t)}$, equivalent to the highest cumulative reward of the episode (like a jackpot), and the episode is terminated early, reducing training time. In general, however, the agent runs through all timesteps in all episodes, computing the cumulative reward as the sum of immediate rewards $r_{(t)}$. The agent's training follows typical deep learning principles, aiming for convergence and stability with regard to the loss function, which in this context is the maximization of cumulative reward. Higher numbers of timesteps per episode can increase training accuracy by allowing more opportunities for convergence, but at the cost of longer training times. To enhance learning efficacy, modifications to the neural network architecture underlying the agent's behavior can also be applied. The agent's learning dynamics are illustrated in Figure 25: the agent is depicted as a light blue dot,

and the correct solution as a red dot, with orange lines representing possible paths the agent can take according to actions (green arrows).

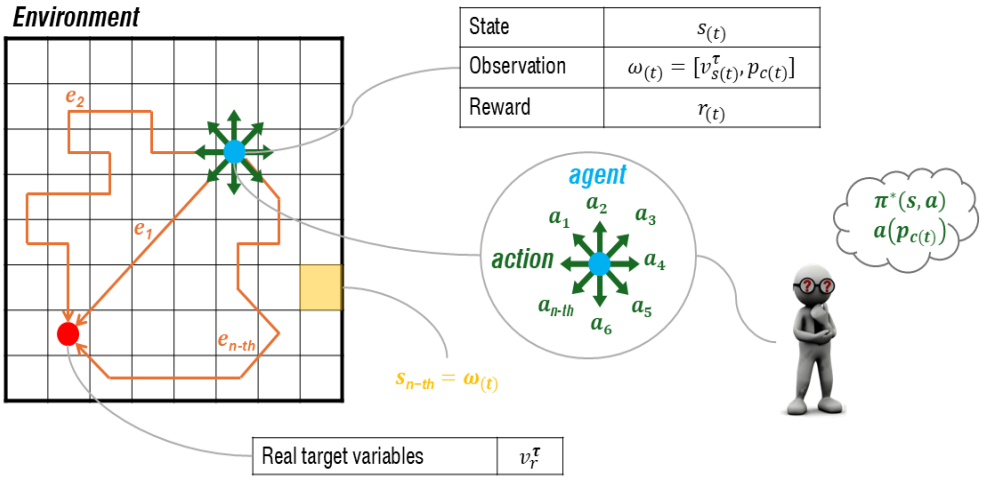


Figure 25. Two-dimensional representation of the agent characteristics and environment of the DRL for the MU problem.

The detailed training algorithm is summarized in Algorithm 1, which integrates the full DRUM-A methodology from Step 1 to Step 4. During training (lines 11 to 32 of Algorithm 1), the agent repeatedly performs the following:

1. Modify the uncertain parameters of the numerical model within the NSE.
2. Execute the analysis to obtain simulated target variables v^{τ}_s.
3. Access the results and calculate the immediate reward based on the discrepancy between v_s and the real target variables v^{τ}_r.

Key elements of the algorithm include the explicit interaction between agent and NSE (lines 17–22), where model parameters are set, analyses run (e.g., eigenvalue analysis for extracting frequencies), and outputs retrieved. Termination rules (line 26) optimize training duration and may end episodes early if: (a) a reward threshold expressed as a percentage of the maximum reward is not met, or (b) the solution is considered sufficiently accurate according to a predefined tolerance on the difference between v^{τ}_s and v^{τ}_r. This mechanism helps reduce unnecessary training time.

Algorithm 1. Outlines the fundamental interaction loop between DRUM-A and the NSE, guiding the agent toward learning an optimal policy for effective model updating.

	First algorithm of DRUM-A	
1	***Output SHM**: read real target variables v_r^τ*	← ***STEP 1***
2	*Initialize the environment*	← ***STEP 2***
3	***Run the NSE***	
4	***Open** structure model file*	
5	*Start DRL-ENGINE algorithm*	← ***STEP 3***
6	***Input:** set type of uncertain parameters $p_{u(t)}$*	
7	*set type of simulated target variables $v_{s(t)}^\tau$*	
8	*set observation space* **Ω**	
9	*set action space* **A**	
10	***Output:** Read the state of environment $s_{(t0)} = \omega_{(t0)}$*	
11	***For** episode = 1: ε **do:***	← ***STEP 4*** ↻
12	*Reset the initial state randomly*	
13	***While** True **do:***	
14	*Observe the state $s_{(t)} = \omega_{(t)}$*	
15	*Select action $a({p_u}^j)$ using the old policy π*	
16	*Compute new values of uncertain parameters $p_{u(t)}$*	
17	***In NSE:***	
18	***Set** uncertain parameters $p_{u(t)}$*	
19	***Set** analysis*	
20	***Run** analysis*	
21	***Save** model output*	
22	***Read** simulated target variables $v_{s(t)}^\tau$*	
23	***Output:** Obtain the observation $\omega_{(t)}$*	
24	*Calculate reward r_t and obtain new state $s_{(t+1)}$*	
25	*Update policy π*	
26	*Check terminating rules*	
27	*Break while*	
28	*Update timestep t+=1*	
29	***End While***	
30	*Compute cumulative reward*	
31	*Save new optimal policy function π^**	
32	***End for***	

3.2.5 Step5: model selection

The final step of the proposed procedure involves defining the best solution achieved by the trained agent. This means determining the final values of the uncertain parameters $p_{u(t)} = \widehat{p_u}$, which correspond to the estimated optimal parameters introduced in Step 3 and represent the outcome of the model updating (MU) process, i.e., the model selection. To carry out this, the analyst establishes a set number of timesteps and episodes (different from the training phase). For each episode e, the trained agent performs the same operations described in Step 4 but without modifying the optimized policy function π^*. Due to its acquired experience, the agent knows how to achieve the maximum cumulative reward. At the end of each episode, a solution in terms of $p_{u(t)} = \widehat{p_u}$ is obtained. Repeating this process over multiple episodes ε produces a corresponding set of solutions. If approximately 90% of these solutions reach convergence, i.e., meaning the difference between the estimated simulated target variables $v_{s(t)}^{\tau} = \widehat{v_s^{\tau}}$ and the real target variables v_r^{τ} falls below a tolerance defined by the analyst, the agent is deemed effective in performing the MU task. If this convergence rate is not met, it indicates inadequate training and necessitates another training phase possibly with adjustments to the termination criteria. From the subset of convergent solutions, the final optimal parameter values $\widehat{p_u^J}$ are analyzed statistically. The distribution of each uncertain parameter $\widehat{p_u^J}$ is evaluated, and the modal values (most frequent) of their statistical distributions are selected as the optimal solution for the FOM after applying DRUM-Av1.0.

Moreover, the performance of the output can be evaluated by assessing the ability of the trained agent to consistently reach solutions. This is done by measuring the dispersion of the parameter estimates $\widehat{p_u}$. Depending on the type of their statistical distribution $\widehat{p_u^J}$, different metrics are used: (i) Coefficient of Variation (CV) for normal and lognormal distributions; (ii) range for uniform distributions; (iii) standard deviation for triangular distributions. When agent training is effective, the statistical distributions of each parameters $\widehat{p_u^J}$ should not show strong asymmetries or numerous outliers. A clear indication about the maximum value of the dispersion for each statistical

distribution cannot be provided, but the existing literature can support the decision. As for example, in case of normal distribution of each $\widehat{p_u^J}$, the maximum allowed value of CV can be fixed at 10% (see Mangalathu et al., 2021; Aronhime et al., 2013).

3.3 Dummy case study

The proposed methodology was applied to a simple case study, a dummy RC table structure characterized by four columns, four beams and one storey (Figure 26). Although the investigated structure is simple, this choice was driven by the necessity to maintain full control on all the physical and mechanical characteristics of the structure, in order to assess the effectiveness of the proposed method. In addition, the simplicity of the structure is apparent, because apart from the geometry, several properties were differentiated to increase the number of p_u, and then the number of uncertain variables. The geometrical characteristics of structural elements are shown in the Table 7, in which indication about three types of structural materials is provided (M_1, M_2, M_3). The mechanical properties of the concrete typologies are reported in Table 8, in which are present the specific weight (w_s), the elastic modulus (E), the Poisson coefficient (ν), and the shear modulus (G).

Table 7. Characteristics of the dummy case study elements.

Element	Id.	Section (mxm)	Length (m)	Materials	Nodes Id.
Columns	C1	0.5 x 0.5	5	M_1	1-5
	C2	0.5 x 0.5	5	M_1	2-6
	C3	0.5 x 0.5	5	M_2	3-7
	C4	0.5 x 0.5	5	M_2	4-8
Beams	B1	0.3 x 0.5	10	M_3	5-6
	B2	0.3 x 0.5	5	M_3	5-7
	B3	0.3 x 0.5	5	M_3	6-8
	B4	0.3 x 0.5	10	M_3	7-8

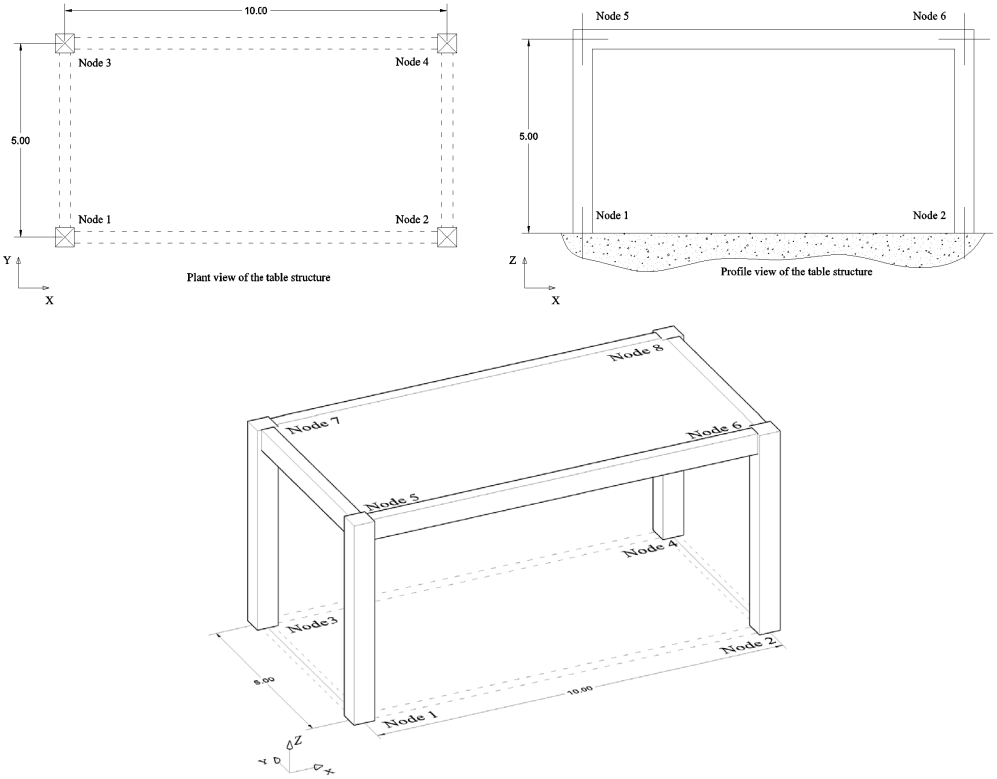


Figure 26. Structural diagram of the dummy table model.

Table 8. Mechanical characteristics of the materials of structural elements

Material	W_s **[kN/m³]**	E **[MPa]**	ν	G **[MPa]**
M_1	25.0	30000	0.2	12500
M_2	25.0	33000	0.2	13750
M_3	25.0	31000	0.2	12917

The floor was considered as infinitely rigid and uniformly distributed loads were applied, i.e., permanent structural loads equal to 3.5 kN/m^2, permanent non-structural loads equal to 2.5 kN/m^2 and live loads equal to 2.0 kN/m^2, according with the Italian Building code for residential buildings (MIT, 2018). At the base of the dummy structure, different restraint typologies were considered. Nodes 1, 3, and 4 were considered fixed at the base, while degree of freedoms of node 2 were considered null in terms of

translations (for the main directions *X*, *Y*, *Z*, according to the reference system in Figure 26), null in terms of rotation around the Z axis, and assume certain values in terms of rotations around *X* and *Y* axes. In fact, two rotational springs were considered in *X* and *Y* directions, indicated as K_{rx} and K_{ry}, respectively, and assumed value of $1.0\mathrm{E}+10$ kN*m/rad.

Let us suppose that the above defined structure is the one for which MU should be performed. It is worth clarifying that, although the mechanical parameters and the degree of restraint of the dummy structure were detailed, in the application of the proposed procedure the values of K_{rx}, K_{ry}, E_1, and E_2, were assumed as uncertain, while other quantities were considered known. This assumption, made for demonstration purposes only, does not limit the applicability of DRUM-Av1.0, when increasing number of uncertain parameters are present. This particular set up of the problem allows to identify the values of K_{rx}, K_{ry}, E_1, and E_2 as target uncertain parameters $p_u^\tau = [K_{rx}, K_{ry}, E_1, E_2]$, which represent the real values of the control parameters that agent should achieve (always unknown in real-life problems).

3.3.1 Surrogate SHM and NM

For the surrogate SHM campaign, since laboratory reproduction was not feasible, numerical simulation was used to generate reference target variables. Specifically, the first three vibration frequencies from an OMA. The FEM was constructed in SAP2000 (Computers and Structures, Inc., 2024), and the system was excited with an ambient vibration accelerogram. Definition of the environmental excitation. An accelerogram simulating an ambient vibration in a time range of 600 seconds and sampling frequency equal to 0.01 Hz was applied at the base of the structure (see Figure 27). Simulated accelerations were collected from selected nodes and processed using ARTeMIS software (Structural Vibration Solutions, 2023) for modal identification, ensuring that the reference values for the frequencies matched the numerically computed ones.

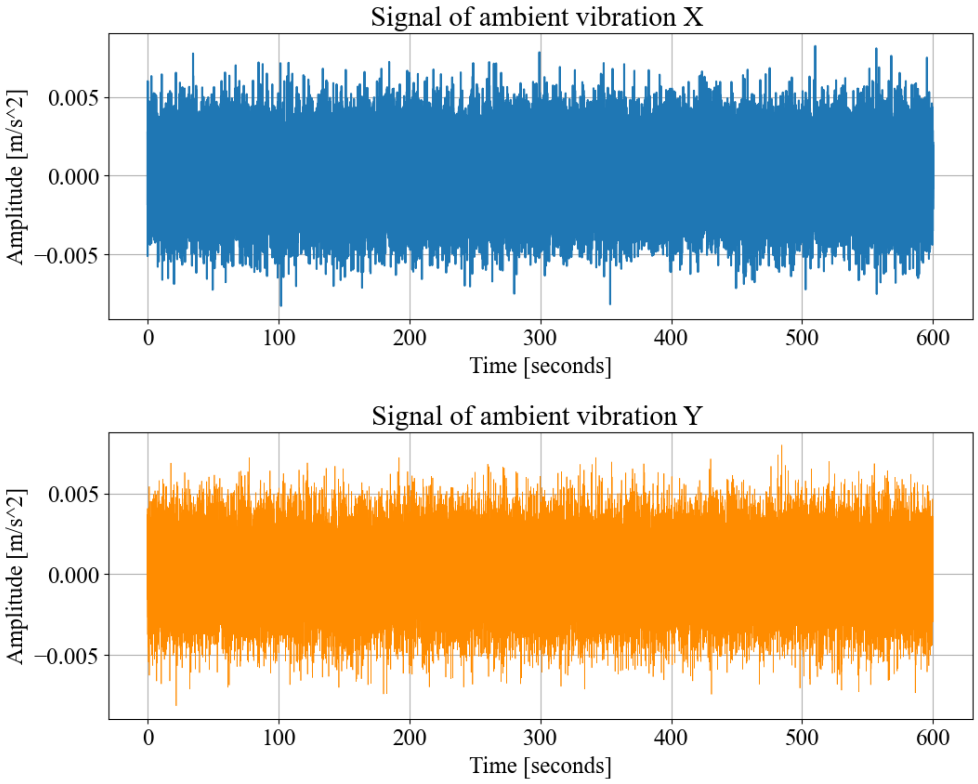


Figure 27. Accelerograms applied in the X-direction (top) and Y-direction (bottom) to simulate the presence of white noise on the structure

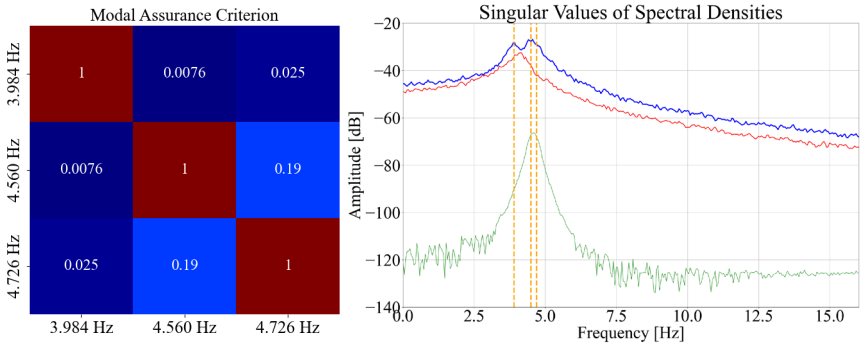


Figure 28. Results obtained from OMA. On the left, the Modal Assurance Criteria values are reported for the first three vibration modes. On the right, the Singular Value Decomposition graph is reported.

Table 9. Values of the first three vibration modes for the dummy structure

Vibration mode	Frequency [Hz]
1° mode	$f_1^{SHM} = 3.9839$
2° mode	$f_2^{SHM} = 4.5603$
3° mode	$f_3^{SHM} = 4.7265$

The OMA-extracted frequencies $[f_1^{SHM}, f_2^{SHM}, f_3^{SHM}]$ defined the real target vector v_r^{τ}. Following the procedure at the Step 2, a NM of the structures should be realized. Nevertheless, having the FEM of the dummy structure, an initial and manual modification of the uncertain parameters, i.e., K_{rx}, K_{ry}, E_1, and E_2, was performed, by assuming random values, without any engineering assumption on a value range. According to the description in Section 3.2.3, the adequate degree of detail for the modelled structure is intrinsically correct, because the unknown parameters were modelled. As output of this new NM, representing the first version of the simulation, the simulated target variables were defined after eigenvalue analysis, i.e., $v_{s(t0)}^{\tau} = [f_{1(t0)}^{FEM}, f_{2(t0)}^{FEM}, f_{3(t0)}^{FEM}]$, as well as the initial random uncertain parameters, i.e., $p_{u(t0)} = [E_{1(t0)}, E_{2(t0)}, K_{rx(t0)}, K_{ry(t0)}]$.

3.3.2 Definition of DRL-engine

The DRL engine represents the computational core of the proposed DRUM-Av1.0 framework, specifically designed for the model updating (MU) of structural systems. For this dummy case study, the engine was implemented to interact autonomously with the finite element model of the RC table structure (Figure 29). In this configuration, the NSE is represented by SAP2000, a commercial structural analysis program. Access to the NSE was automated through its Open Application Programming Interface (OAPI) (Computers and Structures, Inc., 2024), which makes it possible to directly read, modify, and execute the FE model from an external environment. The DRL engine was coded entirely in Python programming language (Python Software Foundation, 2024). This allowed seamless communication between the Python-based agent and SAP2000, including model initialization, parameter modification, execution of eigenvalue analyses, and automatic extraction of modal results after each iteration.

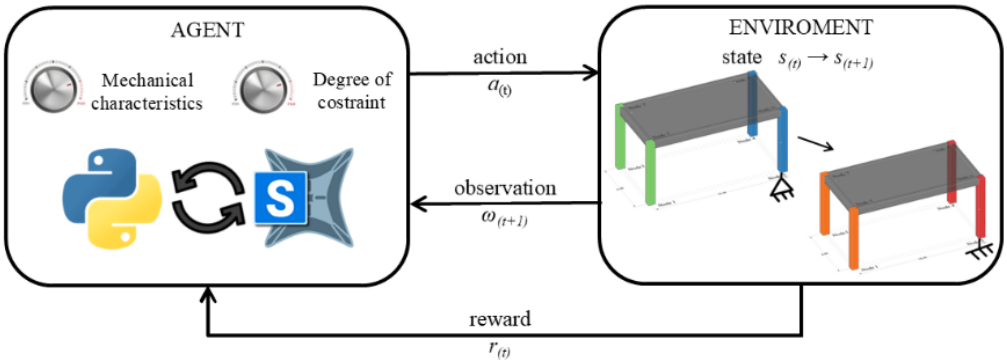


Figure 29. DRL-engine for MU

According to the structure defined in Section 3.3.1, at each timestep *t*, the observation $\omega_{(t)}$ was defined as the combination of the simulated outputs and the uncertain structural parameters: $\omega_{(t)} = \{v^{\tau}_{s(t)}, p_{u(t)}\} = \{f^{FEM}_{1(t)}, f^{FEM}_{2(t)}, f^{FEM}_{3(t)}, E_{1(t)}, E_{2(t)}, K_{rx(t)}, K_{ry(t)}\}$ where $v^{\tau}_{s(t)}$ are the simulated target variables (the first three natural frequencies obtained via eigenvalue analysis) and $p_{u(t)}$ are the structural uncertain parameters (the elastic moduli of materials M1 and M2, and the rotational stiffnesses of the springs at node 2). To limit the dimensionality of the observation space (**Ω**) and ensure physical plausibility, boundary conditions were imposed on the parameters, for $E_1, E_2 \in [29000, 35000]$ Mpa and for $K_{rx}, K_{ry} \in [0.0 \times 10^0, 1.0 \times 10^{12}]$ kN·m/rad (corresponding to the support and to fixed base respectively). This setup ensured that all parameter updates remained within realistic engineering constraints.

The action space (***A***) defines the set of discrete modifications the DRL agent can apply to the uncertain parameters at each time step. For this case, a multi-discrete space was chosen, where each structural parameter could be adjusted according to five possible variations (Δ_1 to Δ_5), either additively or multiplicatively. The range of possible actions is summarized in Table 10. At each iteration, the agent selects an action for every parameter: $\Delta E_1, \Delta E_2$ (additive adjustments)

and $\Delta K_{rx}, \Delta K_{ry}$ (multiplicative adjustments). Choosing $\Delta_3 = 0$ indicates no modification from the previous timestep (t-1).

Table 10. Multi-discrete action space available for the agent

Set of actions on the p_u	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5
$\mathbf{\Delta E_1}$	-150	-80	0	80	150
$\mathbf{\Delta E_2}$	-80	-30	0	30	80
$\mathbf{\Delta K_{rx}}$	0.1	0.2	1	5	10
$\mathbf{\Delta K_{ry}}$	0.1	0.2	1	5	10

Following the parameter updates, an eigenvalue analysis was automatically performed in SAP2000, and the resulting modal frequencies were extracted to form the new observation $v^{\tau}_{s(t)}$.

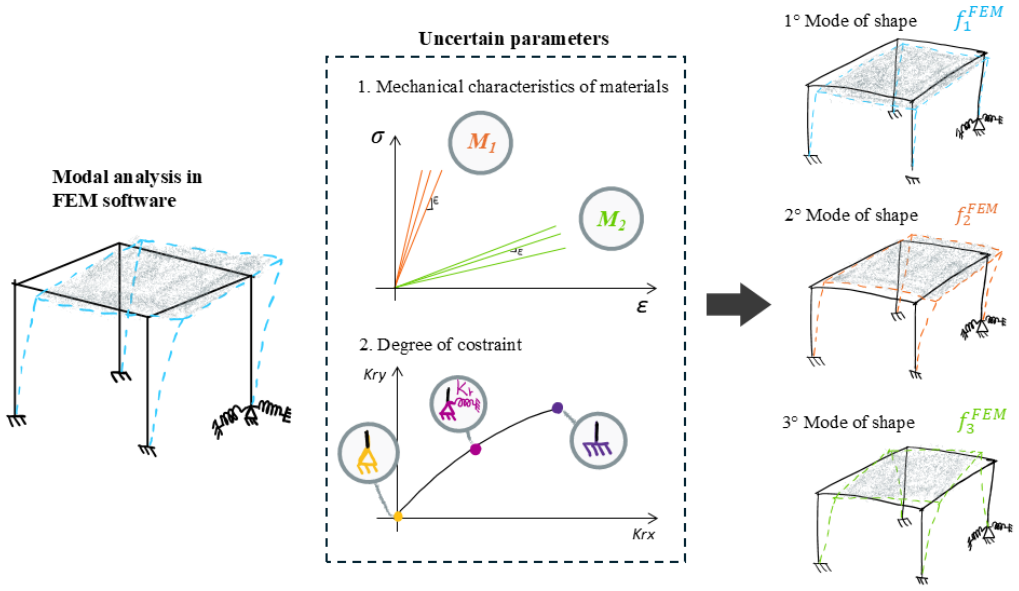


Figure 30. Action space of DRUM-Av1.0 in the environment NSE.

A customized reward function was developed to guide the training process by penalizing deviations between the simulated and real frequencies. The agent continuously received a negative reward, approaching zero as the frequency mismatch

decreased, forcing it to minimize the difference between v_s^τ and v_r^τ as efficiently as possible. For each timestep *t*, the selected reward defined for this problem was:

$$r_{(t)} = -Co + \sqrt[3]{\Delta\sigma} + p \tag{17}$$

where *Co* is a constant equal to 1, *p* is the penalization, $\Delta\sigma$ is expressed in Equations (18) and (19):

$$\Delta\sigma = \sigma_{(t-1)} - \sigma_{(t)} \tag{18}$$

$$\sigma_{(t)} = \frac{\sum\left|\boldsymbol{v_r^\tau} - \boldsymbol{v_{s(t)}^\tau}\right|}{b} \tag{19}$$

where $\sigma_{(t)}$ represents a scaled value of the difference between the real target variables $v_r^\tau = [f_1^{SHM}, f_2^{SHM}, f_3^{SHM}]$ and the simulated target variables $v_{s(t)}^\tau = [f_{1(t)}^{FEM}, f_{2(t)}^{FEM}, f_{3(t)}^{FEM}]$, *b* is a manually scaling fine-tuned parameter equal to 3, $\sigma_{(t-1)}$ represents the same scaled difference at the previous timestep, *t-1*. As occurs for the custom equations, Equation (17) is characterized by four fundamental aspects:

1. c with the negative sign is a constant component that the agent receives at each timestep *t*. This implies that for each iteration, a fixed penalization is always assigned. Hence, to reduce the penalization, the agent is forced to quickly converge toward the solution.
2. $\sigma_{(t)}$ represents a normalized measure of the distance between $v_{s(t)}^\tau$ and v_r^τ at each timestep.
3. $\sqrt[3]{\Delta\sigma}$ compares the distance $\sigma_{(t)}$ in two subsequent timesteps. By minimizing this component, the agent is driven to more direct trajectories toward the right solution. The cube root is used to properly scale the $\Delta\sigma$ value.
4. *p* returns a penalization equal to -1 only when the agent updates structural control parameters, which assume values exceeding the observation space.

Each episode ended when one of the following criteria was satisfied: (i) the total number of timesteps m was reached; (ii) the deviation between simulated and real values dropped below a prescribed tolerance ($\sigma_{(t)} < \sigma_{(t,end)} = 0.05$). The training logic is summarized in Algorithm 2, implemented in Python to coordinate model

updates, result extraction, and convergence assessment across 100 timesteps per episode.

Algorithm 2. *Rules for episode termination in DRUM-A (Python–SAP2000 integrated engine).*

Terminate rules	
1	***Output SHM**: read real target variables*
2	$v_r^T = [\, f_1^{SHM}, f_2^{SHM}, f_3^{SHM}]$
3	*start timestep t (Algorithm 1-lines 14-25)*
4	$v_{s(t)}^T = \left[f_{1(t)}^{FEM}, f_{2(t)}^{FEM}, f_{3(t)}^{FEM}\right]$
5	***while** True **do**:*
6	$\sigma_{(t)} = sum\left(f_{1,2,3}^{SHM} - f_{1,2,3(t)}^{FEM}\right)/b$
7	***if** $\sigma_{(t)} \leq \sigma_{(t,end)}$ or m =100 **do:***
8	*Save results*
9	*Break*
10	***end if***
11	***end while***
12	*Save results*

The Proximal Policy Optimization (PPO) algorithm (Schulman et al., 2017) was adopted to generate and refine the policy guiding the agent's interactions with the NSE. PPO is a policy-gradient method that optimizes a clipped surrogate objective function $J(\psi)$ to maintain stable and efficient learning, particularly in high-dimensional continuous problems. These two fundamental aspects can guarantee excellent stability, ensured also by its excellent performance for operating in complex and high-dimensional environments. From the formal point of view, PPO presents an actor-critic architecture, where both the actor, η, and the critic, *V*, are approximated by deep neural networks with the parameters θ^{η} and θ^{V}, respectively. The critic network is used to estimate the value of the state-value function and provide feedback to the actor through an advantage function (enclosed in the objective function $J(\psi)$), while the actor network is used to estimate the policy function $\pi_{\psi}\,(a|s, \theta^{\eta})$, which for the case at hand leads to have a policy described by a normal distribution (*N*), $\pi_{\psi}(a|s) \sim N(\mu, \sigma^2)$, defined through mean μ and the standard deviation σ. A detailed explanation of PPO can be found in Schulman et al. (2017).

3.3.3 Training and test of the agent

The training phase of the DRUM-Av1.0 framework was implemented according to the setup described in Figure 29. Computations were executed on a workstation equipped with a 12th Gen Intel(R) Core(TM) i7-12700H CPU, 32 GB RAM, and an NVIDIA GeForce RTX 3070Ti GPU with 8 GB of VRAM. This configuration ensured a balance between computational performance and stability when executing multiple FE analyses sequentially through the Python–SAP2000 interface. The training phase used the data obtained from the surrogate SHM campaign (Section 3.3.1), specifically the true vibration frequencies (v_r^τ) and corresponding uncertain parameters (p_u^τ). The agent was initialized to start from the lower extremes of the uncertain parameter vector $p_u = [E_1, E_2, K_{rx}, K_{ry}]$, progressively adjusting values during training to minimize the deviation between simulated and real target variables. This initialization strategy, though simplified, allows a consistent and controlled evaluation of the algorithm's ability to explore and converge autonomously. The training was performed for 1000 episodes, each containing a maximum of 100 timesteps. The termination conditions were those reported in Algorithm 2: stopping when the maximum timesteps were reached or when the deviation value σ_t fell below the fixed tolerance value of 0.05.

Figure 31 illustrates the evolution of the cumulative reward over all episodes. Each grey line represents the individual per-episode reward, while the orange line indicates the simple moving average, which smooths out short-term oscillations, giving a clearer view of the general learning trend. In the initial episodes (1–100), the agent's cumulative reward fluctuated strongly between -60 and -30, reflecting the exploration phase where the policy was highly variable. After approximately 500 episodes, the cumulative reward stabilized between -20 and 0, marking the convergence region of the training. The simple moving average reveals a consistent upward trend in performance, confirming an increased efficiency in state–action policy updates and a stable PPO-based learning behavior. Overall, the reward converged at around -15, corresponding to an average deviation $\sigma_{(t)}$ below the tolerance threshold, confirming effective learning convergence.

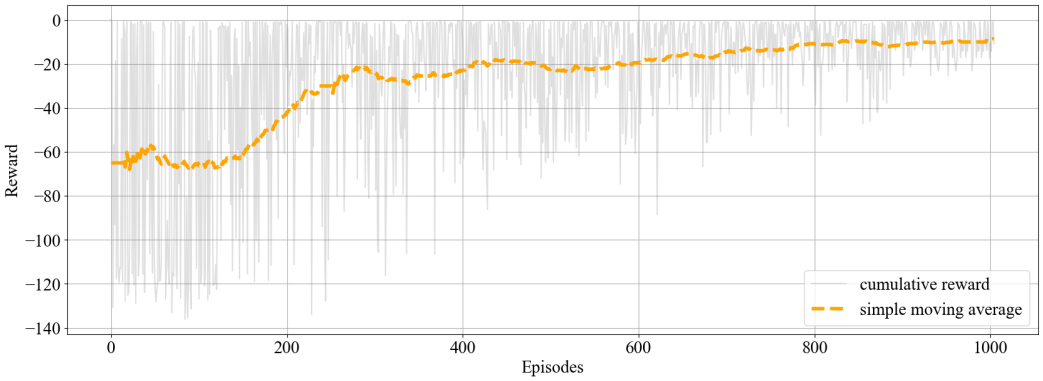


Figure 31. Cumulative reward trend during the training phase. The orange line shows the simple moving average; best, average, and worst episode performances are indicated.

In Figure 32, the evolution of the deviation function $\sigma_{(t)}$ (Equation 19) is plotted at the end of each episode. During the first 100 episodes, the $\sigma_{(t)}$ values remain relatively high, which indicates an unstable policy behavior typical of DRL exploration stages. After this early phase, $\sigma_{(t)}$ gradually decreases and the moving average (in dashed red) stabilizes around a value below $\sigma_{(t,end)} = 0.05$, confirming policy convergence.

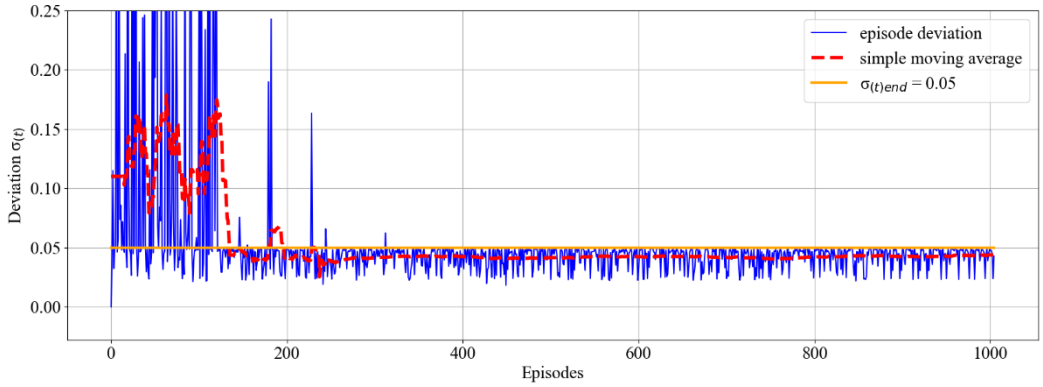


Figure 32. Evolution of $\sigma_{(t)}$ and its moving average during the training phase; convergence within the set tolerance.

The complete training process required less than 9 hours. This overall duration accounted for executing over 100,000 FE eigenvalue analyses in SAP2000, transferring results to Python, and updating the DRL algorithm's state after each iteration.

Importantly, approximately one-third of the episodes (about 300) were sufficient for stabilization, meaning the agent achieved practical convergence after roughly 3 hours of computational time.

After training, performance testing was carried out to validate the agent's ability to generalize the learned policy and accurately perform model MU. Two modifications were introduced to introduce variability and prevent overfitting:

1. Modification of the target uncertain parameters and target variables;
2. Adjusting the tolerance threshold to $\sigma_{(\mathrm{t,end})} = 0.03$.

The data used for the test scenario are reported in Table 11.

Table 11. *Testing uncertain parameters and corresponding target frequencies.*

Target uncertain parameters $p_c^{\tau 2}$				Real Target Variables $v_r^{\tau 2}$		
E_1 [Mpa]	E_2 [Mpa]	K_{α}	K_{γ}	f_1^{SHM} [Hz]	f_2^{SHM} [Hz]	f_3^{SHM} [Hz]
31500	31500	1.0E+8.5	1.0E+8.5	3.98443	4.59497	4.69410

The trained agent was tested for 100 episodes, each consisting of 100 timesteps. The uncertain parameters $\widehat{p_u^2}$ were initialized randomly at the start of each episode, and the agent iteratively updated model parameters within the NSE to minimize the difference between $\widehat{v_s^{\tau 2}}$ (simulated frequencies) and $\boldsymbol{v_r^{\tau 2}}$ (real frequencies). Figure 33(a) presents the cumulative reward values for all test episodes (grey) and their moving averages (orange). The agent consistently achieved average reward values around -7, primarily reflecting the constant penalty $c = 1$ introduced in the reward function (Equation 14). Despite starting from random initial states, the agent rapidly converged toward optimal parameter configurations within a few iterations. Figure 33(b) plots the deviation values $\sigma_{(\mathrm{t})}$ for each episode (blue) alongside the moving average (red). The agent consistently crossed the tolerance line $\sigma_{(\mathrm{t,end})} = 0.03$ (orange), confirming convergence within all test runs. Moreover, approximately 20% of episodes reached highly accurate solutions with deviations ≤ 0.02, showing robust precision.

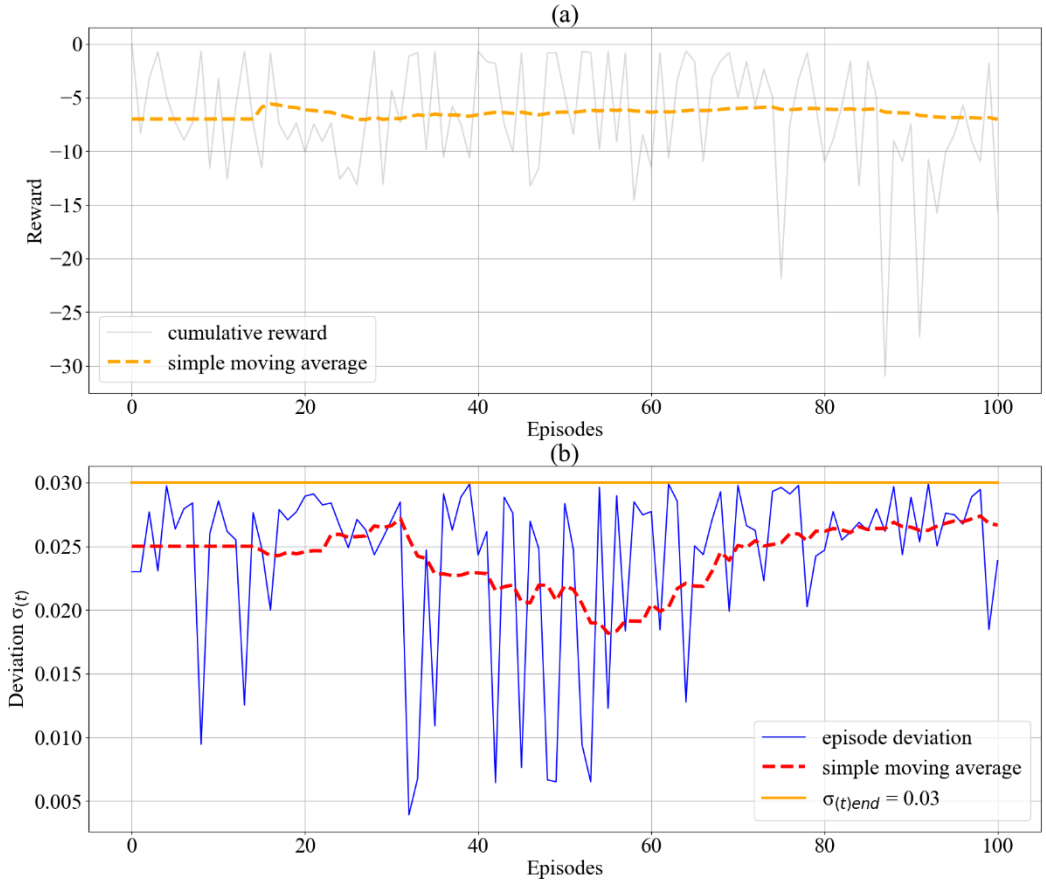


Figure 33. *Results of the DRUM-Av1.0 test phase: (a) cumulative reward trend per episode; (b) deviation $\sigma_{(t)}$ and moving average trend across test episodes.*

Figures 34(a) and 34(b) analyze the relationship between deviations and parameters. *Figure 34(a)* illustrates how E_1 and E_2 combinations influence $\sigma_{(t)}$. The lowest errors occur near $E_1 = 30500$ MPa and $E_2 = 31500$ MPa (light blue zone). *Figure 34(b)* displays similar results for K_{rx} and K_{ry}, where the minimal deviation occurs around 1.0×10^8 kN·m/rad. Figure 35 compares the simulated and real frequency values across all episodes. The color scale represents the deviation $\sigma_{(t)}$. Frequencies with the smallest deviation are nearest to the real target frequencies, confirming correct calibration. Minor discrepancies, such as slightly lower simulated values, are attributed to the initialization method where the training always starts from the low

end of parameter ranges. Nevertheless, deviations remain within the selected tolerance limits.

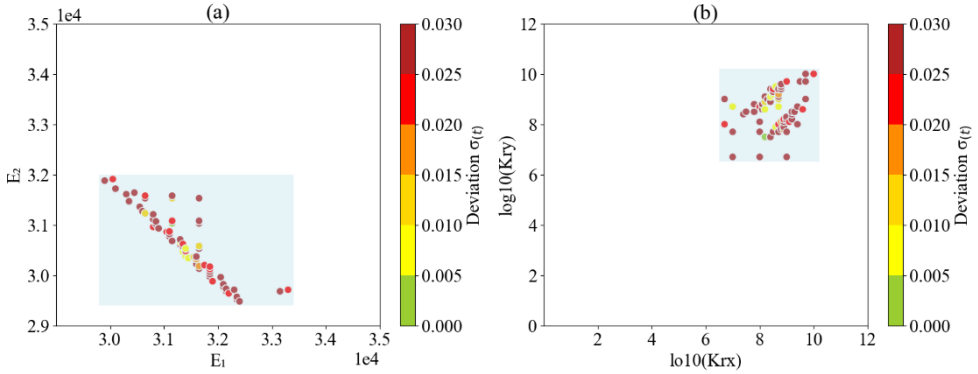


Figure 34. *Dependence of deviation $\sigma_{(t)}$ on estimated uncertain parameters: (a) E_1 and E_2 combinations; (b) K_{rx} and K_{ry} combinations.*

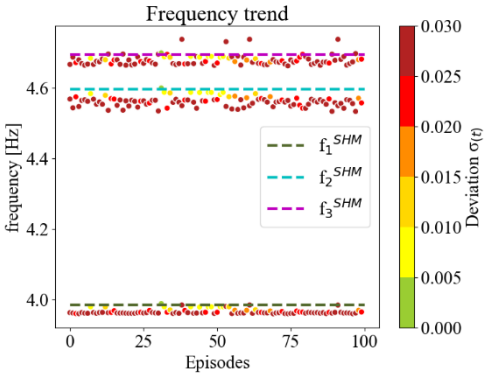


Figure 35. Values of the simulated frequencies $\widehat{v_s^{\tau 2}} = [f_1^{FEM}, f_2^{FEM}, f_3^{FEM}]$ obtained for all episodes and values of the real frequencies $v_r^{\tau 2} = [f_1^{SHM}, f_2^{SHM}, f_3^{SHM}]$ imposed in the scenario

Once the agent was tested for the current case study scenario, the phases of model selection and performance evaluation were performed. According to Section 3.2.5, more than 90% of solutions achieved the convergence (100% for the case at hand), which implies the effectiveness of the agent in performing MU. Therefore, the estimated uncertain parameters $\widehat{p_c^2}$ can be treated in statistical terms, and the results are shown in Figure 36. Several results are reported in Figure 36, which shows the correlations between pairs of structural control parameters E_1, E_2, K_{rx}, K_{ry} (the latter two expressed in logarithmic scale to investigate only the exponent) through pairplot

representations. In each graph, the correlation between parameters is reported, showing on the diagonal the statistical distribution for single parameters, on the upper diagonal the scattered plot distribution of the value pairs, and under the diagonal the continuous kernel density functions (upper and lower diagonal graphs are differently represented, but present the same meaning). The first aspect to highlight from Figure 36 is the output on the diagonal, which show that the statistical distributions of all estimated structural control parameters $\widehat{p_c^2}$ present very low dispersion. Still, the shape of the statistical distributions is normal-like for all estimated uncertain parameters $\widehat{p_c^2}$, as visually definable. Regardless the kind of distribution, which is not a fundamental aspect in the model selection, it is worth providing measures of dispersion, such as standard deviation, and coefficient of variation CV, and measures of the values to consider for the correct MU, such as mode, median, and mean. These values were computed and reported in Table 12, besides showing minimum and maximum values assumed by the agent during the test.

Table 12. Statistics on the results obtained in the testing phase by DRUM-A for the estimated structural control parameters $\widehat{p_c^2}$

Measure	E_1	E_2	$Log_{10}(Krx)$	$Log_{10}(Kry)$
Minimum	29900	29480	6.70	6.70
Maximum	33300	31910	10.0	10.0
Mean	31430	30644	8.55	8.63
Median	31400	30615	8.70	8.65
Mode	31650	31080	8.70	8.49
Standard deviation	704.5	679.0	0.80	0.77
CV (%)	2.24 %	2.21 %	9.36 %	8.92 %

The results show that the dispersion measures are low, especially looking at the values of CV, which present a percentage lower than 10%, as a limit suggested by the literature for analogous problems (e.g., Mangalathu et al., 2021; Aronhime et al., 2013). In addition, the values of mean, median, and modes are very similar, confirming the absence of outliers or asymmetric values of estimated uncertain parameters $\widehat{p_c^2}$ in all combinations at the end of all episodes. Still, the minimum and maximum values

represent a limited subrange of the proposed engineering assumptions (i.e., extremes), which supports the stability of the policy adopted by the agent.

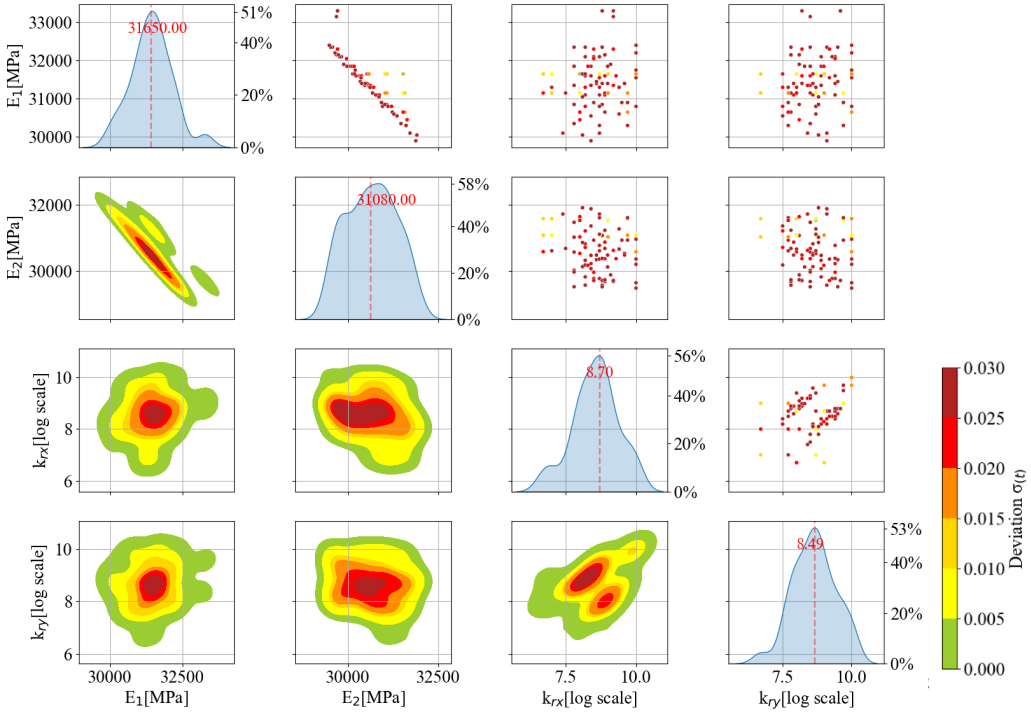


Figure 36. Pairplot of the relationships between pairs of estimated structural control parameters used [E_1, E_2, K_{rx}, K_{ry}] in the testing phase. On the diagonal of the plot matrix the statistical distribution of the individual parameter. The upper and lower diagonal plots represent the correlation between parameter pairs, shown through a colourmap according to the value of the deviation obtained at the end of each episode

Given the success of the test, the results of the MU process are provided by the mode values for all the estimated structural control parameters $\widehat{p_c^2}$, which represent the criterion for the model selection. The values for obtaining the correct solution are reported in Table 13, which show the comparison with the real values. As observed, the percentage differences in terms of percentage variation are very low (< 5%), confirming once the accuracy of DRUM-Av1.0.

Table 13. Mode values of the structural control parameter $\widehat{p_c^2}$ obtained by DRUM-Av1.0 and comparison with real values.

Parameter	E_1 [Mpa]	E_2 [Mpa]	K_{rx} [kNm/rad]	K_{ry} [kNm/rad]
Real values $p_c^{\tau 2}$	31500	31500	1.0E+8.50	1.0E+8.50
DRUM-Av1.0 $\widehat{p_c^2}$	31650	31080	1.0E+8.70	1.0E+8.49
Percentage variation (%)	0.48	1.35	2.35	0.12

3.3.4 Comparative analysis with traditional MU process

To evaluate the overall performance and computational efficiency of DRUM-Av1.0, a comparative analysis was conducted against two traditional MU strategies, a deterministic approach, implemented via the automated procedure SAP.Py; and a probabilistic Bayesian approach with non-informative priors. This comparison aims to assess accuracy, computational effort, and scalability while ensuring equivalent boundary and loading conditions for the dummy case study.

The deterministic MU technique relied on systematically varying the uncertain parameters $p_u^2 = [E_1, E_2, K_{rx}, K_{ry}]$ to minimize the difference between the simulated target variables $v_r^{\tau 2}$ and the real target variables $v_r^{\tau 2}$. The optimization framework was implemented through SAP.Py, a Python-based automated parametric modeling tool (Bruno et al., 2024), which manages parameter combinations in SAP2000, executes analyses, extracts frequencies, and minimizes the residuals according to Equations (4) and (5). To quantify computational demand, four sampling configurations, O_1, O_2, O_3, and O_4, were tested using varying step sizes (δ) for the uncertain parameters. Smaller steps gradually increased the number of combinations and total computation time. Table 14 summarizes the tested configurations, showing the number of FE analyses executed and estimated computation times based on an average run of 6 seconds per model (using the hardware described in Section 3.3.3).

Table 14. Sampling step, number of combinations, and estimated time for scenarios O_1, O_2, O_3, and O_4, run through SAP.Py.

SAP.Py Id	δE_1 [MPa]	δE_2 [MPa]	δK_{rx} [%]	δK_{ry} [%]	Combinations	Estimated Time (hours)
O_1	1500	1500	100	100	900	1.30
O_2	800	800	10	10	7744	11

O_3	600	600	10	10	14641	22
O_4	150	80	10	10	1200000	2333

Considering that for running a single model, a time of about 6 seconds was required, assuming larger values of sampling steps, an estimated time of 1.30 hours was found for scenario O_1. On the other hand, by reducing the values of sampling steps (especially for δE_1 and δE_2), the number of combinations strongly increased, achieving an estimated time of 2333 hours for scenario O_4. Due to the computational effort estimated, the offline procedure was performed only for scenarios O_1, O_2 and O_3. The results, shown in Table 15, report the optimal parameters $\widehat{p_c^2}$, simulated frequencies $\widehat{v_s^{\tau 2}}$, and relative differences with respect to the experimental (real) frequencies $v_r^{\tau 2}$.

Table 15. *Results of deterministic MU via SAP.Py for scenarios* O_1, O_2 and O_3.

Scenario	Structural control parameters p_c^2				Target variables		Percentage differences (%)
	E_1[MPa]	E_2[MPa]	K_{rx}[kN*m/rad]	K_{ry}[kN*m/rad]	v_r^{τ}[Hz]	v_s^{τ}[Hz]	
O_1	32000	33500	1.0E+6	1.0E+6	3.9844	3.9916	-0,180
					4.5950	4.6052	-0,223
					4.6941	4.7127	-0,396
O_2	31400	33800	1.0E+6	1.0E+6	3.9844	3.9848	-0,009
					4.5950	4.5977	-0,059
					4.6941	4.7033	-0,196
O_3	31400	33800	1.0E+6	1.0E+6	3.9844	3.9848	-0,009
					4.5950	4.5977	-0,059
					4.6941	4.7033	-0,196

Scenarios O_2 and O_3 produced identical results, suggesting convergence at finer sampling resolutions. However, the computational effort required remained extremely high. The deterministic process proved reliable but inefficient for multi-parameter problems due to its combinatorial nature.

Concerning the probabilistic approach, a Bayesian MU was employed. According to Equations (6) and (7), the aim of the Bayesian MU is to find the posterior probability $p(\theta|\bar{d}, \mathcal{M})$, while maximizing the likelihood, $p(\bar{d}|\mathcal{M})$. Therefore, the first step was

the definition of the prior PDF, which according to (Ierimonti et al., 2020) can be of different nature, based on the available information on the uncertain parameters. For the case at hand, uniform distributions for p_u^2 were considered, materializing the so-called non-informative prior. The uniform distributions for the p_u^2 were set by assuming engineering extremes coincident with the minimum and maximum values above fixed (E_1, E_2 ranging from 29000 MPa to 35000 MPa, K_{rx}, K_{ry} ranging from $0.0\mathrm{E}+00$ kN*m/rad to $1.0\mathrm{E}+12$ kN*m/rad). Considering the assumption of non-informative prior, the value of the posterior $p(\theta|\bar{d},\mathcal{M})$ is basically equal to the value of the likelihood, $p(\bar{d}|\theta,\mathcal{M})$, which means that to maximize the posterior implies the maximization of the likelihood. In other words, this is the maximum likelihood estimation strategy, which can be written as:

$$\theta = \widehat{p_c^2} = \mathrm{argmax}\left\{p(\bar{d}|\theta,\mathcal{M})\right\} = \mathrm{argmin}\left\{-log\big(p(\bar{d}|\theta,\mathcal{M})\big)\right\} \qquad (20)$$

Using the likelihood as metric to find $\widehat{p_c^2}$ and given a size of the sampling, the procedure was applied as follows: (1) random selection of the values from the prior distributions; (2) run of eigenvalue analysis in the FEM software with the sampled values; (3) record the values for the main three frequencies; (4) estimate the matching between experimental and numerical data by maximizing the likelihood to find $\widehat{p_c^2}$.

The above approach was applied on the dummy structure, by considering different sizes of the samples to assess the convergence of the likelihood and the related standard deviation. As shown in Figure 37, convergence was achieved after approximately 16000 runs, corresponding to a computational time of roughly 27 hours, confirming the heavy processing demand.

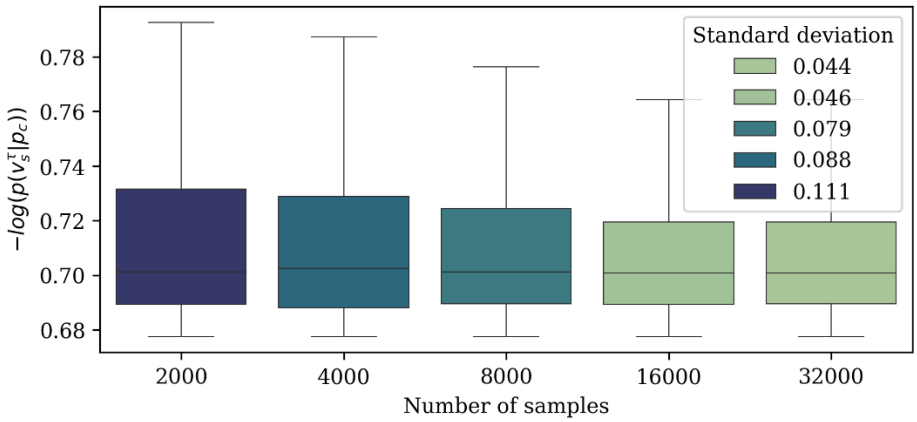


Figure 37. Variation of log-likelihood standard deviation with increasing sample size. Likelihood stabilization achieved for >16,000 runs.

The posterior distributions of each parameter and their comparison against prior distributions are reported in Figure 38. The results show that K_{rx} and K_{ry} hold larger influence on the structural response, with posterior distributions converging toward higher stiffness values close to those observed experimentally.

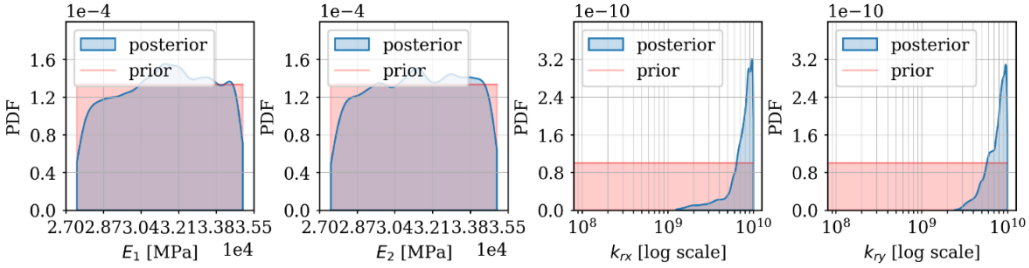


Figure 38. *Overlapping probability density functions (PDFs) of prior (red) and posterior (blue) distributions for uncertain parameters.*

The comparative evaluation employed two metrics capturing both absolute errors and relative parameter importance, the Root Mean Square Error (RMSE), a traditional metric assessing average deviation magnitude and the Weighted Absolute Percentage Error (WAPE), accounting for parameter magnitudes to provide normalized error insights. It is worth noting that the RMSE is the common metric to evaluate the difference between experimental and numerical data, and for this reason was considered. On the other hand, it does not account for the weight of each parameter, and to this end, the WAPE was also quantified. Concerning the probabilistic approach, the

values of $\widehat{p_c^2}$ and $\widehat{v_s^{\tau 2}}$ derived by the posterior distributions and corresponding to the maximum likelihood and the stabilized standard deviation were considered for comparison purpose. Results summarized in Table 16 disclose that the DRUM-Av1.0 significantly outperforms the deterministic approach in RMSE for the uncertain parameters $\widehat{p_c^2}$, indicating superior ability to locate the correct parameter space efficiently while reducing computational overhead. The results show that the value of RMSE by DRUM-A for uncertain parameters $\widehat{p_c^2}$ was strongly lower than the ones obtained for the two scenarios of automated deterministic MU, confirming the ability of the agent in finding the correct set up of the FEM parameters. Nevertheless, RMSE by the probabilistic approach was lower than the one by DRUM-Av1.0. Looking at the RMSE for the target variables, it results higher for DRUM-Av1.0 than the two comparative approaches. Observing instead the WAPE, both for structural control parameters and target variables, DRUM-Av1.0 presents lower values than the deterministic scenarios and higher values than the probabilistic approach. This outcome suggests that the most accurate approach is the Bayesian MU, without observing the information about time of analysis. In fact, DRUM-Av1.0 achieves the overall correct solution in a little less time than deterministic scenario O_2, and in much less time than probabilistic approach (about a third of the time). Overall, the proposed approach seems to be able to globally search for the correct solution, considering both input and output quantities in the problem investigated, with a reasonable time and computational effort

Table 16. *Comparison of DRUM-Av1.0 results with deterministic and probabilistic MU methods.*

Data	Real values (p_c^{T2} and $v_r^{\tau 2}$)	SAP.Py O_1 ($\widehat{p_c^2}$ and $\widehat{v_s^{\tau 2}}$)	SAP.Py O_2 ($\widehat{p_c^2}$ and $\widehat{v_s^{\tau 2}}$)	Bayesian ($\widehat{p_c^2}$ and $\widehat{v_s^{\tau 2}}$)	DRUM-Av1.0 ($\widehat{p_c^2}$ and $\widehat{v_s^{\tau 2}}$)
E_1[MPa]	31500	32000	31400	31658	31650
E_2[MPa]	31500	33500	33800	31526	31080
k_{rx}[kN*m/rad]	1.0 E+8.5	1.0E+6	1.0E+6	5.1E+9.0	1.0E+8.70
k_{ry}[kN*m/rad]	1.0 E+8.5	1.0E+6	1.0E+6	8.6E+9.0	1.0E+8.49
f_1[Hz]	3.9844	3.9916	3.9848	3.9805	3.9644
f_2[Hz]	4.5950	4.6052	4.5977	4.5901	4.5672
f_3[Hz]	4.6941	4.7127	4.7033	4.6889	4.6714
RMSE - p_c^2	---	1030.778	1151.088	80.068	222.991

RMSE - $v_r^{\tau 2}$	---	0.0128	0.0055	0.0047	0.0237
WAPE - p_c^2 [%]	---	3.975	3.816	0.296	0.905
WAPE -$v_r^{\tau 2}$[%]	---	0.271	0.093	0.105	0.531
Time [hours]	---	1.30	11	26.67	9

3.4 Real case study

In order to assess the efficiency of DRUM-Av1.0 for real structures, the proposed approach was tested on a real-life existing prestressed RC bridge. For privacy reasons, no registry information can be provided about the case study. The structure is characterized by three bays, for a total length of 87 m in longitudinal direction (the lateral bays present a length equal to 27 m, while the central bay presents a length of 33 m) and 13 m in transversal direction. The deck of each bay is characterized by four box-girder beams (height of 1.80 m) in the longitudinal direction and five lateral transverse beams in the perpendicular direction. The deck is placed on two frame piers (constituted by 3 piers of 1.30 m diameter and a pier cap of 1.5 m height) and two RC abutments, with an overall height of 6 m. Figure 39 reports some images of the case study bridge, showing the configuration of the deck and the disposition of the box-girder beams. The static scheme of the bridge is simply supported, but a continuous cast-in-place concrete slab, 30 cm thick, is placed on the beams. Two external expansion joints are placed in correspondence of the abutments.





Figure 39. Images of the case study bridge

3.4.1 Surrogate SHM and NM

For the considered bridge, two main experimental campaigns were available. The first one was oriented to define the mechanical parameters of materials. Of interest for this study is the mean value of the elastic moduli for three bays of the bridge. The second one was a monitoring campaign, to assess the dynamic behaviour of the bridge. The latter, schematized in Figure 40, was characterized by 18 monoaxial accelerometers positioned on the deck along the longitudinal development of the bridge, 2 biaxial accelerometers positioned on the two frame piers, one triaxial accelerometer positioned on one abutment. Of interest to characterize the dynamic behaviour of the deck are the monoaxial accelerometers (as reported in Figure 40), which revealed that the presence of the continuous slab varied the static scheme of the bridge, ensuring a certain continuity of the deck. Hence, a fixed connection between consecutive bays was detected.

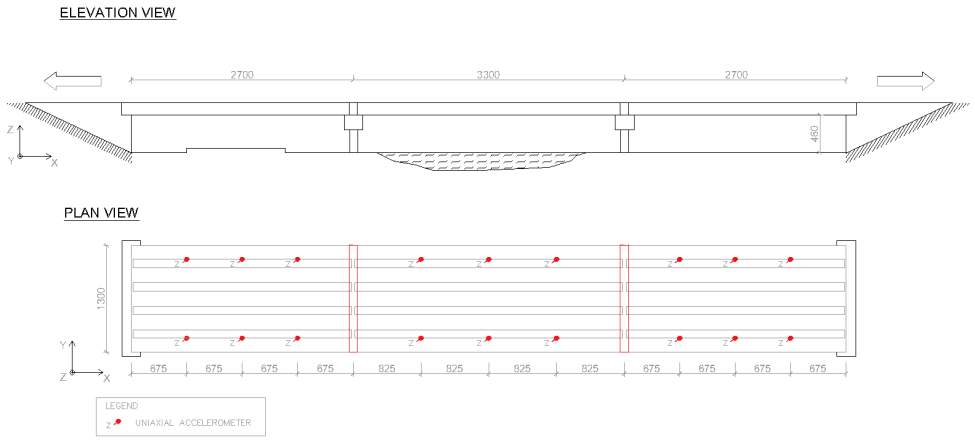


Figure 40 Plan and elevation view of the case study bridge, with the distribution of the uniaxial accelerometers along the longitudinal development of the bridge.

For the purpose of DRUM-A application, five target uncertain parameters, p_u^τ, were considered, i.e., the elastic moduli of the three bays (E_1, E_2, and E_3), and the degree of constraints between two consecutive bays (C_1 and C_2). The real values of p_u^τ are reported in Table 17, for which it is worth specifying that C_1 and C_2 are Boolean values, i.e., fix assumed equal to 1 and no-fix assumed equal to 0.

Table 17. Values of p_u^τ for the case study bridge

E_1 [MPa]	E_2 [MPa]	E_3 [MPa]	C_1	C_2
34000	33500	33450	1	1

According to the proposed procedure, the first steps consisted of: (a) performing OMA to derive the main vibration modes and the related frequencies; (b) defining an NM to update. Concerning point (a), OMA was performed in the software ARTeMIS (Structural Vibration Solutions, 2023), and the values of the first three main frequencies, i.e., $v_r^\tau = [f_1^{SHM}, f_2^{SHM}, f_3^{SHM}]$, were derived (see Table 18). It is worth specifying that the use of three main frequencies is a simplification for the case at hand, due to the presence of higher modes. On the other hand, the aim of this application is to assess the efficiency of the proposed procedure for any value and quantity of v_r^τ. Regarding

the NM, SAP2000 (Computers and Structures, Inc., 2024) was used. Longitudinal and transversal beams of decks were simulated through frame elements, as well as the frame piers, which were fixed at the base. External sides of lateral bays were schematized as simply supported, to simulate the expansion joints and the constraint on the abutments. Regarding masses, self-weight, dead (3 kN/m^2 for pavements, 1 kN/m^2 for parapet, and 2.5 kN/m^2 for barriers) and live (uniform and tandem) loads were simulated and combined, according to the reference building italian code (MIT, 2018). To distribute loads, a null shell area, meshed according to the beams grillage disposition, was set for each bay, and to simulate the constraint between consecutive bays (i.e., C_1 and C_2), equal-degree-of-freedoms command was used. Figure 41 reports a sketch of the developed models, both for OMA (Structural Vibration Solutions, 2023) (left) and for MU (Computers and Structures, Inc., 2024) (right).

Table 18. Values of the first three vibration modes for the case study structure

Vibration mode	Frequency [Hz]
1° mode	$f_1^{SHM} = 4.7429$
2° mode	$f_2^{SHM} = 5.7443$
3° mode	$f_3^{SHM} = 6.6051$

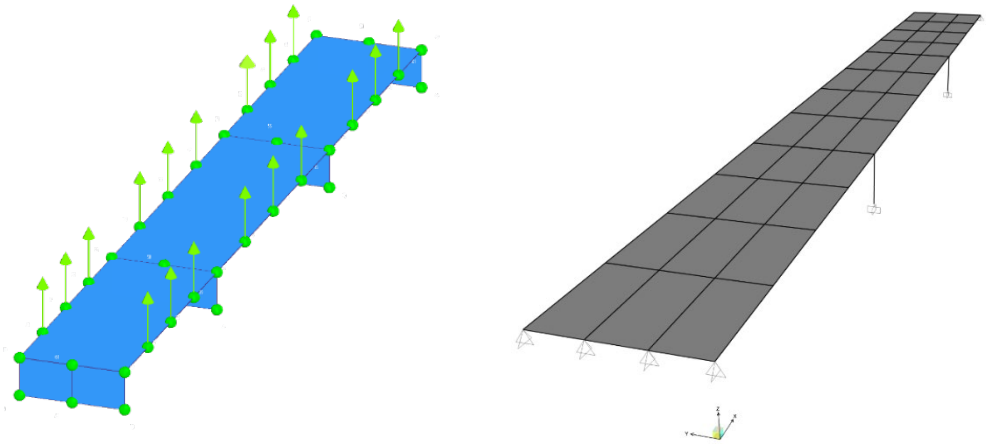


Figure 41. Models for (left) OMA, developed in Artemis (Structural Vibration Solutions, 2023); (right) MU, developed in SAP2000 (Computers and Structures, Inc., 2024).

3.4.2 Definition of DRL-engine

The main features of the DRUM-Av1.0 system remain the same as those described in the previous case study (see Section 3.3.2). As for the dummy structure, the training of the DRL-engine was set, by assuming large lower and upper limits for E_1, E_2, and E_3 equal to 29000 MPa and 37000 MPa, respectively. After, the action space $\mathbf{A} = \{\mathbf{\Delta E_1}, \mathbf{\Delta E_2}, \mathbf{\Delta E_3}, \mathbf{\Delta C_1}, \mathbf{\Delta C_2}\}$ was defined, by imposing multi-discrete actions for the five structural control parameters, as reported in Table 19. Regarding reward strategy, the functions defined in Equations (17)-(19) were employed.

Table 19. Multi-discrete action space available for the agent for the case study.

Set of actions on the p_u	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5
$\boldsymbol{\Delta E_1}$	-150	-80	0	80	150
$\boldsymbol{\Delta E_2}$	-150	-80	0	80	150
$\boldsymbol{\Delta E_3}$	-150	-80	0	80	150
$\boldsymbol{\Delta C_1}$	-1	+1	/	/	/
$\boldsymbol{\Delta C_2}$	-1	+1	/	/	/

3.4.3 Training and test of the agent

The training phase was set by assuming a number of 500 episodes and by defining two termination rules: (i) the agent reaches a number of timesteps equal to 100; (ii) $\sigma_{(t)} < \sigma_{(t,end)}$, where $\sigma_{(t,end)} = 0.03$. Figure 42 reports the results of the training, showing the values of the cumulative reward for each episode in grey and the related trend through the simple moving average cumulative reward in orange. The results show that after the first 200 episodes, the agent was able to select the best strategy to achieve the result (cumulative reward around a value of -20), considering that the best values are obtained for reward values ranging from 0 to -40.

Still, Figure 43 shows the values of $\sigma_{(t)}$ (Equation 16) recorded at the end of each episode of the training. During the first 110 episodes, the agent performed the actions with high value of $\sigma_{(t)}$ showing the initial instability in the learning. After, the

agent increases the capacity of assuming the correct policy and progressively improves its behaviour until achieving a stable value of the cumulative reward, i.e., $\sigma_{(t)} < 0.03$. For the case at hand, the duration of the training phase, using the same machine defined in Section 3.3.3, was quantified equal to 21 hours.

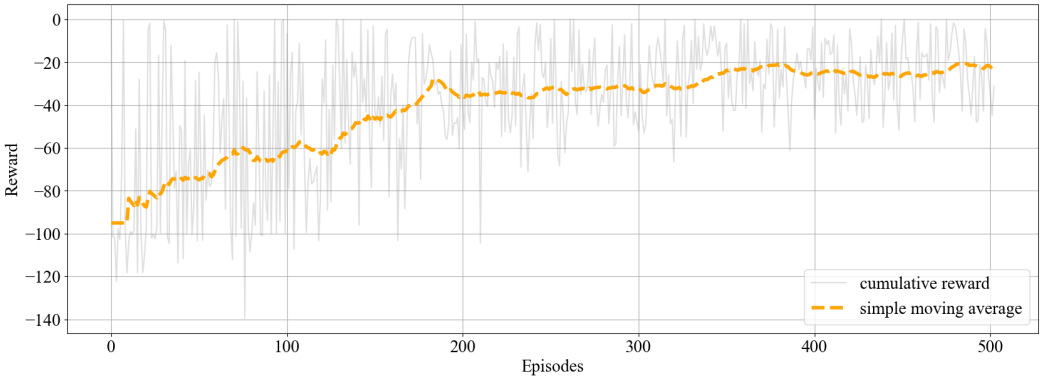


Figure 42. Cumulative reward trend during the training phase for the case study.

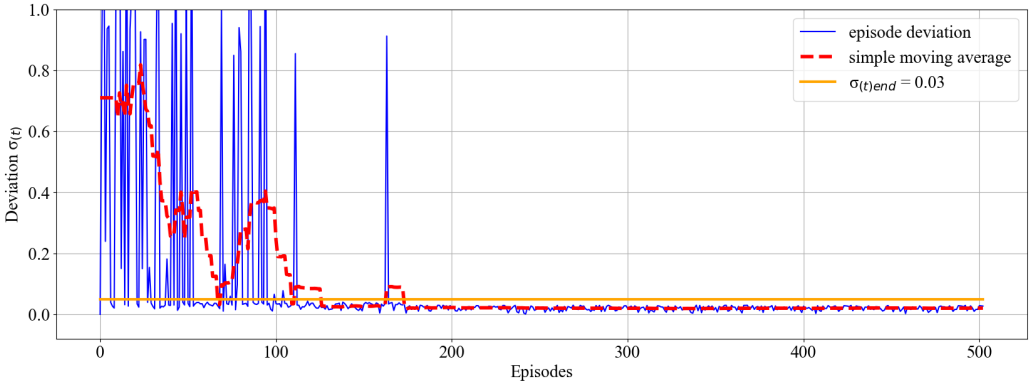


Figure 43. Trend of sigma $\sigma_{(t)}$ function and simple mobile average during the training phase at the end of each episode for the case study

After training, the test and validation phases were performed, to determine the values of $\widehat{p_c}$. To difference of the dummy structure, considering the impossibility of changing p_c^{τ}, two main variations were proposed for excluding overfitting phenomena:

1. Random definition of $p_{c(t0)}$ for each episode.
2. Modification of the terminate rule imposed on the tolerance $\sigma_{(t)}$, assumed equal to 0.02.

The obtained results can be observed in Figure 44 for a total of 100 episodes, in which both trends of cumulative reward and deviation $\sigma_{(t)}$ are shown. In particular, the cumulative reward for each episode, Figure 44(a), was comprised between -5 and -25, showing a certain stability of the agent to converge towards the solution by taking the lower number of timesteps. Regarding the deviation $\sigma_{(t)}$, Figure 44(b), it is evident that the recorded values are always lower than the imposed threshold, to confirm the accuracy of the agent in each episode of the test and validation process and the good stability of the policy found.

After test, model selection and performance evaluation were performed. In particular, the values of $\widehat{p_c}$ were recorded for each episode and the statistical distribution of each parameter were computed. The graph of the continuous distributions for E_1, E_2, and E_3 and frequencies of C_1 and C_2 are reported in Figure 45. As can be observed, C_1 and C_2 were almost always assumed as fix by the agent, according to the experimental evidence. Concerning the values of E_1, E_2, and E_3, the mode values (marked with red dashed line on the graphs in Figure 45) were identified, showing low values of the dispersions, with CV in a percentage of less than 5%. Similar outcomes were observed by comparing the obtained $\widehat{p_c}$ and the real values p_c^{τ}. In fact, as shown in Table 20, a percentage variation lower than 5% for each parameter was denoted, confirming once again the high accuracy of the agent to find the correct solution.

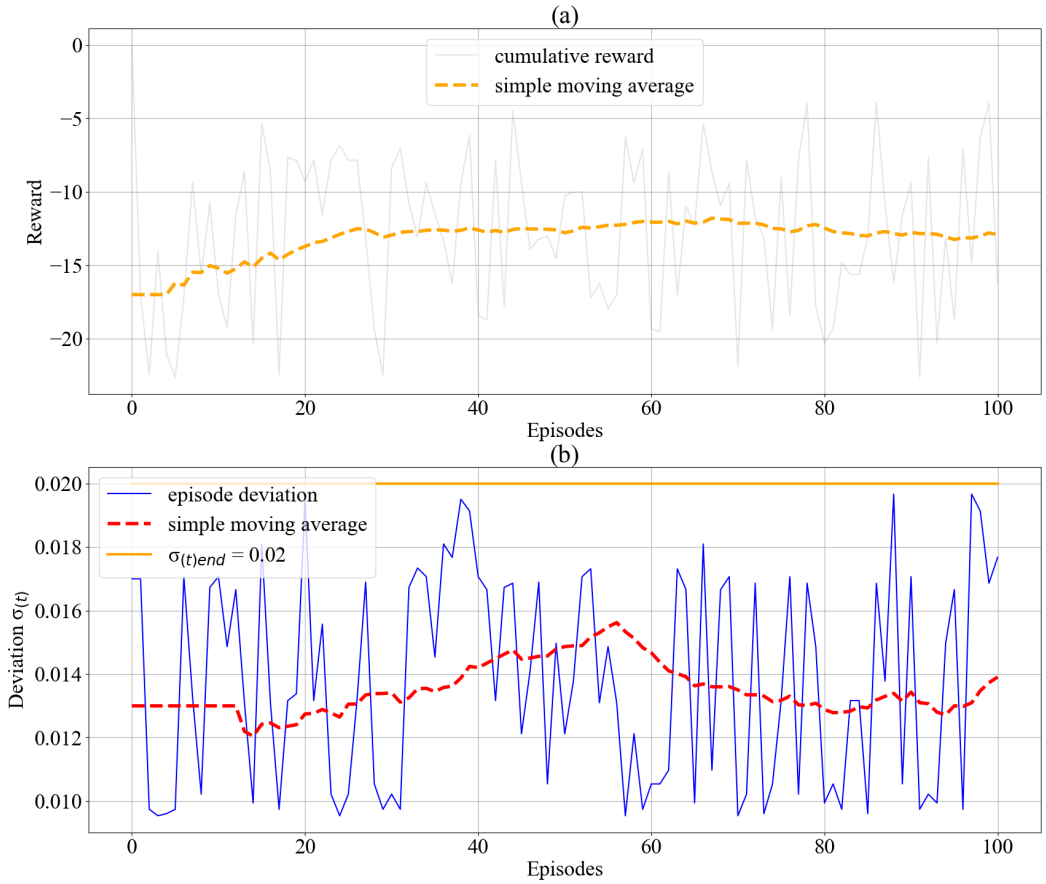


Figure 44. Results of the DRUM-A test and validation phase for the case study: a) cumulative reward trend at the end of each episode and the average cumulative reward during the test phase; b) the trend of the deviation $\sigma_{(t)}$ achieved at the end of each episode and the average deviation during the test phase.

Table 20 Mode values of the uncertain parameter $\widehat{p_u}$ obtained by DRUM-A for the case study and comparison with real values.

Parameter	E_1 [Mpa]	E_2 [Mpa]	E_3 [Mpa]	C_1	C_2
Real values p_u^τ	34000	33500	33450	fix	fix
DRUM-Av1.0 $\widehat{p_u}$	33780	33230	33270	fix	fix
Percentage variation (%)	0.647	0.806	0.538	0.0	0.0

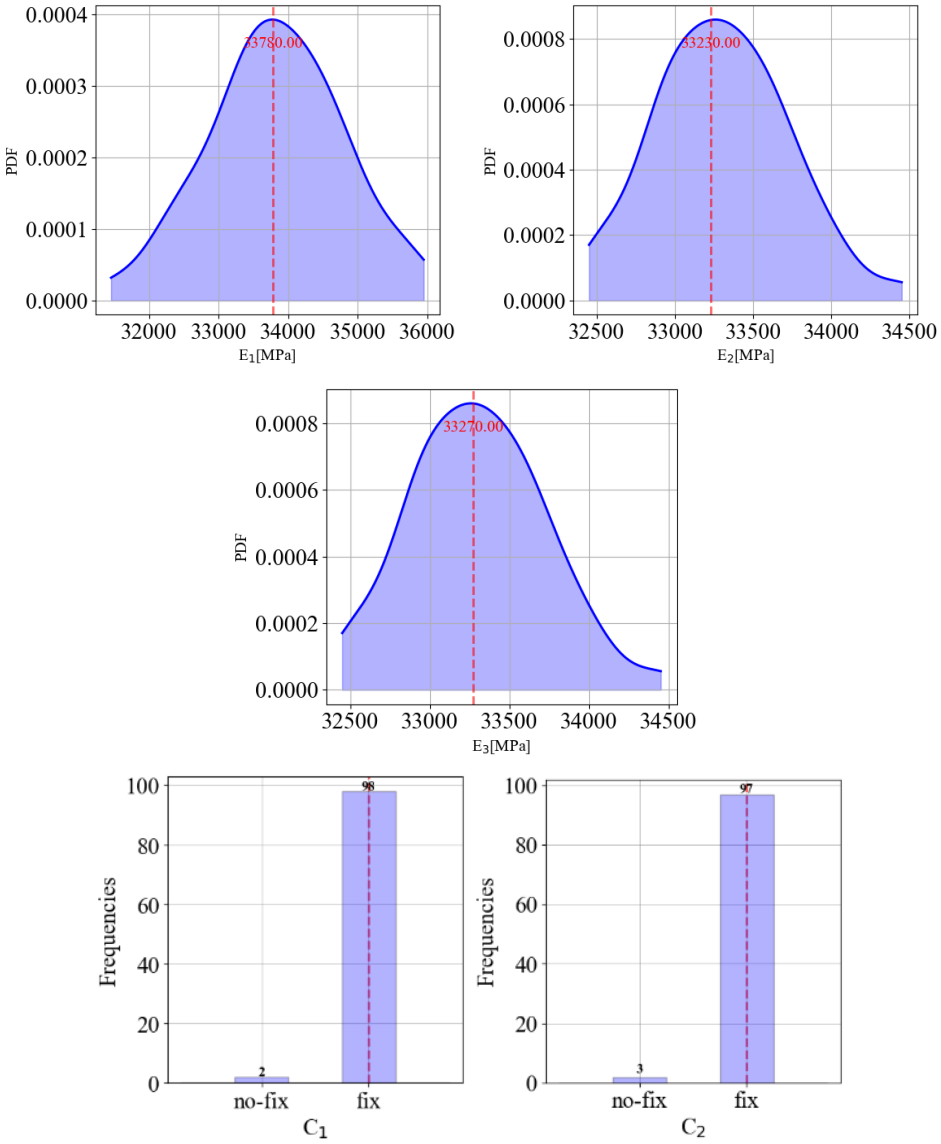


Figure 45. Statistical distributions of E_1, E_2, and E_3 (continuous) and C_1 and C_2 (discrete). The estimated values of CV were equal to 2.85%, 1.27%, and 1.30%, respectively for E_1, E_2, and E_3.

3.5 Discussion

This section presents DRUM-Av1.0, an innovative method based on intelligent agents that uses Deep Reinforcement Learning for the numerical updating of structural models. DRUM-Av1.0 was designed to simulate the iterative logic of manual model

updating, replacing the human analyst with an intelligent agent that autonomously tunes model parameters within a numerical software environment. The agent's goal is to minimise discrepancies between numerical simulations and experimental measurements through iterative interaction and learning. The method follows five fundamental steps, starting with a SHM campaign to characterise the actual structural behaviour and define the unknown uncertain parameters and real target variables. The second step is the construction of a representative numerical model (NM) that incorporates uncertain parameters and produces simulated target variables. Than the design of the DRL framework is performed, in which the agent learns to modulate uncertain parameters by observing the states and rewards generated based on the accuracy of the simulations. After the training of the agent over multiple episodes to enable rapid convergence towards accurate solutions, the final optimal model is selected from statistically analysed solutions, emphasising the values of parameters that occur frequently.

DRUM-Av1.0 was validated on a dummy RC structure with four uncertain parameters and three target variables (modal frequencies), successfully achieving precise and stable solutions. Comparisons with traditional deterministic and probabilistic model updating approaches showcased DRUM-Av1.0's capacity to balance accuracy and computational efficiency. Unlike deterministic approaches requiring exhaustive search, DRUM-Av1.0 efficiently navigates the parameter space without exploring all combinations. Compared to probabilistic methods, it significantly reduces computational time while delivering near-comparable accuracy. Key advantages of DRUM-Av1.0 include scalability to complex structures with many uncertain parameters, autonomous decision-making by the agent eliminating the need for a-priori assumptions or selections of parameter values, and an efficient, optimized workflow producing accurate parameter estimations within a reasonable computational budget.

However, some current limitations are worth discussing. The main limitation of DRUM-Av1.0 lies in its computational intensity, which derives from the need to perform repeated analyses in FOM during the training process. Each iteration requires a complete solution of the structural equilibrium equations, making the process time-consuming for large-scale or highly refined finite element models. Although the training cost is

warranted by the accuracy and self-adaptive nature of the method, it nevertheless highlights a critical trade-off between model fidelity and computational feasibility. To overcome this limitation, the next section introduces a significant update to the framework based on the integration of reduced-order models as surrogates for FEM simulations. ROMs can replicate the dynamic behaviour of the system at a significantly lower computational cost, enabling faster agent training, real-time interaction, and potential online model updating powered by continuous monitoring data. In conclusion, DRUM-Av1.0 represents a significant step forward in updating structural models. It demonstrates how artificial intelligence, and DRL in particular, can automate, optimise and accelerate what was once a highly manual process. The framework has the potential to evolve into a powerful tool for real-time digital twin development and adaptive SHM.

4. Integration of ROM in the DRL framework

To overcome the limitations of DRUM-Av1.0, described in Section 3, the use of a less computationally demanding model was implemented, while maintaining the accuracy and reliability of the proposed framework. This chapter presents an innovative update to the methodology, which integrates Reduced Order Modelling within the DRL framework to address MU problems in the structural field. The main objective is to combine the high fidelity and accuracy of traditional Full Order Models, as seen in Section 3, with the computational efficiency of ROM, thus allowing accurate and efficient updating of the uncertain parameters of the monitored structures. As discussed in introductory Section 1.2, the combined use of FOM and ROM represents a balanced approach in MU: FOMs provide a high level of detail and fidelity in simulating structural behaviour, while ROMs allow for a drastic reduction in computation time while maintaining accurate global results. This synergy allows ROM to be used for efficient global calibration and FOM for local insights, thus ensuring an optimal compromise between accuracy and computational costs.

The initial section introduces the fundamental concepts underlying ROM and its synergy with DRL, laying the theoretical foundations for this integration. This is followed by a detailed description of the procedure for creating the dataset from the FOM and subsequently defining the ROM through parametric reduction techniques such as Proper Orthogonal Decomposition (POD). Attention is then focused on the evolution of the DRUM algorithm, with the updated version DRUM-Av2.0, which represents the core of the DRL framework enhanced by ROM. This version introduces substantial improvements in terms of efficiency, while maintaining solid standards of accuracy. The Section continues with the practical application of the methodology to two progressively more complex case studies, a single-storey reinforced concrete frame and a two-storey frame with irregularities, illustrating the results obtained, the comparative analysis of the algorithm versions and the evaluation of overall computational efficiency. Finally, a critical discussion summarises the strengths and potential limitations of the proposed framework, highlighting future developments for broader applicability and optimisation of the method.

4.1 Basics of Reduced Order Modelling

A ROM can be defined as a computational simplification of a high-fidelity model, able to efficiently capture its dynamic behaviour with fewer degree of freedoms (DOFs) while retaining accuracy. To create a ROM from a FOM, the equation of motion for a linear structural dynamics problem can be used (note that the terms in Equation (21) are reported without notation for multi degree of freedoms for the sake of simplicity):

$$M\ddot{u}(t) + C\dot{u}(t) + Ku(t) = f(t) \tag{21}$$

where $M, C, K \in \mathbb{R}^{N\times N}$ are the mass, damping, and stiffness matrices of the FOM, respectively; $u(t), \dot{u}(t), \ddot{u}(t) \in \mathbb{R}^N$ are the vectors of nodal displacements, pseudo-velocities, and pseudo-accelerations, respectively; $f(t)$ represents the vector of external forces. Considering that the dimension N of the FOM system is typically very large, the aim of the model reduction is to find an approximation of $u(t)$ in a space $r \ll N$. The solution can be represented as a linear combination of a set of basis vectors ϕ_i, which constitute the columns of a projection matrix $\Phi \in \mathbb{R}^{N\times r}$:

$$u(t) \approx \Phi u_r(t) \tag{22}$$

where $u_r(t) \in \mathbb{R}^r$ is the vector of reduced or generalized coordinates. By applying Galerkin projection, i.e., by substituting the approximation in Equation (22) in the equation of motion in Equation (21), and pre-multiplying all left terms by Φ^T, the reduced system can be expressed as:

$$\Phi^T M\Phi\ddot{u}(t) + \Phi^T C\Phi\dot{u}(t) + \Phi^T K\Phi u(t) = \Phi^T f(t) \tag{23}$$

The latter can be re-edited as

$$M_r\ddot{u}_r(t) + C_r\dot{u}_r(t) + K_r u_r(t) = f_r(t) \tag{24}$$

where $M_r, C_r, K_r \in \mathbb{R}^{r\times r}$ are the reduced order matrices and $f_r \in \mathbb{R}^r$ is the vector of reduced order forces, presenting significantly smaller size than the ones in Equation 21). While order reduction improves efficiency, the accuracy of the ROM depends on

the choice of the reduced base, i.e., the projection matrix Φ, which enables the ROM to capture the fundamental dynamics characteristics. Among the most used techniques for performing the projection, Proper Orthogonal Decomposition stands out for its optimal approximation properties and strong relationship with Singular Value Decomposition. POD is a powerful data-driven technique for extracting an optimal orthonormal basis from a set of experimental or numerical data (Kerschen et al., 2005). The application of the POD method preliminarily requires the collection of c snapshots of the FOM solution $u(t_j)$ by the FOM, computed at different points in time t_j or for different combinations *h* of the characteristic parameter values, with $h = 1, \dots, c$. From the formal point of view, $u(t_j)$ can be also defined as $u(h)$. The solutions $u(h)$ are assembled as columns of a matrix $S_m \in \mathbb{R}^{N \times c}$

$$S_m = [u(1), \dots, u(h), \dots, u(c)] \tag{25}$$

By applying the SVD, the best low-rank approximation of the matrix is obtained, which is expressed as

$$S_m = L \Sigma V^T \tag{26}$$

where $L \in \mathbb{R}^{N \times N}$ is an orthogonal matrix whose columns L_i (L_i is the column *i* of matrix L, termed as POD mode), called left singular vectors, form a basis for the solution space; $\Sigma \in \mathbb{R}^{N \times c}$ is a diagonal matrix containing the singular values $v_1 \geq v_2 \geq \cdots \geq 0$ in descending order; $V \in \mathbb{R}^{c \times c}$ is an orthogonal matrix whose columns are the right singular vectors. The optimal POD basis Φ consists of the first r singular vectors of S_m, with r equal to the number of principal orthonormal components (L in its reduced version can be noted as L_r).

$$\Phi = [L_1, \dots, L_i \dots, L_r] \tag{27}$$

4.2 Combining use of the FOM and ROM

The combination of the FOM and the ROM is an effective strategy for balancing accuracy and computational efficiency in structural analysis processes. The FOM, characterised by a high level of detail and a complete description of structural characteristics, allows local behaviours and complex phenomena to be captured accurately, but at the cost of high computation times. In contrast, the ROM provides a simplified representation of the system, drastically reducing the number of degrees of freedom and therefore computational complexity without significantly compromising overall accuracy. In the context of the proposed framework, the FOM is used as a reference model for generating a large initial dataset, from which essential information is extracted to build the ROM using reduction techniques such as Proper Orthogonal Decomposition. In this way, the ROM becomes the operating environment for the DRL agent, which interacts with a computationally light model that is sufficiently faithful to physical reality.

This synergy allows for efficient and accurate calibration of uncertain parameters in a much shorter time than when using the FOM directly, thus offering an optimal compromise for complex scenarios where simulation repeatability and response speed are critical. Furthermore, in cases where local analysis or detailed control is required, the FOM can be called upon for specific checks, thus maintaining a dynamic balance between detail and computational speed. This combined strategy enhances the potential of both models, extending the applicability of structural updating techniques to contexts with high dimensionality and uncertainty, without sacrificing either the quality or efficiency of the process.

4.2.1 Definition of the FOM and dataset creation

To achieve high-fidelity and representativeness of the real structural system, different numerical modelling strategies can be employed, e.g., finite element method, boundary element method, discrete element method (Bathe, 2008; Aliabadi, 2020; Bi ćanić, 2007). Of interest for the scopes of the paper is to identify the uncertain structural features, which are indicated as p_u and represent the parameters to update. The FOM generation is crucial for the proposed procedure, because it covers a twofold role: (a)

the FOM acts as reference model for defining the dynamic response of the structure; (b) the FOM represents the main tool to generate a dataset, $\boldsymbol{D}$, necessary for the definition of the ROM (see Section 4.2.2). To this scope, by using an automated procedure like the one in (Bruno et al., 2024), a number of c eigenvalue analyses can be performed on the FOM characterized by different c combinations of values associated with the uncertain parameters, i.e., $p_u(h)$ with $h = 1, \dots, c$. The size of c is established according to the level of accuracy established by the analyst for making the ROM. The higher the size of $\boldsymbol{D}$, the higher the accuracy of the ROM. From the formal point of view, the dataset can be characterized as

$$\boldsymbol{D} = (d_1, \dots, d_h, \dots, d_c)^T \; with \; h = 1, \dots, c \tag{28}$$

where for each generic combination h, the row vector d_h, component of $\boldsymbol{D}$, can be written as a vector containing the vector of uncertain parameters, $p_u(h)$, the vector of the simulated target variables, $v_{FOM}^{\tau}(h)$, the mass and the stiffness matrices of the FOM, indicated as $M(h)$ and $K(h)$. If only natural frequencies are considered as simulated target variables, $v_{FOM}^{\tau}(h)$, as for the case at hand, d_h formally also contains the matrix of eigenvectors (defined as the matrix containing all eigenvectors for each natural frequency, composed by a sum of columns), $u(h)$. Therefore, d_h can be expressed as

$$d_h = [p_u(h), v_{FOM}^{\tau}(h), u(h), M(h), K(h)] \tag{29}$$

All terms in Equation (29) can be explicitly defined according to the size of the problem at hand. As above stated, keeping separated eigenvalues in $v_{FOM}^{\tau}(h)$ and eigenvectors in $u(h)$, and assigning a generic size n to $p_u(h)$, l to $v_{FOM}^{\tau}(h)$, and the total number of DOFs, n_{DOF} to $M(h)$, $K(h)$ and $u(h)$, the terms of d_h can be expressed as

$$p_u(h) = \left[p_u^1(h), \dots, p_u^j(h), \dots, p_u^n(h)\right] \; with \; j = 1, \dots, n \tag{30}$$

$$v_{FOM}^{\tau}(h) = \left[v_{FOM}^{\tau 1}(h), \dots, v_{FOM}^{\tau k}(h), \dots, v_{FOM}^{\tau l}(h)\right] with\ k = 1, \dots, l \tag{31}$$

$$\begin{cases} u(h) = u^{x,y}(h) \\ M(h) = M^{x,y}(h) \\ K(h) = K^{x,y}(h) \end{cases} with\ x = 1, \dots, n_{DOF};\ \ y = 1, \dots, n_{DOF} \tag{32}$$

It is worth observing the matrix of damping coefficients is not accounted for in the proposed procedure, but it can be easily employed by adding another term to d_h.

4.2.2 Definition of the ROM

After defining the FOM, the order reduction can be processed, where the simulated target variables can be defined as v_{ROM}^{τ}. The success of the ROM stands in the definition of a model able to minimize the difference between v_{FOM}^{τ} and v_{ROM}^{τ}. If the two sets of values coincide, the MU process can be directly performed on the ROM, without losing accuracy and saving computational time. With this goal in mind, starting from the dataset $\boldsymbol{D}$, the ROM can be created by employing the POD technique as follows:

- Creation of the snapshot matrix $\boldsymbol{S_m}$, extracted from the dataset $\boldsymbol{D}$ as described in Equation (25), for all combination c. In this way, all values of $u(h)$ are accounted for.
- Computation of the reduced basis Φ, by consecutively applying SVD and truncation techniques, as expressed in Equations (26) and (27). The truncation depends on the level of order reduction to achieve, since this operation allows to obtain the DOFs considered in the ROM and the related influence on the structural response.
- Parameterization of M and K matrices (i.e., the ones by FOM) through a sensitivity analysis to partial derivatives (Khodaparast et al., 2008). Considering n uncertain parameters to update, the above matrices can be approximated using a first-order Taylor expansion around a reference condition $p_u(0)$. For the stiffness matrix K, the technique is developed as follows (the same procedure is applied to the mass matrix M):

$$K(p_u(h)) \approx K(p_u(0)) + \sum_{j=1}^{n} \left.\frac{\partial K}{\partial p_u^j(h)}\right|_{p_u(0)} (p_u^j(h) - p_u^j(0)) \tag{33}$$

where $K(p_u(h))$ is the stiffness matrix evaluated for the combination of uncertain parameters $p_u(h)$ related to the scenario h; $K(p_u(0))$ is the stiffness matrix for the reference values of the parameters $p_u(0) = [p_u^1(0), \dots, p_u^j(0), \dots, p_u^n(0)]$; $\frac{\partial K}{\partial p_u^j(h)}$ is the sensitivity matrix, i.e. the partial derivative of the stiffness matrix with respect to the parameter $p_u^j(h)$ in the point of expansion; $(p_u^j(h) - p_u^j(0))$ is the parametric variation of the uncertain parameter with respect to the reference. The reference condition and sensitivity matrix are computed by using the dataset $\boldsymbol{D}$ created by FOM.

- Definition of the ROM, by projecting mass M and stiffness K matrices into the reduced space derived from the POD basis, in order to obtain M_r and K_r. Equation (34) reports the definition of the reduced size matrices:

$$M_r = \Phi^T M \Phi;\ K_r = \Phi^T \mathrm{K} \Phi \tag{34}$$

- Validation of the ROM, by running eigenvalue analysis for obtaining the simulated target variables v_{ROM}^{τ}. Under the same conditions, i.e., FOM and ROM not updated but characterized by the same values for p_u, the ROM (with the selected reduced base Φ) is efficient if the differences between v_{ROM}^{τ} and v_{FOM}^{τ} are negligible according to a tolerance selected by the analyst (defined case-by-case, see for example Section 4.4 and Section 4.5). It is worth specifying that if the above differences are not negligible, the entire process must be repeated by implementing two modifications (ore one of these): (a) higher discretization of the values of the parameters assumed in the initial dataset $\boldsymbol{D}$ (see Equation (29)); (b) variation of the size for the reduced base Φ.

4.3 DRUM-Av2.0: algorithm and difference

This Section describes the operation of the DRUM-Av2.0 algorithm, an improved evolution of the previous DRUM-Av1.0, developed for the parametric calibration of structural numerical models using DRL integrated with ROM. The general architecture is outlined in Figure 46.

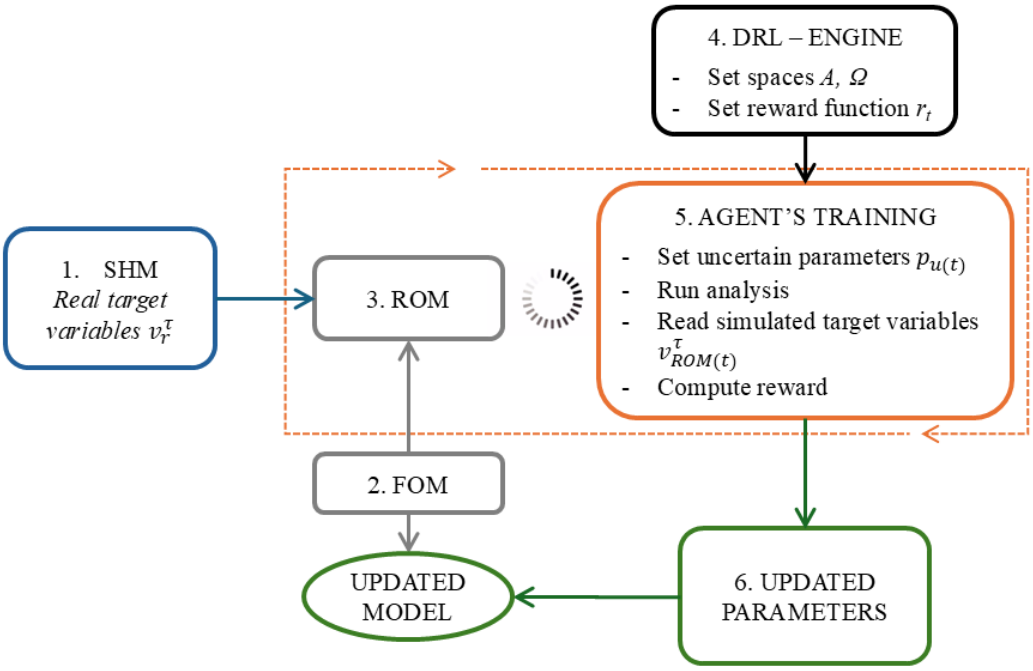


Figure 46. General framework of the DRUM-Av2.0

4.3.1 SHM campaign

The first step of DRUM-Av2.0 is the SHM campaign, aimed to collect experimental data to characterize the dynamic behaviour of the real structure, like to the previous version. The post-processing of onsite measurements enables the identification of the real target variables, v_r^{τ}, i.e., natural frequencies and modal shapes (Fuentes et al., 2020), which drive the MU process (for the definition of the proposed procedure, the notations in bold represent vectors or matrices). To obtain v_r^{τ}, experimental data should be processed through OMA techniques (e.g., in the time domain or in the frequency domain) (Brincker et al., 2001; Rainieri & Fabbrocino, 2014). At this stage, in order to characterize the dynamic behaviour of the monitored structure, it is essential an aware design of the monitoring system. Type of sensors, monitoring points, and acquisition duration are aspects depending on the type of structure, the number of uncertain parameters, the parameters to monitor, the costs of the experimental campaign. To summarize, the real target variables $\boldsymbol{v_r^{\tau}}$ represent the output of this first phase.

4.3.2 Definition of the FOM and the ROM

After identifying the target variables v_r^τ, the definition of a high-fidelity FOM is required. In this phase, the definition of geometry, material properties, external restraints, internal constraints, and loads is processed according to engineering assumption by the analyst, even though the real values of the parameters are not really known. For the creation of the FOM and ROM, see Section 4.2, which illustrates the integration of the ROM at the FOM.

4.3.3 Implementation of DRL engine

After defining ROM, it is integrated into the DRL framework, which requires the definition of:

- The environment, the space of states $\boldsymbol{S}$ and the space of observations $\boldsymbol{\Omega}$. For the case at hand, the space is fully observable and the environment is the ROM.
- The space of actions $\boldsymbol{A}$, according to the possible variations of the uncertain parameters.
- The reward function R, to define the error between the real target variables v_r^τ and the simulated target variables v_{ROM}^τ.
- The agent's learning strategy.

The goal of the agent is to interact with the ROM at each timestep t according to the learning strategy to maximize its cumulative reward. For each timestep t, the agent receives the vector of observations $\omega_{(t)} = \left\{v_{ROM(t)}^\tau, p_{u(t)}\right\}$, where $v_{ROM(t)}^\tau = \left[v_{ROM(t)}^{\tau 1}, \dots, v_{ROM(t)}^{\tau k}, \dots, v_{ROM(t)}^{\tau l}\right]$ with $k = 1, \dots, l$ represents the vector of target variables obtained as output by the analyses and $p_{u(t)} = \left[p_{u(t)}^1, \dots, p_{u(t)}^j, \dots, p_{u(t)}^n\right]$ with $j = 1, \dots, n$ represents the uncertain parameters. Subsequently, the agent applies the actions $a_{(t)} = a\left(p_{u(t)}\right) = \left[a(p_{u(t)}^1), \dots, a(p_{u(t)}^j), \dots, a(p_{u(t)}^n)\right]$ to individual parameters $p_{u(t)}$ of the ROM to modify their values. The process is organized in episodes e, each one characterized by all timesteps t, i.e., the iterations. The latter are interrupted when the output values $v_{ROM(t)}^\tau$ are closed to v_r^τ, according to certain threshold fixed by the analyst (i.e., the achievement of the solution of the MU problem). Alternatively, if after all iterations in the episode e the agent does not find a solution, the convergence

is not achieved and then, the process is stopped. The interactions between the agent and the environment are driven by a policy function $\pi(s)$, learned progressively by the agent, which updates continuously during episodes. The learning strategy is based on the reward value $r_{(t)}$, computed according to the reward function $r_{(t)} = R(v^{\tau}_{ROM(t)}, p_{u(t)}, v^{\tau}_{r})$. The aim of the reward (and of its definition) is lead the agent to learn a policy strategy that minimizes the error between $v^{\tau}_{ROM(t)}$ and v^{τ}_{r}, and that penalizes the agent when deviations or ineffective actions are carried out. In the proposed procedure, the agent's learning strategy is an action-value function $Q^{\pi}(s, a)$, and considering the problem as continuous and high dimensional, an Actor-Critic approach is employed. In detail, a Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm (Fujimoto et al., 2018) is adopted, which represents an advanced evolution of Actor-Critic frameworks, because it operates in off-policy mode and it is specifically designed for control problems with continuous actions.

4.3.4 Agent's training phase

Given the DRL engine, the training phase can be performed, by fixing a certain number of episodes. For each episode e, the agent starts from a random initial condition for uncertain parameters $p_{u(t0)}$, explores the options available in the state, performs actions, gains a reward and in the end of each episode obtains the values of uncertain parameters, p_u, by gradually updating the policy. If the agent repeatedly obtains suboptimal values for the uncertain parameters $\widehat{p_u}$ for multiple consecutive episodes, the agent behaves stably and has learned the optimal policy $\pi^*(s)$. In this case, the values of $v^{\tau}_{ROM(t)}$ are sufficiently closed to $\boldsymbol{v^{\tau}_{r}}$, according to a threshold defined in terms of a tolerance value, σ_{end} (the subscript *end* indicates that the episode terminates). It is worth specifying that the training phase is ruled by three key aspects established by the analyst: (a) the maximum number of iterations or timesteps t that the agent can perform in each individual episode e to find the solution; (b) the tolerance value σ_{end} defining the achievement of the solution; (c) the maximum number of iterations or timesteps t that the agent can perform to learn the optimal policy. The number

of episodes is not fixed a-priori, because it is a consequence of the number of iterations or timesteps t and of the training phase results.

Algorithm 3 shows the DRUM-Av2.0 procedure. From row 13 to row 32, the training process is shown. In detail, rows 19 to 23 illustrate the iterative process of interaction between the agent and the ROM environment, in which analyses are performed to achieve the desired results. In row 26, the conditions for ending the agent's training are reported, and are assessed at each timestep t in the cycle.

Algorithm 3. DRUM-Av2.0 code and definition of the procedure in steps.

Algorithm of DRUM-Av2.0		
1	***Output SHM**: read real target variables* $\boldsymbol{v_r^\tau}$	***← STEP 1***
2	*Create FOM and generate the dataset* $\boldsymbol{D}$	***← STEP 2***
3	***Create a ROM***	***← STEP 3***
4	*Set matrix of the snapshot* $\boldsymbol{S_m}$	
5	*Compute the base of reduction* $\boldsymbol{\Phi}$	
6	*Compute the matrix of sensitivity* $\partial \boldsymbol{K} / \partial p_u^j(h)$	
7	*Start DRL-ENGINE algorithm*	***← STEP 4***
8	***Input:** set type of uncertain parameters* $\boldsymbol{p_{u(t)}}$	
9	*set type of simulated target variables* $\boldsymbol{v_{ROM(t)}^\tau}$	
10	*set observation space* $\boldsymbol{\Omega}$	
11	*set action space* $\mathbf{A}$	
12	***Output:** Read the state of environment* $\boldsymbol{s_{(t0)} = \omega_{(t0)}}$	
13	***For** episode = 1:* ε ***do:***	***← STEP 5*** ↻
14	*Reset the initial state randomly*	
15	***While** True **do:***	
16	*Observe the state* $\boldsymbol{s_{(t)} = \omega_{(t)}}$	
17	*Select action* $a(p_u{}^j)$ *using the old policy* π	
18	*Compute new uncertain parameters* $\boldsymbol{p_{u(t)}}$	
19	***In ROM:***	
20	***Set** uncertain parameters* $\boldsymbol{p_{u(t)}}$	
21	***Run** analysis*	
22	***Read** simulated target variables* $\boldsymbol{v_{ROM(t)}^\tau}$	
23	***Output:** Obtain the observation* $\boldsymbol{\omega_{(t)}}$	
24	*Calculate reward* r_t *and obtain new state* $\boldsymbol{s_{(t+1)}}$	

25	*Update policy π*
26	*Check terminating rules*
27	*Break while*
28	*Update timestep $t+=1$*
29	***End While***
30	*Compute cumulative reward*
31	*Save new optimal policy function π^**
32	***End for***

4.3.5 Validation step and model selection

The final step of DRUM-Av2.0 consists of the validation phase to determine the optimal updated values of the uncertain parameters $\widehat{p_u}$, which represent the solution of the MU process. This phase consists of using the trained agent to search for the solution. First, a number of episodes e is set to run the agent on different conditions of the training phase and then, evaluate its performance. In this phase, the agent does not update its policy anymore but simply exploits the learned optimal policy $\pi^*(s)$ to search for the solution. If 90% of the episodes converges to the solution, (error between simulated target variables $v_{ROM(t)}^T$ and real target variables v_r^τ lower than tolerance value), the validation phase is considered successful. Otherwise, the policy learned by the agent is not acceptable and the agent needs further training to achieve a better performance.

To carry out the model selection, a statistical analysis of the solutions is performed, in which the recurring values of the uncertain parameters characterize different probability distributions, from which extract the solution $\widehat{p_u}$, which is represented by the mode. To evaluate the performance of DRUM-Av2.0, dispersion metrics can be employed. As indicated in (Bruno et al., 2025; Mangalathu et al., 2021; Aronhime et al., 2013), the agent is well-trained if the results are not dispersed over a coefficient of variation (CV) higher than 10%.

In the end, the update values of the uncertain parameters $\widehat{p_u}$ can be assigned to the FOM to assess the accuracy of the ROM. This verification is needed to ensure that the order reduction does not compromise the success of the MU process.

4.4 First case study: one storey RC frame

To test the proposed methodology, DRUM-Av2.0 is applied to two fictitious structures with increasing complexity. The first case study is a simple single storey, one-bay reinforced concrete (RC) structure, characterized by 4 uncertain parameters, while the second case study is a two storey frame RC structure presenting in-height irregularity and 12 uncertain parameters to tune.

The first case study is shown in Figure 47. All geometrical and mechanical properties are reported Tables 21 and 22. P indicates columns and B indicates beams with subscripts indicating the progressive number of elements, M indicates material with subscripts indicating difference in terms of elastic and shear moduli (E and G, respecively).

Table 21. Geometrical properties of the first case study

Id.	Cross-section (mxm)	Length (m)	Materials	Nodes Id.
P_1	0.30 x 0.30	5	M_1	1-5
P_2	0.35 x 0.35	5	M_2	2-6
P_3	0.30 x 0.30	5	M_1	3-7
P_4	0.30 x 0.30	5	M_1	4-8
B_1	0.30 x 0.50	10	M_3	5-6
B_2	0.30 x 0.50	5	M_3	5-7
B_3	0.30 x 0.50	5	M_3	6-8
B_4	0.30 x 0.50	10	M_3	7-8

Table 22. Mechanical properties of the first case study

Material	W_s [kN/m^3]	E [MPa]	ν	G [MPa]
M_1	25.0	31840	0.2	12500
M_2	25.0	34330	0.2	13750
M_3	25.0	31000	0.2	12917

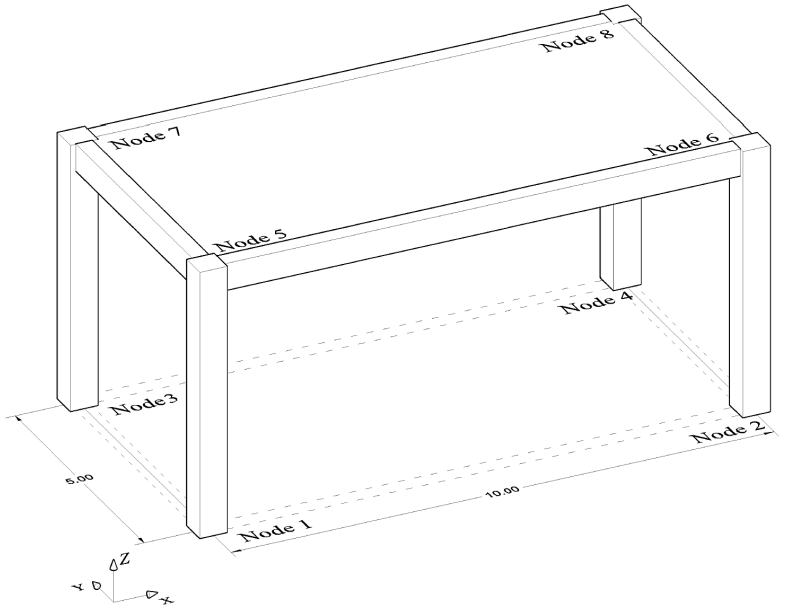


Figure 47. Geometrical configuration of the first case study

The structure presents fixed restraints at the base of P_1, P_3, and P_4, while a different degree of restraint at the base is assigned to P_2, by means of two rotational springs in the two main horizontal directions (i.e., *X* and *Y* as in Figure 47), characterized as $K_{rx} = 1.0\text{E} + 2.6999\ \text{kN} * \text{m/rad}$ and $K_{ry} = 1.0\text{E} + 5.2999\ \text{kN} * \text{m/rad}$.

4.4.1 Application of DRUM-Av2.0

For this case study, a set of uncertain parameters $p_u = [K_{rx}, K_{ry}, E_1, E_2]$ was defined, where E_1 and E_2 are the elastic moduli of the materials M_1 and M_2 and K_{rx} and K_{ry} are the rotational restraint conditions in node 2. A FOM of the structure was created in the structural software SAP2000 v.24 (Computers and Structures, Inc., 2024) to simulate a real monitoring campaign. According to the geometrical and mechanical features above defined and assuming the floor as infinitely rigid, a number of DOF is $n_{DOF} = 17$ was obtained. To simulate environmental vibrations, the FOM was subjected to a white noise with a frequency of 0.01 Hz and a duration of 600 s. The

acceleration values at nodes 5 and 8 were collected to simulate the presence of sensors at those points and an OMA analysis was performed using ARTeMIS software (Structural Vibration Solutions, 2023). The results are reported in Table 23, showing the frequencies of the first 7 vibration modes of the structure. These values represent the real target variables, $v_r^{\tau} = [f_1^{SHM}, f_2^{SHM}, f_3^{SHM}, f_4^{SHM}, f_5^{SHM}, f_6^{SHM}, f_7^{SHM}]$. It is worth specifying that all 17 vibration modes and the related frequencies could be considered, but the choice to use of only the first 7 modes lies in the representativeness of each mode for the dynamic behaviour of the structure.

Table 23. Values of the first 7 vibration modes for the first case study

Vibration mode	Frequency [Hz]
1° mode	$f_1^{SHM} = 1.88511$
2° mode	$f_2^{SHM} = 2.11519$
3° mode	$f_3^{SHM} = 2.20930$
4° mode	$f_4^{SHM} = 35.01944$
5° mode	$f_5^{SHM} = 35.06496$
6° mode	$f_6^{SHM} = 35.13608$
7° mode	$f_7^{SHM} = 42.05302$

Using the FOM, the dataset $\boldsymbol{D}$ was created by using the automated procedure in (Bruno et al., 2024) and considering different combinations of uncertain parameters p_u. To avoid unphysical results, extremes for the uncertain parameters were set, i.e., E_1 and E_2 ranges from $27500\ MPa$ to $35500\ MPa$, while K_{rx} and K_{ry} range from $1.0\mathrm{E}+00\ \mathrm{kN}*\mathrm{m/rad}$ (pinned constrain) to $1.0\mathrm{E}+12\ \mathrm{kN}*\mathrm{m/rad}$ (fixed condition). To perform combinations, the ranges identified were subdivided into discrete intervals: a step size $1000\ MPa$ was considered for E_1 and E_2, while for K_{rx} and K_{ry} the following values were assumed: $[1, 1.0\mathrm{E}+2, 1.0\mathrm{E}+5, 1.0\mathrm{E}+8, 1.0\mathrm{E}+10]\ \mathrm{kN}*\mathrm{m/rad}$. By combining all values, a total number of combinations $c = 1600$ was considered. The dataset $\boldsymbol{D} = (d_1, \dots, d_h, \dots, d_c)^T\ with\ h = 1, \dots, 1600$ was generated in around 5.7 hrs., with $d_h = [p_u(h), u(h), v_{FOM}^{\tau}(h), M(h), K(h)]$ and it represents the input of the ROM. To this end, the snapshot matrix $\boldsymbol{S_m}$ was defined according to Equation (25), in which the parametrization proposed in Equation (33) was applied only to

K, since M was not object of updating. Hence, according to the number of selected vibration modes, the number of principal orthonormal components r is equal to 7, while the columns of the matrix of left singular vectors, L_i, presents the size of 17 (according to the SVD application). By projecting mass M and stiffness K matrices into the reduced space derived from the POD basis, a reduced order matrices were obtained for mass and stiffness, i.e., M_r and K_r, respectively. The order reduction leads to obtain a model with 7 DOFs from 17 initial DOFs. To validate the ROM, 100 eigenvalue analyses with random values for $p_u(h) = [K_{rx}(h), K_{ry}(h), E_1(h), E_2(h)]$ were performed and a comparison in terms of frequencies (i.e., $v_{ROM}^{\tau}(h) = [f_1^{ROM}(h),\ f_2^{ROM}(h),\ f_3^{ROM}(h),\ f_4^{ROM}(h),\ f_5^{ROM}(h),\ f_6^{ROM}(h),\ f_7^{ROM}(h)]$) with the results obtained by the FOM (i.e., $v_{FOM}^{\tau}(h) = [f_1^{FOM}(h), f_2^{FOM}(h), f_3^{FOM}(h), f_4^{FOM}(h), f_5^{FOM}(h), f_6^{FOM}(h), f_7^{FOM}(h)]$) under the same condition was carried out. Figure 48 and Table 24 show the comparison in terms of differences between the eigenvalues computed with ROM and FOM. As can be observed, a maximum error of 1.6E-14 Hz was detected, which is extremely low if compared to the main 7 frequencies. Still, it is interesting observing the computational time required to perform the 100 analyses with both methods: the ROM employs a time about 5 hundreds of times less than the one required by the FOM. This result confirms that the ROM is sufficiently accurate, as well as it is strongly efficient in terms of computational time.

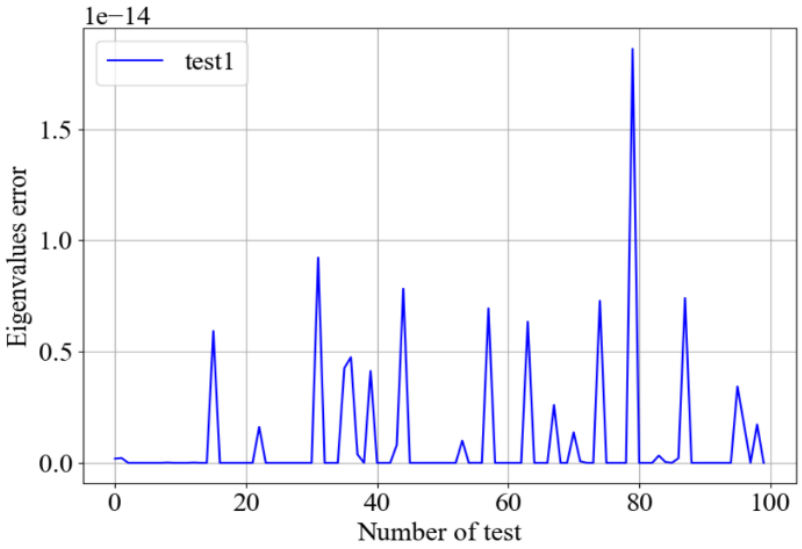


Figure 48. Eigenvalues error between ROM and FOM for the first case study

Table 24. Comparisons between ROM and FOM for the first case study

Test	ROM	FOM in SAP2000
No. of iterations	100	100
No. of DOFs	7	17
Time	1.8 s	840 s

Once ROM is available, the DRL framework was implemented in Python programming language (Python Software Foundation, 2024). The state space coincides with the observation space and each observation at the timestep t is defined as $\omega_{(t)} = \{v^{\tau}_{ROM(t)}, p_{u(t)}\}$ where $p_{u(t)} = [K_{rx(t)}, K_{ry(t)}, E_{1(t)}, E_{2(t)}]$ and $v^{\tau}_{ROM(t)} = [f^{ROM}_{1(t)}, f^{ROM}_{2(t)}, f^{ROM}_{3(t)}, f^{ROM}_{4(t)}, f^{ROM}_{5(t)}, f^{ROM}_{6(t)}, f^{ROM}_{7(t)}]$. The agent acts on the environment by progressively modifying the values of each $p^{j}_{u(t)}$, so that $a_{(t)} = [K_{rx(t-1)} \cdot \Delta K_{rx(t)}, K_{ry(t-1)} \cdot \Delta K_{ry(t)}, E_{1(t-1)} + \Delta E_{1(t)}, E_{2(t-1)} + \Delta E_{2(t)}]$ where $\Delta E_{1(t)}$ and $\Delta E_{2(t)}$ are terms to add and are selected in the continuous range [-300, +300] MPa, while $\Delta K_{rx(t)}$ and $\Delta K_{ry(t)}$ are terms to multiply and are selected in the continuous range in [0.1, 10]. At each iteration t, the agent receives negative instantaneous value of the reward $r_{(t)}$ expressed as

$$r_{(t)} = -Co + \sqrt[3]{\Delta\sigma} + p \tag{35}$$

where Co is a constant equal to 1 and imposed with a negative sign to incentive the agent to reach the solution quickly, p is a penalty given to the agent when performs actions that exceed the extreme ranges of uncertain parameters p_u (p is equal to -1 to discourage the agent in pursuing physically implausible solutions), $\Delta\sigma = |(\sigma_{(t)} - \sigma_{(t-1)})|$, i.e., is the normalized difference between the error value at between two successive instants. The minimization of $\Delta\sigma$ drives the agent along more direct trajectories towards the optimal solution. The extraction of the cube root allows to appropriately resize the contribution of $\Delta\sigma$. Finally, the deviation $\sigma_{(t)}$can be expressed as

$$\sigma_{(t)} = \frac{\sum_{k=1}^{l} |v_r^{\tau k} - v_{ROM(t)}^{\tau k}|}{l} \tag{36}$$

Equation (35) represents the selected reward strategy, set to penalizes the agent when the error between $v_r^{\tau k}$ and $v_{ROM}^{\tau k}$ increases and rewarding it when approaching the optimal solution (reduction of the error at the increment of timestep). Two terminating conditions, checked at each timestep, were added to determine the end of each episode. The first restricts the agent to perform a maximum $m = 100$ timesteps to avoid non-convergence. The second is related to achieve a solution for the MU problem, i.e., to find values of uncertain parameters $\widehat{p_u}$ which imply $\sigma_{(t)} < \sigma_{end}$, with $\sigma_{end} = 0.03$, which represents the tolerance and accuracy value to achieve during the training phase. The learning procedure was performed with machine 12th Gen Intel(R) Core (TM) i7-12700H CPU, 32 GBs RAM, and Nvidia GeForce RTX 3070Ti GPU with 8 GBs. The agent's learning capability was modelled the Twin Delayed Deep Deterministic Policy Gradient algorithm (Fujimoto et al., 2018).

Figure 49 shows the results of the agent's training phase, with Figure 49(a) reporting the trend of the deviation $\sigma_{(t)}$ at the end of each episode and Figure 49(b) depicting the trend of cumulative reward in each episode. Both graphs highlight the ability of the agent in exploring solutions progressively more efficiently, converging to optimal values after about a half of the total episodes. In fact, the moving average of

the reward was stabilized around the value -15 and the deviation achieved low values according to the tolerance ($\sigma_{(t)} < 0.03$).

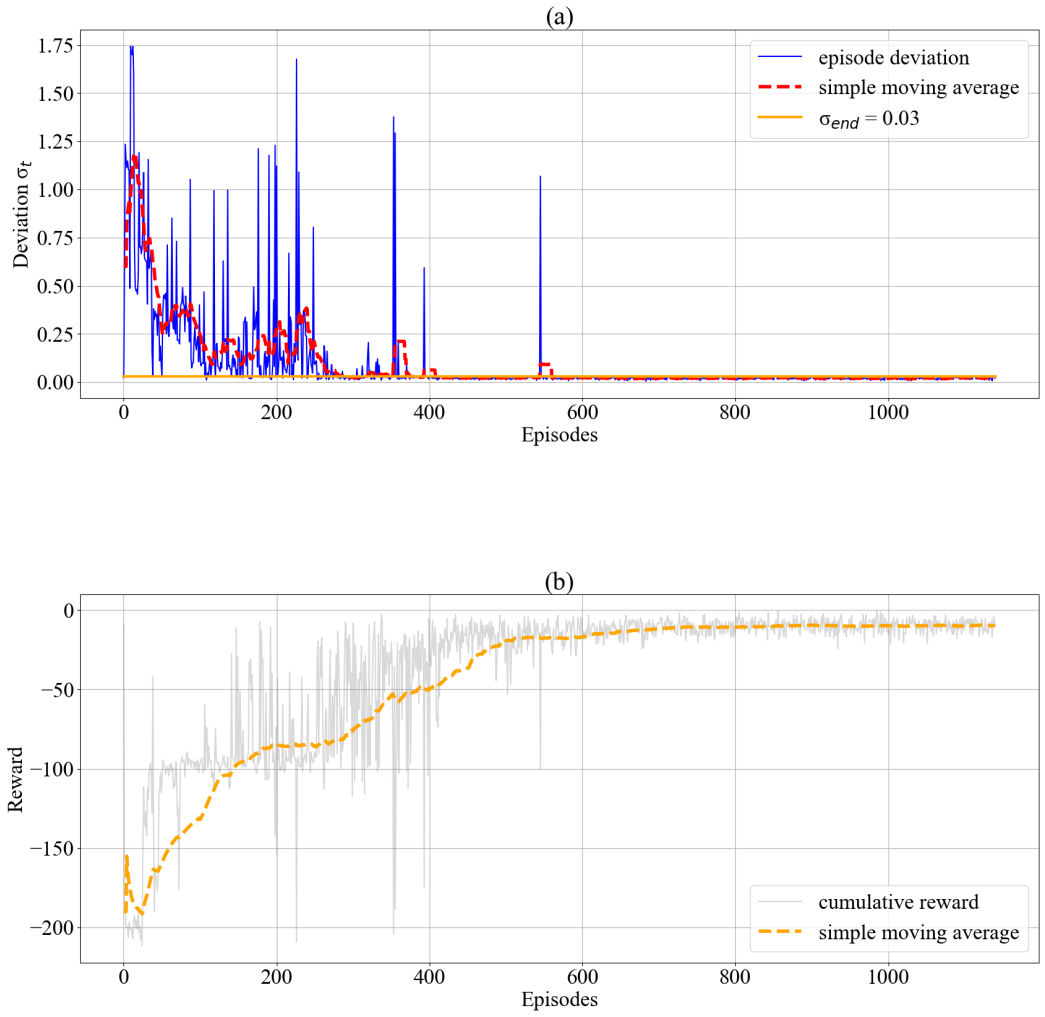


Figure 49 Result of training phase for the first case study: (a) trend of the error $\sigma_{(t)}$ with simple moving average in red line; (b) culuative reward for each episode with the simple moving average in orange line.

After training, the validation phase was conducted by setting a maximum of 100 episodes and a threshold $\sigma_{end} = 0.03$. Figure 6 reports the validation phase with the same metrics of Figure 49, and the results confirm the agent converge to a solution (i.e., $\widehat{p_u}$) by minimising the error between numerical and experimental frequencies. The cumulative reward trend at the end of the episode stabilizes around the value of -10, where the residual value is given by the number of iterations necessary to reach the optimal solution (Figure 50(b)).

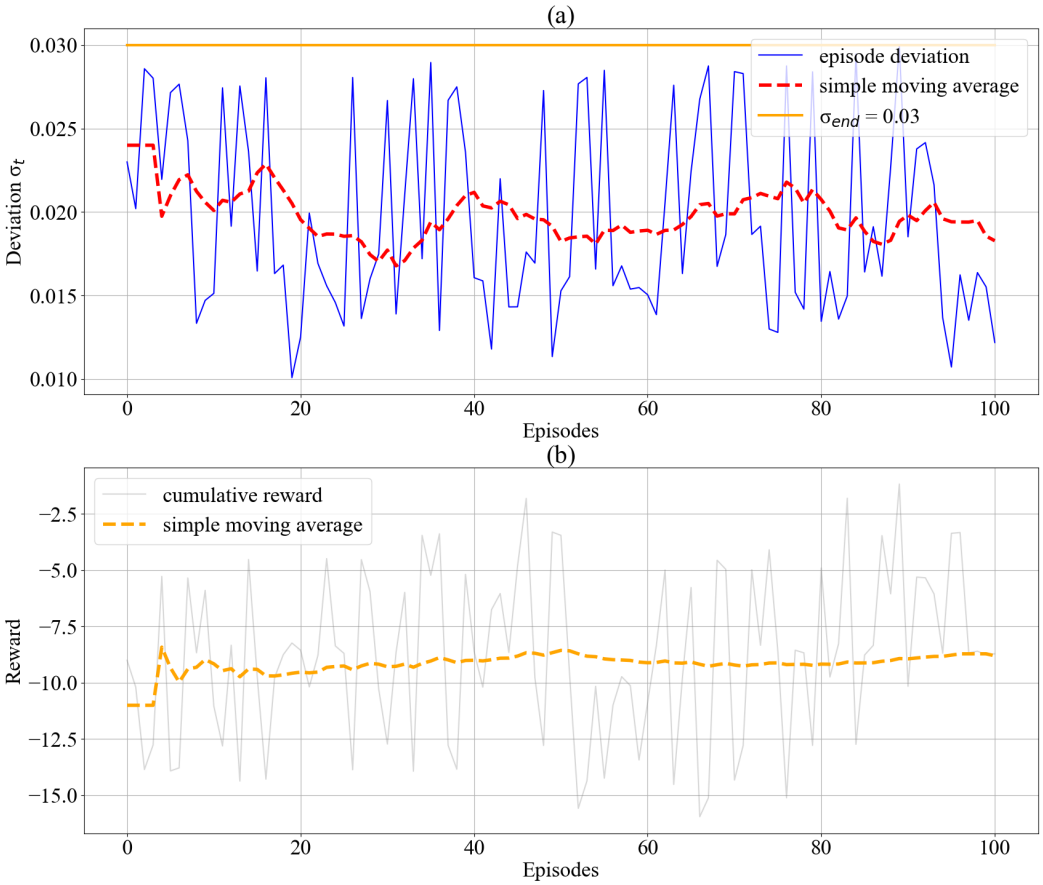


Figure 50. Results of the DRUM-Av2.0 test and validation phase for the case study: a) trend of the deviation achieved at the end of each episode and average deviation during the test phase; b) cumulative reward trend at the end of each episode and average cumulative reward during all test phases.

Once obtained the solutions in terms of p_u, a statistical study of the obtained solutions was performed for estimating the optimal solution $\widehat{p_u}$. The results of the statistical analysis are reported in Table 25 and Figure 51, where the distributions show low dispersion, do not present outliers, and are always comprised within the extremes considered. Furthermore, the consistency between the mean, median and mode confirms the high stability of the estimates. In Figure 51, the graphs along the diagonal show the trend of the probability distributions of all the individual uncertain parameters, which are narrow and present a normal-like shape, with CV lower than 10%.

Table 25. Statistical analysis on the results obtained in the testing phase by DRUM-Av2.0 with ROM for the first case study

Measure	E_1	E_2	$Log_{10}(Krx)$	$Log_{10}(Kry)$
Minimum	31826	34095	0.34	2.59
Maximum	31885	34336	4.44	9.64
Mean	31845	34212	2.50	8.34
Median	31400	34245	2.53	8.80
Mode	31827	34294	1.42	7.86
Standard deviation	11.66	72.19	0.23	0.63
CV (%)	0.04 %	0.21 %	9.2 %	7.6 %

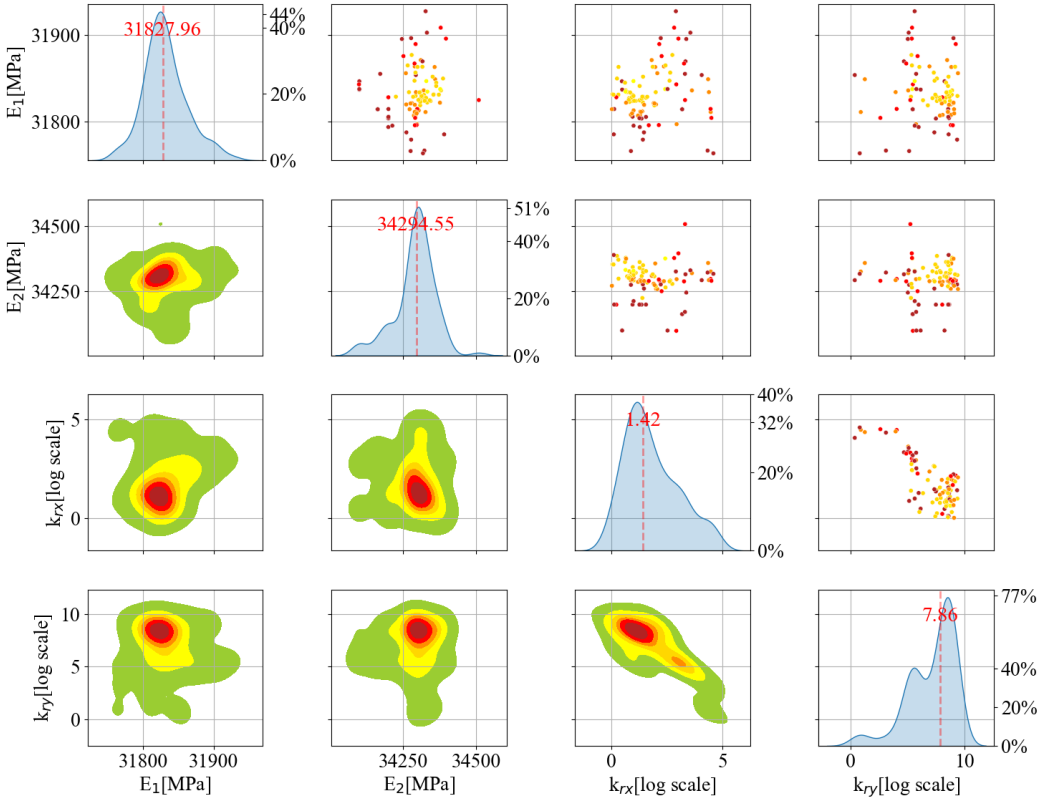


Figure 51. Pairplot of the results obtained during the agent validation phase. The graphs along the diagonal show the probability distribution of each uncertain parameter $[E_1, E_2, K_{rx}\ e\ K_{ry}]$. Outside the diagonal, we find the dispersion of correlations between uncertain parameters, with a colour indicating the respective deviation value $\sigma_{(t)}$.

After the selection of uncertain parameter $\widehat{p_u}$ values, a comparison with the real values can be performed. Results are reported in Table 26, and they show the high accuracy of DRUM-Av2.0.

Table 26. Comparison between the most frequent values (mode) obtained through DRUM-Av2.0 and the real values

Parameter	*Real values*	*DRUM-Av2.0*
E_1 [Mpa]	31842	31827
E_2 [Mpa]	34330	34294
K_{rx} [kNm/rad]	1.0 E+1.7	1.0E+1.42
K_{ry} [kNm/rad]	1.0 E+6.6	1.0E+7.86

The obtained solution by the MU process was further tested by updating the FOM with the same values of $\widehat{p_u}$, to obtain after eigenvalue analysis the results in terms of v_{FOM}^{τ}. As expected, the comparison between the values in output from the ROM and from the FOM confirms the accuracy of the solution obtained by the DRUM-Av2.0 (i.e. very low error rate, for this reason are not reported, see Table 24).

To further stress the proposed approach, the robustness of DRUM-Av2.0 against a slight variation of the target variables v_r^{τ} was tested. In particular, the performance of the agent was assessed towards different values of v_r^{τ}, selected as progressively more distant from the training values. Increasing percentage deviations were assigned to v_r^{τ}, i.e., 1%, 2%, 3%, 6% and 10%. For each of the new values of v_r^{τ}, the validation and model selection phase were repeated. The results reported in Table 27, which shows full convergence (100%) of the episodes towards the solution for percentage deviation values up to 3%. In these cases, an average cumulative reward stable at around -10 was observed, value comparable to the one observed in the last part of the training, although different values v_r^{τ} were used. This implies the ability of the agent in achieving the solution also with slight modifications of v_r^{τ}. Different considerations can be drawn if increase the percentage deviation, for which results are not satisfactory. In these cases, a training from scratch of the agent is required, or as future developments, a modification of the algorithm could be implemented to extend the ability of the agent in converging towards different solutions.

Table 27. Results in terms of robustness of the trained agent in converging on the solution for target variable values other than those used during training v_r^{τ}

Deviation from v_r^T	$\widehat{\boldsymbol{p_u}} = [\boldsymbol{E_1}, \boldsymbol{E_2}, \boldsymbol{K_{rx}}, \boldsymbol{K_{ry}}]$	***Convergence of episodes***	***Mean of cumulative reward***	***Mean deviation at the end of episodes***
-1%	[31812, 34281, 1.63, 8.52]	100	-9	0.017
+1%	[31834, 34319, 1.25, 8.35]	100	-10	0.020
-2%	[31801, 34245, 1.91, 8.47]	100	-10	0.020
+2%	[31856, 34365, 1.50, 8.07]	100	-11	0.023
-3%	[31792, 34222, 1.80, 8.46]	100	-12	0.022
+3%	[31871, 34366, 1.87, 8.66]	100	-13	0.025
-6%	\	82	-50	0.031

+6%	78	-61	0.032
-10%	5	-90	0.10
+10%	0	-98	0.11

4.4.2 Comparative analysis of DRUM-Av1.0 and DRUM-Av2.0

To assess the main difference of the new version of the methodology, a comparison between the performances of DRUM-Av1.0 and DRUM-Av2.0 was provided. In detail, the comparison was conducted in terms of accuracy according two performance metrics, i.e., Root Mean Square Error (RMSE) and Weighted Absolute Percentage Error (WAPE), and in terms of efficiency, by observing the computational time required by the agent in the phases of training and validation. Table 28 shows the comparative results by referring to the first three main frequencies for the sake of synthesis. In terms of accuracy, the results by ROM for RMSE and WAPE, both for uncertain parameters and target variables, are lower than the ones obtained by FOM. This result does not imply that the updating with the ROM is more accurate than the one by FOM but implies that the reduced size of the ROM allowed to run more simulations, enabling the high potentiality of the proposed DRL framework to find the optimal solution. Concerning the efficiency, it is worth noting that the computational time required for a single timestep by the agent interacting with the ROM is approximately 0.02 s, while the same operation using the FOM requires approximately 5 s. This consistent difference, as above stated, allows to perform more iterations with the ROM during the training phase (for the same amount of time, 250 times more than FOM), leading to a more stable and efficient learning ability. Obviously, the use of the ROM is strongly suggested for complex structures, in which the number of uncertain parameters increases. According to the performed analyses (50000 timesteps for the ROM, 20000 timesteps for the FOM), the total computational time for both the training and validation phases with the ROM is approximately 17 minutes, while with FOM is approximately 15 hours. It is worth noting that, although the ROM is extremely faster than the FOM, it requires additional time for the creation of the dataset at the base of its generation. In any case, adding also this extra computational time, victory leans on ROM's side.

Table 28. Comparison between the results obtained with ROM in DRUM-A and the direct use of FOM as an environment for DRUM-A

Data	OMA	DRUM-Av1.0	DRUM-Av2.0
E_1[MPa]	31842	31400	31827
E_2[MPa]	34330	33800	34294
k_{rx}[kN*m/rad]	1.0 E+1.7	1.0E+3	1.0E+1.42
k_{ry}[kN*m/rad]	1.0 E+6.6	1.0E+6	1.0E+7.86
f_1[Hz]	1.88864	1.90477	1.88794
f_2[Hz]	2.14317	2.14040	2.14359
f_3[Hz]	2.25529	2.24605	2.25697
RMSE - p_u	---	345.060	19.511
RMSE - v_r^τ	---	0.0109	0.0011
WAPE - p_u [%]	---	0.01472	0.00079
WAPE - v_r^τ[%]	---	0.00448	0.00045
Time for each timestep [s]	---	5.3	0.018
Timesteps in training	---	10 000	50 000
Time for training [hours]	---	14.7	0.25
Timesteps in validation	---	100	100
Time for validation [min]	---	29	1.1
Total time [hours]	---	14.9	0.27

4.5 Second case study: two storey RC frame

The proposed procedure is applied to a more complex fictitious structure, characterized by a high number of uncertain parameters. The geometry of the structure is shown in Figure 52, and it consists of 10 columns with variable cross-sections and elastic moduli, and 11 beams with a rectangular cross-section of 30x50 cm and same material (E equal to 31,000 MPa). Columns P_1, P_2, P_3, P_5, P_6, P_7 present a rectangular cross-section of 30x60 cm, while the others present a square cross-section of 30x30 cm. Each column is characterized by different materials between the bottom and upper portions, equally subdivided. The upper parts of the columns (black branches in Figure 52) present the same material of the beams (E equal to 27,000 MPa), while the bottom parts of the columns (blue branches in Figure 52) present different materials with different elastic moduli. The structure is totally fixed at the base, except for node 6 (red

dot in Figure 52), which presents two rotational springs in the two main horizontal directions (i.e., *X* and *Y* as in Figure 52), characterized as K_{rx} and K_{ry}. Table 9 reports all real values of the uncertain parameters $\boldsymbol{p_u}$.

Table 29. Real values of uncertain parameters $\boldsymbol{p_u}$

Parameter	Value	Parameter	Value
E_1	28324 MPa	E_{13}	29675 MPa
E_3	31895 MPa	E_{15}	27930 MPa
E_5	33519 MPa	E_{17}	32985 MPa
E_7	31674 MPa	E_{19}	28312 MPa
E_9	33704 MPa	$Log_{10}(Krx)$	4.80 kN*m/rad
E_{11}	33147 MPa	$Log_{10}(Kry)$	7.40 kN*m/rad

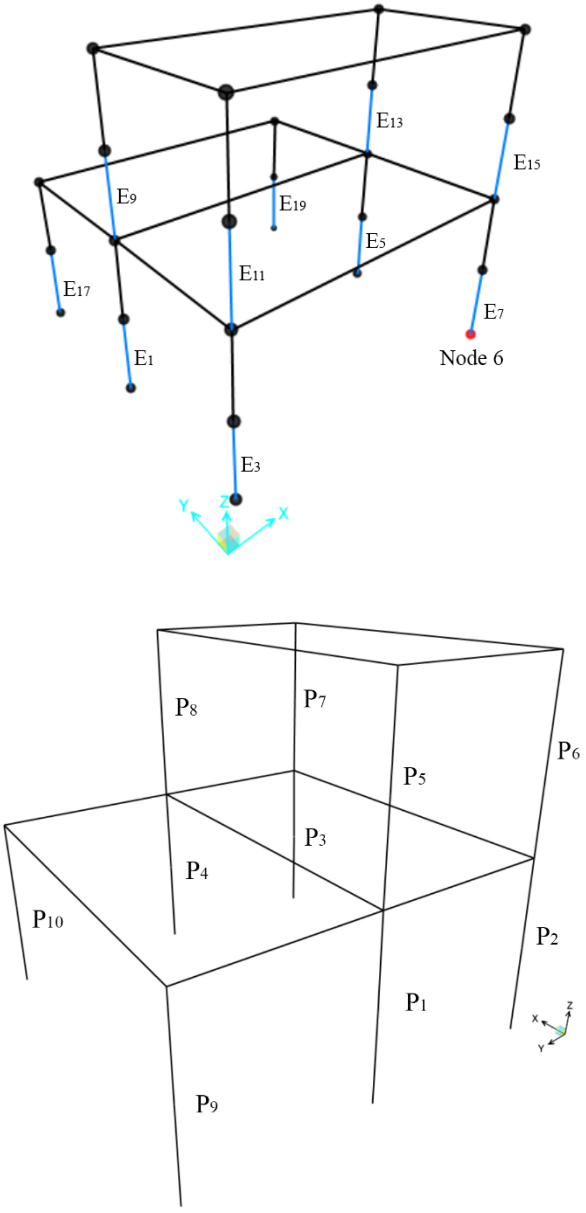


Figure 52. Axonometric representation of the second case study: definition of structural materials (left), denomination of the columns (right)

4.5.1 Application of DRUM-Av2.0

After defining the geometry of the structure, the uncertain parameters $\boldsymbol{p_u} = [K_{rx}, K_{ry}, E_1, E_3, E_5, E_7, E_9, E_{11}, E_{13}, E_{15}, E_{17}, E_{19}]$ were chosen. A FOM was created with the structural software SAP2000 v.24 (Computers and Structures, Inc., 2024) (floors assumed as infinitely rigid) and a simulated monitoring campaign was performed (same force of the first case study). According to the geometrical and mechanical features of the structure, a number of DOF is $n_{DOF} = 98$ was obtained. OMA analysis was performed using ARTeMIS software (Structural Vibration Solutions, 2023) and the first 10 frequencies were selected as real target variables, i.e., $\boldsymbol{v_r^\tau} = [f_1^{SHM}, f_2^{SHM}, f_3^{SHM}, f_4^{SHM}, f_5^{SHM}, f_6^{SHM}, f_7^{SHM}, f_8^{SHM}, f_9^{SHM}, f_{10}^{SHM}]$. Table 30 reports the values of the considered frequencies, accounting for 10 modes over 98 ones, considered as fully representative of the structure.

Table 30. Values of the first ten vibration modes for the dummy structure

Vibration mode	Frequency [Hz]
1° mode	$f_1^{SHM} = 1.45247$
2° mode	$f_2^{SHM} = 1.63000$
3° mode	$f_3^{SHM} = 2.05910$
4° mode	$f_4^{SHM} = 3.24120$
5° mode	$f_5^{SHM} = 4.70396$
6° mode	$f_6^{SHM} = 6.07058$
7° mode	$f_7^{SHM} = 18.93247$
8° mode	$f_8^{SHM} = 25.33314$
9° mode	$f_9^{SHM} = 27.09041$
10° mode	$f_{10}^{SHM} = 27.71189$

Using the FOM, the dataset $\boldsymbol{D}$ was created and to avoid unphysical results and to reduce the computational burden, discrete values were used for the uncertain parameters, i.e., $27000\ MPa$ and $35000\ MPa$ for elastic moduli (extreme values) and $1.0\mathrm{E}+00\ \mathrm{kN}*\mathrm{m/rad}$ (pinned constrain), 1.0E+6 kN*m/rad (intermediate), and 1.0E+12 kN*m/rad (fixed condition) for the rotational spring. According to the

considered values, dataset $\boldsymbol{D}$ was constituted by a number of combinations $c = 9216$, requiring around 19.2 hrs of analyses. From $\boldsymbol{D}$, the snapshot matrix $\mathbf{S_m} \in \mathbb{R}^{n_{DOF} \times c}$ was derived, and the application of the SVD to $\boldsymbol{S_m}$ generates a reduced representation of the structure with 10 DOFs, i.e. $\boldsymbol{\Phi} = [\boldsymbol{L_1}, \dots, \boldsymbol{L_i}, \dots, \boldsymbol{L_{10}}]$. After, $\boldsymbol{D}$ was used to parametrize K, and by projecting mass M and stiffness K matrices into the reduced space derived from the POD basis, the ROM was elaborated for the MU process.

Figure 53 shows the comparison between the results obtained by the ROM and the FOM, expressed in terms of the eigenvalues error for 100 random combinations of the uncertain parameters p_u. The maximum observed value of the metric employed amounted to a maximum of 0.004 Hz for the frequencies of the first ten modes of vibration. Table 31 shows the differences in terms of DOFs and computational time required to perform the ROM tests in comparison with the one required by the FOM. In detail, the time required to perform the test of the ROM accuracy reveals that FOM requires 300 times the time necessary to achieve the same solution with respect to the ROM.

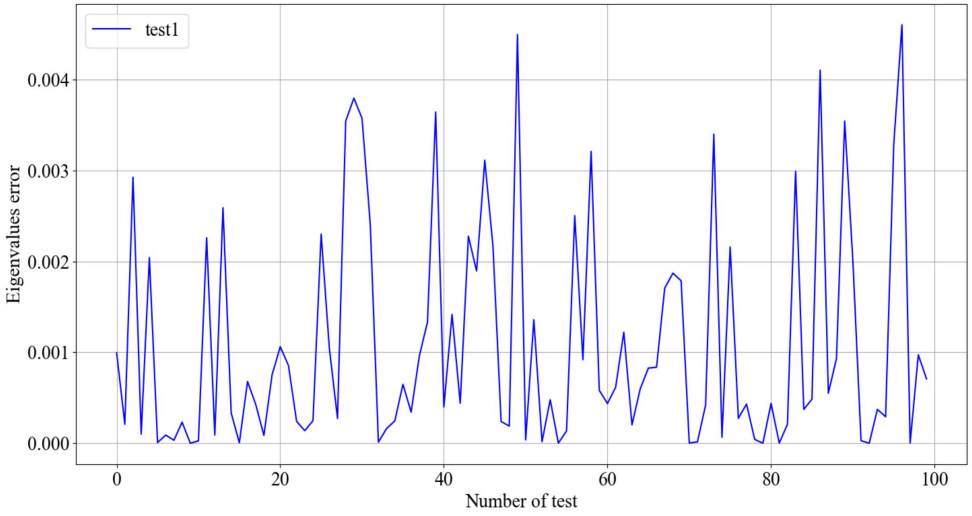


Figure 53 Eigenvalues error between ROM and FOM for the second case study

Table 31. Comparisons between ROM and FOM for the second case study

Test	ROM	FOM in SAP2000
No. of iterations	100	100
No. of DOFs	10	98
Time	3 s	778 s

Once ROM was created, it was implemented in DRUM-Av2.0. Due to the high complexity of the structural system, some further refinements are required with respect to the first case study, to define the episode termination conditions. In particular, the tolerance σ_{end} to terminate an episode was implemented as adaptive on the training phase. The logic behind this choice is to improve the training phase of the agent, because the adaptive threshold is initialized with a near-high value, and during training, as the agent obtains better results, the threshold is reduced. This approach aims to prevent that in a complex system like the one investigated, the agent associated a local minimum to the solution, which can deviate the success of the training phase in achieving the correct solution (Portelas et al., 2020; Xu et al., 2019). Substantially, the termination condition is expressed by $\sigma_{(t)} < {\sigma_{end}}^{Adp}$, where $\sigma_{(t)}$ is defined by Equation (36), while the adaptive threshold ${\sigma_{end}}^{Adp}$ represents the tolerance value. For the case at hand, an initial threshold ${\sigma_{end}}^{Adp} = 0.07$ was set, and as the agent improves its ability to direct toward the correct solution, ${\sigma_{end}}^{Adp}$ decreases. The decreasing rule for ${\sigma_{end}}^{Adp}$ was fixed to a multiplying factor, named decay factor δ, equal to 0.95, when the cumulative reward over the last 300 episodes (considered as in a sliding window) was higher than -30. Otherwise, ${\sigma_{end}}^{Adp}$ remains fixed. The minimum value of that ${\sigma_{end}}^{Adp}$ can assume was fixed to $\sigma_{end} = 0.03$. Although some differences from the first case study, the learning algorithm for the agent (TD3 in Fujimoto et al., 2018) and the machine used for the training phase remain unchanged.

Figure 54 shows the results of the training phase, where Figure 54(a) reports the trend of the deviation $\sigma_{(t)}$ at the end of each episode and Figure 54(b) reports the trend of cumulative reward in each episode. Figure 54(a) shows also the influence of the adaptive tolerance strategy: during the first 500 episodes, the agent rapidly learned the

best strategy for achieving convergence, showing an initial decrease of ${\sigma_{end}}^{Adp}$. After, ${\sigma_{end}}^{Adp}$ continues to decrease until achieving the value close to $\sigma_{end} = 0.03$ around the episode 1800. At this point, as can be observed in Figure 54(b), the agent explores other possibilities, but this behaviour induces a reduction of the cumulative reward, which in turn interrupts the decrease of ${\sigma_{end}}^{Adp}$. Starting from the episode 2000, the agent aims to the best solution in all episodes, the cumulative reward increases, and then the ${\sigma_{end}}^{Adp}$ around episode 2500 decreases up to the final value, i.e., $\sigma_{end} = 0.03$. In the end, the average cumulative reward trend from mid-training onwards remains around -20.

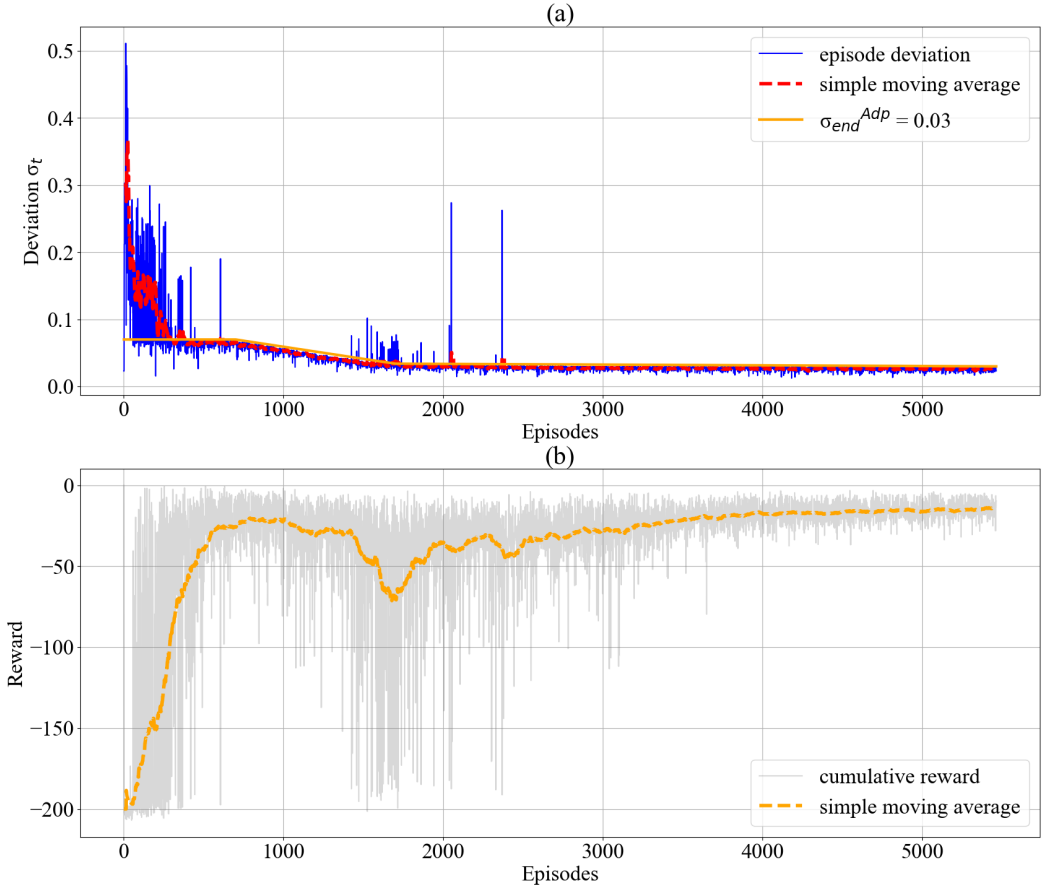


Figure 54. Result of training phase for the second case study: (a) trend of the error $\sigma_{(t)}$ with simple moving average in red line; (b) cumulative reward for each episode with the simple moving average in orange line.

After, the validation phase of the trained agent was conducted on 100 episodes with a fixed threshold $\sigma_{end} = 0.03$. During this phase, the agent always converged toward the solution, obtaining a cumulative reward value at the end of each episode equal to -13. According to the proposed procedure, a statistical analysis was performed to derive the optimal solution $\widehat{\boldsymbol{p_u}}$. Figure 55 reports the probability distributions for all uncertain parameters ($K_{rx}, K_{ry}, E_1, E_3, E_5, E_7, E_9, E_{11}, E_{13}, E_{15}, E_{17}E_{19}$), showing a normal-like shape, low dispersion, without outliers, with a CV lower than 10%. Additionally, Table 32 summarizes all results of the statistical analysis, in which the mode is to consider as the optimal value for updating the model.

Table 32. Statistical analysis on the results obtained in the testing phase by DRUM-Av2.0 with ROM for the second case study

p_u	Minimum	Maximum	Mean	Median	Mode	Standard deviation	CV (%)
$\mathrm{E_1}$	27877	29117	28689	28711	28697	238.55	0.83
$\mathrm{E_3}$	28739	34628	31831	32048	32627	1495.21	4.70
$\mathrm{E_5}$	31698	34037	32912	32892	32907	458.85	1.40
$\mathrm{E_7}$	29112	32283	30604	30523	30504	623.06	2.04
$\mathrm{E_9}$	30780	34772	32546	32513	32138	953.51	2.93
$\mathrm{E_{11}}$	28088	34854	31611	31388	31141	1657.14	5.24
$\mathrm{E_{13}}$	27693	34983	31502	31714	31751	1716.99	5.45
$\mathrm{E_{15}}$	27500	34471	29929	29828	29683	1311.82	4.38
$\mathrm{E_{17}}$	27766	33958	30703	30571	30510	1321.15	4.30
$\mathrm{E_{19}}$	27627	34336	31005	31110	31758	1454.32	4.69
$Log10(\mathrm{k_{rx}})$	4.44	9.21	6.31	5.67	5.31	0.59	9.34
$Log10(\mathrm{k_{ry}})$	5.81	9.95	8.38	8.69	8.97	0.82	9.78

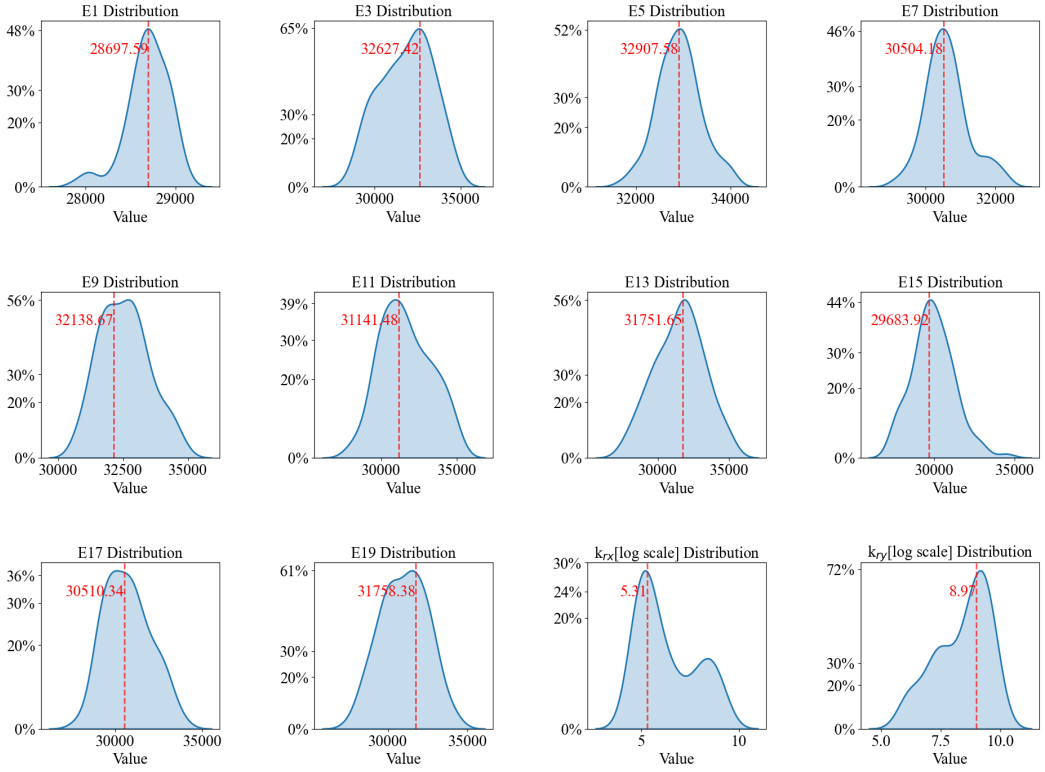


Figure 55 Probability distributions for all uncertain parameters. The red dotted line on the individual graphs indicates the mode for each parameter.

Using the mode parameters as $\widehat{p_u}$, the FOM was updated and values of v_{FOM}^{τ} were derived to compare with v_r^{τ}. Table 33 shows the above comparison, highlighting percentage differences in the first ten frequencies of less than 1.5%, negligible values for the problem under consideration. As can be observed, the proposed methodology is strongly accurate, although the high complexity of the proposed case study.

Table 33. Comparison in terms of v_r^{τ} between real values and numerical updated values obtained by DRUM-Av2.0.

Target variables	*DRUM-Av2.0 [Hz]*	*Real values [Hz]*	*Percentage difference [%]*
f_1^{SHM}	1.4727	1.45247	-1.40
f_2^{SHM}	1.6485	1.63	-1.13

f_3^{SHM}	2.0625	2.0591	-0.17
f_4^{SHM}	3.2590	3.2412	-0.55
f_5^{SHM}	4.7327	4.70396	-0.61
f_6^{SHM}	6.0204	6.07058	0.83
f_7^{SHM}	18.9211	18.93247	0.06
f_8^{SHM}	25.3562	25.33314	-0.09
f_9^{SHM}	27.0764	27.09041	0.05
f_{10}^{SHM}	27.9003	27.71189	-0.68

4.5.2 Efficiency of the framework

After assessing the accuracy of DRUM-Av2.0 (see Table 33), a note about the efficiency of the proposed approach is provided, with reference to the second case study. The use of the ROM in DRUM-Av2.0 demonstrates also efficiency in finding uncertain parameters and then in speeding MU process. To demonstrate this aspect, Table 34 summarizes the computational time required to complete each phase of the entire process for the second case study. The time required for each timestep is 0.048 s, which allows a substantial increment of the total number of timesteps for implementing an extended and robust training phase. For the case at hand, 100000 time intervals were used, for a total time of 80 minutes for training. Over the same length of time, if using FOM and considering about 7 seconds for each timestep, less than 700 timesteps could be employed for the training, with an evident impossibility in performing this phase while achieving high accuracy. Regarding validation phase, using 100 timesteps, a time of 4.8 minutes is required, for a total time of 85 minutes for the entire process (if excluding 21 hrs required for creating the dataset), which is a negligible effort if considering the time required for the same operation by using FOM. As demonstrated for this complex case study, the reduced computational requirement by the ROM allows to employ a longer training phase for aiming high accuracy and agent reliability, while retaining high efficiency.

Table 34. Efficiency parameters to evaluate DRUM-Av2.0 on the second case study

Metric	Value
Time for each timestep [sec]	0.048
Timesteps in training	100 000
Time for training [min]	80
Timesteps in validation	100
Time for validation [min]	4.8
Total time [min]	85

4.6 Discussion

This section provides a comprehensive description of the evolution and application of the DRUM-Av2.0 method, highlighting its central role as an integrated approach between DRL and ROM in the MU process for monitored structures. The combination of reduced-order numerical models and autonomous learning techniques represents a significant advance over traditional procedures, allowing accurate solutions to be obtained with a considerable reduction in computational costs. The first significant aspect that emerged concerns the consistency of the proposed framework: each phase, from the collection of experimental data via SHM and the identification of target variables, to the construction of models and validation, was designed to ensure a continuous balance between physical consistency and numerical efficiency. Replacing the Full Order Model as the training environment for the DRL agent with a carefully validated reduced model made it possible to maintain the accuracy of the solution while drastically reducing computation times, making the algorithm compatible with real-time applications or multiple iterations on complex systems.

From a methodological point of view, the adoption of TD3 proved to be strategic. This algorithm, capable of managing continuous and high-dimensional action spaces, improved stability and convergence speed compared to previous solutions implemented in DRUM-Av1.0. The introduction of targeted reward functions, capable of penalising physically implausible actions and rewarding progressively optimised

learning trajectories, has further refined the training process, leading to an increase in the robustness of policy learning.

The case studies examined, relating to single- and double-storey RC frames, demonstrated the flexibility and reliability of the approach. In the first case, the model showed highly stable behaviour with negligible errors between the frequencies obtained from ROM and FOM, confirming the validity of the order reduction and the effectiveness of the parameterisation phase. In the second case, characterised by greater structural complexity and a higher number of uncertain parameters, DRUM-Av2.0 continued to show high efficiency and consistency in the results, with significantly reduced calculation times compared to the previous method. Another significant result that emerged from the validation concerns the model's capacity to generalise: the policy learned by the agent remains effective even when initial conditions vary, demonstrating the robustness of the training process. Furthermore, statistical analysis of the updated values of the structural parameters showed a dispersion of less than 10%, indicating stable and repeatable learning.

In brief, DRUM-Av2.0 represents a substantial evolutionary step forward in automated MU through deep learning techniques and reduced modelling. It combines engineering accuracy, numerical efficiency and adaptive capability, extending the potential use of DRL in the field of structural engineering and paving the way for future implementations in real-world contexts, even in the presence of high uncertainty in structural characteristics.

5. FINAL REMARKS

5.1 Summary

The work presented in this thesis proposed, developed and validated a comprehensive computational framework for solving the engineering problem of Model Updating in an accurate, automatic and computationally efficient manner, using a combination of Reduced-Order Modelling and Deep Reinforcement Learning, called DRUM-Av2.0 (Deep Reinforcement Learning for Updating Model – Version 2.0). The research began with the recognition of a major challenge in SHM: i.e., the need for accurate numerical representations of complex structures subject to environmental and operational variability, combined with the practical limitations of the high computational demands associated with traditional calibration methods. Conventional MU approaches, whether deterministic or probabilistic, often fail to efficiently handle multidimensional parameter spaces, especially when real-time or large-scale applications are required. DRUM-Av2.0 was developed in response to this limitation, providing an automated, adaptive, data-driven tool capable of learning the parametric configuration that best aligns numerical simulations with experimental measurements in a fully autonomous manner. The proposed methodology is based on the integration of three elements:

a. Structural Health Monitoring for the identification of real target variables, such as natural frequencies and modal shapes, which constitute the experimental basis for updating.
b. A Full-Order Model for generating high-fidelity numerical data that comprehensively define structural dynamics.
c. A Reduced-Order Model which projects the FOM into an efficient, low-dimensional space while preserving its essential dynamic characteristics.

This integration allows the DRL agent to interact with a reduced but accurate representation of the physical system, reducing computation time by several orders of magnitude while preserving accuracy. The ROM thus operates as an intelligent interface between physics-based modelling and data-driven learning, capturing essential

response behaviour with a significantly reduced number of degrees of freedom. The ability to shift the computational load from the FOM to the ROM represents one of the most important advances introduced by DRUM-Av2.0 over its predecessor, DRUM-Av1.0. Within this numerical environment, the DRL engine was implemented based on the TD3 algorithm, an advanced Actor-Critic framework designed specifically to handle continuous, high-dimensional action spaces, such as those encountered in structural problems. The agent training strategy is based on iterative interaction with the numerical environment, driven by a reward function that quantifies the difference between the real and simulated target variables. Through progressive episodes, the agent learned an optimal policy able to autonomously converge towards the set of uncertain parameters that minimises this deviation.

The DRUM-Av2.0 workflow follows a multi-step process:

- Step 1: execution and design of the SHM campaign, ensuring experimental representativeness and defining target variables.
- Step 2: construction of the FOM and generation of the numerical data set covering the engineering parameter domain.
- Step 3: definition and validation of the ROM through order reduction and sensitivity parameterisation, ensuring the fidelity of the reduced dynamics.
- Steps 4-5: integration of the ROM into the DRL engine, training of the intelligent agent.
- Step 6: selection of the final model through a statistical study of multiple convergent solutions.

Subsequently, two case studies are presented in chapters 4.4 and 4.5 that demonstrate the full potential of this approach. The first example, a single-storey RC frame, confirmed the accuracy and stability of the proposed method under conditions of controlled complexity. The second, more complex example, involving a two-storey RC structure with twelve uncertain parameters, highlighted the scalability, generalisation capability and efficiency of the framework. In both studies, the agent achieved convergence towards the optimal solution with two different metrics that reached

values comparable to or better than those of traditional MU methods. In quantitative terms, while the training phase using FOM required up to 15 hours of computation, the ROM-based environment reduced training to less than 20 minutes. The time for each simulation phase decreased from 5 seconds to approximately 0.02 seconds, allowing the agent to perform over 250 times more iterations with the same computational constraints. This improvement accelerates learning and also improves policy robustness, enabling more thorough exploration of the parameter spaces. Statistical validation further confirmed the stability of the trained agent, with CV values consistently below 10%. The trained policy also maintained its predictive validity under slightly modified conditions (variation in actual target variable values less than $\pm 3\%$) without requiring retraining, an essential property for real-world SHM applications. From a methodological point of view, DRUM-Av2.0 also represents a philosophical evolution in model updating: it merges physics-based modelling with machine-driven decision-making, forming a hybrid paradigm in which numerical reasoning and learning coexist. The approach preserves the engineering interpretability of classical modelling, through FOM and ROM definitions, incorporating it into an automated optimisation cycle governed by reinforcement learning. Furthermore, the framework is designed for the development and inclusion of additional data that reduces the space of plausible values of uncertain parameters, for example through the use of prior probabilities of individual uncertain parameters (as in probabilistic methods). However, the performance of the proposed framework still depends on three main factors: (i) the quality and level of detail of the FOM, which determines the accuracy of the ROM; (ii) the design of the dataset, whose size and breadth determine the representativeness of the reduced subspace; (iii) the definition of the reward function and training parameters, which control the balance between exploration, exploitation and numerical stability. Proper calibration of these elements is essential to balance computational efficiency with predictive accuracy.

In conclusion, DRUM-Av2.0 sets a new benchmark for automatic structural model updating, demonstrating how the combination of reduced modelling and deep reinforcement learning can produce accurate, efficient, and self-adaptive procedures. The framework is an intelligent learning process that autonomously refines the

estimation of structural parameters through interaction with a reduced physical model. Its integration into future SHM systems could pave the way for self-updating digital twins capable of adapting in real time, performing predictive diagnostics and making autonomous decisions in the field of structural health monitoring. DRUM-Av2.0, therefore, not only represents the technological outcome of this research work, but also defines a methodological and conceptual basis for the next generation of intelligent structural analysis tools.

5.2 Future research

The work carried out during this research and the results obtained are very promising for addressing one of the most consistent challenges in structural engineering: the accurate calibration of numerical models and the progressive transition to digital twins capable of reflecting, in real time, the real behaviour of the monitored structures. In the process, the integration of data-driven approaches (advanced DRL and ROM techniques) with physical modelling (i.e., FOM) has demonstrated that it is possible to reduce computational complexity while maintaining high levels of accuracy and interpretability. These results allow us to define possible future studies aimed at consolidating and extending this hybrid paradigm, which can act as a basis for the next generation of intelligent structural systems integrated into SHM systems.

One of the most significant areas of development concerns the extension and diversification of the target variables used during model updating. Until now, the use of modal frequencies has provided a solid basis for structural identification; however, limiting the calibration process to this class of data provides only a partial description of dynamic behaviour. In future research, target variables should be expanded to include modal shapes (eigenvectors), modal pairing characteristics, and other quantities derived from in situ tests, such as static response parameters, vibration amplitudes, or deformation measurements. These additional characteristics will expand the information available for the learning process, enabling more accurate calibration at both the global and local levels. By incorporating heterogeneous data sources, the updating process can progressively evolve towards multimodal learning, in which the structural

model is refined based on different levels of experimental evidence, improving its physical consistency and predictive robustness.

Also important is improving the learning agent's adaptability so that it can generalise across multiple structural configurations without requiring complete retraining. Future research will therefore explore methods such as transfer learning, meta-reinforcement learning, and continual learning, which would allow the agent to transfer knowledge from one domain to another and refine its understanding through experience (as demonstrated by recent studies in SHM, as in Giglioni et al., 2024; Yano et al., 2023). This evolution from a task-specific agent to a generalist agent represents a fundamental step towards operationally autonomous systems capable of supporting long-term structural monitoring.

A further development concerns the possibility of designing artificial intelligence agents capable of integrating specific engineering and physics-based knowledge into their decision-making process. These agents can exploit simplified models, physical laws and known structural constraints to guide and limit the search space during learning, significantly improving the efficiency and robustness of model calibration. The formalisation of these constraints in an integrated form with the DRL algorithm would greatly increase the accuracy of the updating process, reducing the possibility of implausible solutions or convergence to undesirable local minima of the reward function. This direction reflects an emerging trend in AI applied to engineering, where the aim is to combine the predictive power of data-driven algorithms with the rigour and established knowledge of physical models. This has led to the creation of hybrid systems, such as Physics-Informed Neural Networks (PINN), which have been the subject of numerous recent studies (Fu et al., 2024; Haywood-Alexander et al., 2025; Yamaguchi & Mizutani, 2024). The future development of agents with integrated engineering awareness thus represents a fundamental step towards expanding DRUM-Av2.0's ability to tackle increasingly complex updating problems, including non-linear systems, variable conditions and degradation phenomena, with improved interpretability, reliability and faster convergence.

From a computational perspective, further improvements in the efficiency and scalability of the training phase may be possible. The adoption of parallel computing architectures, adaptive learning rates and incremental updates could enable the model to operate in near real time without compromising accuracy. This would allow for perfect integration of the approach into real-time SHM networks and facilitate the evolution of continuously auto-updating digital twins.

Finally, the development of mechanisms for quantifying uncertainty within the learning process will be a key aspect in ensuring the reliability and safety of predictive assessments. The production of probabilistic rather than deterministic predictions will make the decision-making process more transparent and compatible with the risk-based management frameworks already adopted in structural engineering.

These guidelines and their application to new case studies will contribute to the evolution of these MU techniques (based on the combined use of FOM and ROM with DRL) from efficient research prototypes to operational digital twin platforms capable of continuously learning from structural responses, adapting to environmental changes, and supporting informed decision-making throughout the entire life cycle of civil infrastructure.

6. ACKNOWLEDGEMENTS

This thesis marks the end of a long and meaningful process, and I am deeply grateful to everyone who has supported me over the years.

First and foremost, I would like to express my sincere gratitude to Prof. G. Uva, my main supervisor, for her guidance, trust and continuous support during my doctoral studies. Her experience and insightful feedback have been fundamental to the development of this research. I am also grateful to Prof. E. Chatzi, my secondary supervisor, for her valuable guidance and for welcoming me during my research period in Switzerland; that experience greatly broadened my horizons and enriched my work.

I would like to thank Prof. V. Iacobellis and Prof. F. Fiorito, outgoing and incoming coordinators of the doctoral programme, for their professionalism, commitment and constant support in ensuring a stimulating and rigorous doctoral environment.

A special and heartfelt acknowledgment goes to Eng. PhD F. Parisi. Fabio, you have been far more than a co-tutor: our collaboration grew into a truly fraternal bond. Your presence, generosity, and constant guidance were fundamental throughout the entire PhD path, both professionally and personally. I am sincerely grateful for everything you have done, and continue to do, for me; your dedication and attitude remain a constant source of inspiration. I also warmly thank Eng. PhD S. Ruggieri for his guidance and friendship. Thank you, Sergio, for always supporting me, encouraging me, and standing by my choices during these years.

To all the PhD students and researchers of the Uva group at Politecnico di Bari, thank you for turning daily work into a shared journey made of discussions, ideas, and genuine human connection. Beyond research, we shared experiences of life, often around a table, which made these years truly memorable.

In particular, I want to thank Ale and Mirko, who from day one have been not only a source of inspiration, but also trusted friends I could always rely on. I also gratefully acknowledge the PhD students and researchers of the Chatzi group in Zurich, for the stimulating environment and for making my time abroad a period rich in learning, collaboration, and growth.

Finally, I owe my deepest gratitude to my family. To my mother and father, and to my brothers: thank you for always believing in me, supporting me with constant love, and encouraging me to pursue this path and my life choices with determination.

To my friends, and to all the people who, sometimes without even realizing it, chose to spend time with me over these years: those moments contributed more than you may know to my growth and to the ideas that accompanied this work.

Thank you sincerely,
Gianluca Bruno

7. LIST OF FIGURES

Figure 1. Performance of structural permanence in the case of preventive or corrective maintenance interventions 2

Figure 2. Structural performance trends and probability distribution of structural uncertainty in structures with and without monitoring + model updating 3

Figure 3. Structural Health Monitoring phases. 10

Figure 4. Categories of band filters for signals: (from left to right) i) low-pass filter; ii) high-pass filter; iii) band-pass filter. 13

Figure 5. Complexity plot: a) real mode, b) complex mode []. 17

Figure 6. Steel structure for road sign (a) and static display of some of its experimentally estimated mode shapes (b–d) [].....19

Figure 7.....28

Figure 8.....29

Figure 9. Internal photos of the building showing structural elements (both RC and masonry) at the ground floor (left) and at the first floor (right).....30

Figure 10. Photos of RC structural elements: (a) circular column; (b) L-shape column; (c) rectangular column supporting the helicoidal staircase; (d) external RC wall.....31

Figure 11. Photos of the data acquisition control unit (left) and of the installed accelerometers (right). 34

Figure 12. Photo of the position of the smart vibration monitoring terminal (left) and photo of the instrument (right). 35

Figure 13. Disposition of all accelerometers and of the data acquisition control unit throughout the building, about a biaxial accelerometer at the first storey and placed in the ceiling and photos about the triaxial accelerometer at the ground storey outside the building (placed in a specific manhole). 36

Figure 14. Comparison in terms of vibration (logarithmic scale) among recorded vibrations at the last storey of sentry building and threshold values for human comfort and building damages, X direction. In the Figure are reported the recordings

before the dredging (2021/10/19), during the dredging (2022/07/11) and after the dredging (2022/09/23). 40

Figure 15. Comparison in terms of vibration (logarithmic scale) among recorded vibrations at the last storey of sentry building and threshold values for human comfort and building damages, Y direction. In the Figure are reported the recordings before the dredging (2021/10/19), during the dredging (2022/07/11) and after the dredging (2022/09/23). 41

Figure 16. Maximum daily values of vibration (i.e., peaks), accounting for all main directions (i.e., X, Y, and Z) and compared with the thresholds for human comfort and building damages. 42

Figure 17. SVD graph using FDD algorithm, by elaborating acceleration data at 2021/10/15 for the sentry building. Red circles indicate the estimated modes of vibration. 43

Figure 18. Graphical rapresentation of modal shapes of the first three modes of vibration for the sentry building 43

Figure 19. Variation (indicated as “var”) of the main first three frequencies (indicated as “f1”, “f2”, and “f3”) during the dredging period, with respect to the output in the reference day (indicated as “ref”). 44

Figure 20. FE model of the building. 46

Figure 21. Modal shapes of the first three modes of vibration for the calibrated FE model of the building. 48

Figure 22. Percentage differences among experimental and numerical frequencies for the main first three frequencies (indicated as “Δf1”, “Δf2”, and “Δf3”) during the dredging period. 49

Figure 23. Interaction between Agent and Environment in DRL 57

Figure 24. General framework of DRUM-Av1.0. 58

Figure 25. Two-dimensional representation of the agent characteristics and environment of the DRL for the MU problem. 65

Figure 26. Structural diagram of the dummy table model 69

Figure 27. Accelerograms applied in the X-direction (top) and Y-direction (bottom) to simulate the presence of white noise on the structure 71

Figure 28. Results obtained from OMA. On the left, the Modal Assurance Criteria values are reported for the first three vibration modes. On the right, the Singular Value Decomposition graph is reported. 71

Figure 29. DRL-engine for MU 73

Figure 30. Action space of DRUM-Av1.0 in the environment NSE. 74

Figure 31. Cumulative reward trend during the training phase. The orange line shows the simple moving average; best, average, and worst episode performances are indicated. 78

Figure 32. Evolution of $\sigma(t)$ and its moving average during the training phase; convergence within the set tolerance. 78

Figure 33. *Results of the DRUM-Av1.0 test phase: (a) cumulative reward trend per episode; (b) deviation $\sigma(t)$ and moving average trend across test episodes.*.....80

Figure 34. *Dependence of deviation $\sigma(t)$ on estimated uncertain parameters: (a) E_1 and E_2 combinations; (b) K_{rx} and K_{rY} combinations.* 81

Figure 35. Values of the simulated frequencies $vs\tau 2 = f1FEM, f2FEM, f3FEM$ obtained for all episodes and values of the real frequencies $vr\tau 2 = [f1SHM, f2SHM, f3SHM]$ imposed in the scenario 81

Figure 36. Pairplot of the relationships between pairs of estimated structural control parameters used [E_1, E_2, K_{rx}, K_{ry}] in the testing phase. On the diagonal of the plot matrix the statistical distribution of the individual parameter. The upper and lower diagonal plots represent the correlation between parameter pairs, shown through a colourmap according to the value of the deviation obtained at the end of each episode 83

Figure 37. Variation of log-likelihood standard deviation with increasing sample size. Likelihood stabilization achieved for $>$16,000 runs. 87

Figure 38. *Overlapping probability density functions (PDFs) of prior (red) and posterior (blue) distributions for uncertain parameters.* 87

Figure 39. Images of the case study bridge 90

Figure 40 Plan and elevation view of the case study bridge, with the distribution of the uniaxial accelerometers along the longitudinal development of the bridge. 91

Figure 41. Models for (left) OMA, developed in Artemis []; (right) MU, developed in SAP2000 []. 92

Figure 42. Cumulative reward trend during the training phase for the case study. 94

Figure 43. Trend of sigma $\sigma(t)$ function and simple mobile average during the training phase at the end of each episode for the case study..... 94

Figure 44. Results of the DRUM-A test and validation phase for the case study: a) cumulative reward trend at the end of each episode and the average cumulative reward during the test phase; b) the trend of the deviation $\sigma(t)$ achieved at the end of each episode and the average deviation during the test phase..... 96

Figure 45. Statistical distributions of E_1, E_2, and E_3 (continuous) and C_1 and C_2 (discrete). The estimated values of CV were equal to 2.85%, 1.27%, and 1.30%, respectively for E_1, E_2, and E_3. 97

Figure 46. General framework of the DRUM-Av2.0..... 108

Figure 47. Geometrical configuration of the first case study 114

Figure 48. Eigenvalues error between ROM and FOM for the first case study 117

Figure 49 Result of training phase for the first case study: (a) trend of the error $\sigma(t)$ with simple moving average in red line; (b) culuative reward for each episode with the simple moving average in orange line. 119

Figure 50. Results of the DRUM-Av2.0 test and validation phase for the case study: a) trend of the deviation achieved at the end of each episode and average deviation during the test phase; b) cumulative reward trend at the end of each episode and average cumulative reward during all test phases..... 120

Figure 51. Pairplot of the results obtained during the agent validation phase. The graphs along the diagonal show the probability distribution of each uncertain parameter $[E1, E2, Krx\ e\ Kry]$. Outside the diagonal, we find the dispersion of correlations

between uncertain parameters, with a colour indicating the respective deviation value $\sigma(t)$. 122

Figure 52. Axonometric representation of the second case study: definition of structural materials (left), denomination of the columns (right) 127

Figure 53 Eigenvalues error between ROM and FOM for the second case study 129

Figure 54. Result of training phase for the second case study: (a) trend of the error $\sigma(t)$ with simple moving average in red line; (b) cumulative reward for each episode with the simple moving average in orange line.....131

Figure 55 Probability distributions for all uncertain parameters. The red dotted line on the individual graphs indicates the mode for each parameter. 133

8. LIST OF TABLES

Table 1. 29

Table 2. Vibration thresholds in acceleration for human comfort, expressed in mm/s^2, according to UNI 9614-17 []. Values are reported for different use destination and different period of the day 38

Table 3. Vibration thresholds in velocity for building damages (at the last storey), expressed in mm/s, according to UNI 9916 (2014) [] and DIN 4150-3 (2016) []. Values are reported for different use destination and for short-term and permanent duration of the vibrations 39

Table 4. Frequencies, damping, and complexity for the first five vibration modes of the sentry building, using environmental excitation of 2021/10/15. 44

Table 5. Comparison in percentage terms among experimental and numerical outputs for constrained models (best solution by varying Em and Ec), first three modes of vibration, using as reference the environmental excitation of 2021/10/15. 47

Table 6. Comparison in percentage terms among experimental and numerical outputs for unconstrained models (best solution by varying Em and Ec), first three modes of vibration, using as reference the environmental excitation of 2021/10/15. 47

Table 7. Characteristics of the dummy case study elements. 68

Table 8. Mechanical characteristics of the materials of structural elements . 69

Table 9. Values of the first three vibration modes for the dummy structure .. 72

Table 10. Multi-discrete action space available for the agent 74

Table 11. *Testing uncertain parameters and corresponding target frequencies.* 79

Table 12. Statistics on the results obtained in the testing phase by DRUM-A for the estimated structural control parameters $pc2$ 82

Table 13. Mode values of the structural control parameter $pc2$ obtained by DRUM-Av1.0 and comparison with real values. 84

Table 14. Sampling step, number of combinations, and estimated time for scenarios O_1, O_2, O_3, and O_4, run through SAP.Py. 84

Table 15. *Results of deterministic MU via SAP.Py for scenarios* O_1, O_2 and O_3. 85

Table 16. *Comparison of DRUM-Av1.0 results with deterministic and probabilistic MU methods.* 88

Table 17. Values of $pu\tau$ for the case study bridge 91

Table 18. Values of the first three vibration modes for the case study structure 92

Table 19. Multi-discrete action space available for the agent for the case study. 93

Table 20 Mode values of the uncertain parameter pu obtained by DRUM-A for the case study and comparison with real values. 96

Table 21. Geometrical properties of the first case study.....113

Table 22. Mechanical properties of the first case study 113

Table 23. Values of the first 7 vibration modes for the first case study 115

Table 24. Comparisons between ROM and FOM for the first case study.....117

Table 25. Statistical analysis on the results obtained in the testing phase by DRUM-Av2.0 with ROM for the first case study.....121

Table 26. Comparison between the most frequent values (mode) obtained through DRUM-Av2.0 and the real values.....122

Table 27. Results in terms of robustness of the trained agent in converging on the solution for target variable values other than those used during training $vr\tau$ 123

Table 28. Comparison between the results obtained with ROM in DRUM-A and the direct use of FOM as an environment for DRUM-A.....125

Table 29. Real values of uncertain parameters $\boldsymbol{pu}$ 126

Table 30. Values of the first ten vibration modes for the dummy structure..128

Table 31. Comparisons between ROM and FOM for the second case study130

Table 32. Statistical analysis on the results obtained in the testing phase by DRUM-Av2.0 with ROM for the second case study 132

Table 33. Comparison in terms of $vr\tau$ between real values and numerical updated values obtained by DRUM-Av2.0. 133

Table 34. Efficiency parameters to evaluate DRUM-Av2.0 on the second case study 135

9. LIST OF ACRONYMS

AI	Artificial Intelligent
ANN	Artificial Newral Network
DOF	Degree of Freedom
DL	Deep Learning
DRL	Deep Reinforcement Learning
DRUM-Av1.0	Deep Reinforcement Learning for Updating Model Approach First version
DRUM-Av2.0	Deep Reinforcement Learning for Updating Model Approach second version
EFDD	Enhanced Frequency Domain Decomposition
FDD	Frequency Domain Decomposition
FEM	Finite Element Model
FOM	Full Order Model
FRF	Frequency Response Functions
GPR	Gaussian Process Regression
MAC	Modal Assurance Criterion
MCP	Modal Phase Collinearity
MCMC	Monte Carlo Markov Chains
MDP	Markov Decision Process
MEMS	Micro-ElectroMechanical Systems
ML	Machine Learning
MPD	Mean Phase Deviation
MU	Model Updating
NM	Numerical Model
NSE	Numerical Software Environment
OMA	Operational Modal Analysis
OSP	Optimisation of Sensor Placement
PCA	Principal Component Analysis
PDF	Probability Density Function

POD	Proper Orthogonal Decomposition
PPO	Proximal Policy Optimization
PSD	Power Spectral Density
RC	Reinforced Concrete
RL	Reinforcement Learning
RMSE	Root Mean Square Error
ROM	Reduced Order Model
SHM	Structural Health Monitoring
SSI	Stochastic Subspace Identification
SSI-DATA	Stochastic Subspace Identification DATA driven version
SSI-COV	Stochastic Subspace Identification COVariance-driven version
SVD	Singular Value Decomposition
TD3	Twin Delayed Deep Deterministic Policy Gradient algorithm
TDD	Time Domain Decomposition
WAPE	Weighted Absolute Percentage Error

10. LIST OF SYMBOLS

γ	Discount factor
δ	Decay factor for tolerance value
$\Delta E_{1(t)}$	Action on the elastic modulus of first structural material at generic timestep
$\Delta E_{2(t)}$	Action on the elastic modulus of second structural material at generic timestep
$\Delta K_{rx(t)}$	Action on elastic constant of the rotational spring constraint in the x-direction generic timestep
$\Delta K_{ry(t)}$	Action on elastic constant of the rotational spring constraint in the y-direction generic timestep
$\Delta\sigma$	Variation of the deviation at the generic timestep and the value at the previous timestep
ε	Total number of episodes
η	Actor in algorithm of proximal policy optimization
$\hat{\theta}$	Optimal values of the numerical model feature
θ	Numerical model parameters
θ^{η}	Parameter of actors deep neural networks
θ^{V}	Parameter of critic deep neural networks
ν	Poisson coefficient
$\pi(s)$ or π	Policy
$\pi^*(s)$ or π^*	Optimal policy
Π	Set of all possible policies
σ	standard deviation
σ_t	Scaled value of the difference between the real target variables and the simulated target variables, called “deviation”
σ_{end}	Value of tolerance of deviation
$\sigma_{end}{}^{Adp}$	Adaptive value of tolerance of deviation
Σ	Diagonal matrix of singular values

ϕ_i	Vector of reduced basis
Φ	Reduced basis or projection matrix
ψ	Value of parametrization
$\omega_{(t)}$ or $\boldsymbol{\omega_{(t)}}$	Observation at generical timestep
$\omega_{(t0)}$	Observation at initial timestep
Ω	Observation space
a	Action performed by the agent
$a_{(t)}$ or $\boldsymbol{a_{(t)}}$	Actions at generical timestep
$\boldsymbol{a(p_{u(t)})}$	Set of action on the each uncertain parameters at generical timestep
$a(p^j_{u(t)})$	Action on the generic uncertain parameter at generical timestep
$a(p_c^j)^{min}$	Maximum action on the generic structural control parameter
$a(p_c^j)^{max}$	Minimum action on the generic structural control parameter
A	Actions space
B	Beams in the case study
c	Maximum number of combinations
C	Dumping matrix
C_r	Reduced order damping matrix
Co	Constant in reward at each timestep
CV	Coefficient of Variation
$\bar{d}$	Measured data
$d(\theta)$	Model output
$\boldsymbol{d_h}$	Value in dataset at generic combination of uncertain parameters
$\boldsymbol{D}$	Dataset from the full order model
e	Episode
E	Elastic modulus
E_1	Elastic modulus of first structural material
E_2	Elastic modulus of second structural material

$\mathbb{E}_{\pi}[\cdot]$	Expectation
$f(t)$	Vector of external forces
f_r	Reduced order vector of external forces
f_1^{SHM}	Frequency of first vibration mode obtained by structural health monitoring techniques
$f_1^{FOM}(h)$	Frequency of first vibration mode obtained by full order model in generic combinations
$f_1^{ROM}(h)$	Frequency of first vibration mode obtained by reduced order model in generic combinations
$f_{1(t)}^{ROM}$	Frequency of first vibration mode obtained by reduced order model at generic timestep
f_2^{SHM}	Frequency of second vibration mode obtained by structural health monitoring techniques
$f_2^{FOM}(h)$	Frequency of second vibration mode obtained by full order model in generic combinations
$f_2^{ROM}(h)$	Frequency of second vibration mode obtained by reduced order model in generic combinations
$f_{2(t)}^{ROM}$	Frequency of second vibration mode obtained by reduced order model at generic timestep
f_3^{SHM}	Frequency of third vibration mode obtained by structural health monitoring techniques
$f_3^{FOM}(h)$	Frequency of third vibration mode obtained by full order model in generic combinations
$f_3^{ROM}(h)$	Frequency of third vibration mode obtained by reduced order model in generic combinations
$f_{3(t)}^{ROM}$	Frequency of third vibration mode obtained by reduced order model at generic timestep
f_4^{SHM}	Frequency of fourth vibration mode obtained by structural health monitoring techniques

$f_4^{FOM}(h)$	Frequency of fourth vibration mode obtained by full order model in generic combinations
$f_4^{ROM}(h)$	Frequency of fourth vibration mode obtained by reduced order model in generic combinations
$f_{4(t)}^{ROM}$	Frequency of fourth vibration mode obtained by reduced order model at generic timestep
f_5^{SHM}	Frequency of fifth vibration mode obtained by structural health monitoring techniques
$f_5^{FOM}(h)$	Frequency of fifth vibration mode obtained by full order model in generic combinations
$f_5^{ROM}(h)$	Frequency of fifth vibration mode obtained by reduced order model in generic combinations
$f_{5(t)}^{ROM}$	Frequency of fifth vibration mode obtained by reduced order model at generic timestep
f_6^{SHM}	Frequency of sixth vibration mode obtained by structural health monitoring techniques
$f_6^{FOM}(h)$	Frequency of sixth vibration mode obtained by full order model in generic combinations
$f_6^{ROM}(h)$	Frequency of sixth vibration mode obtained by reduced order model in generic combinations
$f_{6(t)}^{ROM}$	Frequency of sixth vibration mode obtained by reduced order model at generic timestep
f_7^{SHM}	Frequency of seventh vibration mode obtained by structural health monitoring techniques
$f_7^{FOM}(h)$	Frequency of seventh vibration mode obtained by full order model in generic combinations
$f_7^{ROM}(h)$	Frequency of seventh vibration mode obtained by reduced order model in generic combinations
$f_{7(t)}^{ROM}$	Frequency of seventh vibration mode obtained by reduced order model at generic timestep

f_8^{SHM}	Frequency of eighth vibration mode obtained by structural health monitoring techniques
$f_8^{FOM}(h)$	Frequency of eighth vibration mode obtained by full order model in generic combinations
$f_8^{ROM}(h)$	Frequency of eighth vibration mode obtained by reduced order model in generic combinations
$f_{8(t)}^{ROM}$	Frequency of eighth vibration mode obtained by reduced order model at generic timestep
f_9^{SHM}	Frequency of nineth vibration mode obtained by structural health monitoring techniques
$f_9^{FOM}(h)$	Frequency of nineth vibration mode obtained by full order model in generic combinations
$f_9^{ROM}(h)$	Frequency of nineth vibration mode obtained by reduced order model in generic combinations
$f_{9(t)}^{ROM}$	Frequency of nineth vibration mode obtained by reduced order model at generic timestep
f_{10}^{SHM}	Frequency of tenth vibration mode obtained by structural health monitoring techniques
$f_{10}^{FOM}(h)$	Frequency of tenth vibration mode obtained by full order model in generic combinations
$f_{10}^{ROM}(h)$	Frequency of tenth vibration mode obtained by reduced order model in generic combinations
$f_{10(t)}^{ROM}$	Frequency of tenth vibration mode obtained by reduced order model at generic timestep
G	Shear modulus
h	Generic number of combination
j	Generic number of the uncertain parameters
k	Generic number of the simulated target variables
K	Stiffness matrix
K_r or $\boldsymbol{K_r}$	Reduced order stiffness matrix

K_{rx}	Value of the elastic constant of the rotational spring constraint in the x-direction
K_{ry}	Value of the elastic constant of the rotational spring constraint in the y-direction
$\boldsymbol{K}(\boldsymbol{p_u}(0))$	Stiffness matrix at reference combination of uncertain parameters
$\boldsymbol{K}(\boldsymbol{h})$ or $\boldsymbol{K}(\boldsymbol{p_u}(\boldsymbol{h}))$	Stiffness matrix at generic combination of uncertain parameters
l	Maximum number of the simulated target variables
L	Orthogonal left matrix
L_i	Left singular vectors
L_r	Last reduced left singular vectors
m	Number of timestep in each episode
M	Mass matrix
M_r or $\boldsymbol{M_r}$	Reduced order mass matrix
$\boldsymbol{M}(\boldsymbol{h})$	Mass matrix at generic combination of uncertain parameters
M_1	First type of structural materials
M_2	Second type of structural materials
M_3	Third type of structural materials
n	Maximum number of the uncertain parameters
n_{DOF}	Maximum number of degree of freedom
N	Dimension of the system
p	Penalization in reward at each timestep
$p(\theta\|\bar{d},\mathcal{M})$	Posterior probability
$p(\theta\|\mathcal{M})$	Prior Probability Density Function
$p(\bar{d}\|\theta,\mathcal{M})$	Likelihood function
$p\left(\bar{d}\|\mathcal{M}\right)$	Normalization function
$\boldsymbol{p_u}$	Uncertain parameters
$\boldsymbol{p_u}(\boldsymbol{h})$	Generic combination of uncertain parameters
$\boldsymbol{p_u}(0)$	Reference combination of uncertain parameters

$\boldsymbol{p_{u(t)}}$	Uncertain parameters at generical timestep
$\boldsymbol{p_{u(t0)}}$	Uncertain parameters at initial timestep
$p^{j}_{u(t)}$	Generic uncertain parameter at generical timestep
$\widehat{\boldsymbol{p_u}}$	Estimated optimal values of the uncertain parameters
P	Columns in the case study
$Q^{\pi}(s,a)$	Q-value function or Action-value function according to the policy
$Q^{*}(s,a)$	Optimal Q-Value functio
r	Dimension of the reduced system
$r(\theta)$	Residual, differences between measured and numerical data
$r_{(t)}$	Reward at generical timestep
R	Reward function
s	State of the environment
$s_{(t0)}$	Initial state at initial timestep
$s_{(t)}$ or $\boldsymbol{s_{(t)}}$	State of the environment at generical timestep
S	State space of the environment
S_m	Snapshot matrix
t_j	Generic point in time
t or (t)	Generical timestep
t_0	Initial timestep
T	Transition probability function
$u(t)$	Vector of nodal displacements
$u(t_j)$	Vector of nodal displacements at generic point in time
$u(h)$ or $\boldsymbol{u(h)}$	Vector of nodal displacements at generic combination of uncertain parameters
$u_r(t)$	Reduced vector of nodal displacements
$\dot{u}(t)$	Vector of nodal pseudo-velocities
$\ddot{u}(t)$	Vector of nodal pseudo-accelerations
v_1	Singular value
$\boldsymbol{v_r^{\tau}}$	Real target variables

$v_r^{\tau k}$	Generic real target variable
$\boldsymbol{v_{FOM}^{\tau}(h)}$	Full order model target variables at generic combination of uncertain parameters
$\boldsymbol{v_{FOM}^{\tau}}$	Full order model target variable
$\boldsymbol{v_{ROM}^{\tau}}$	Reduced order model target variables
$v_{ROM}^{\tau k}$	Generic target variable obtained by reduced order model
$\boldsymbol{v_{ROM}^{\tau}(h)}$	Reduced order model target variables at generic combination of uncertain parameters
$\boldsymbol{v_{ROM(t)}^{\tau}}$	Reduced order model target variables at generical timestep
V	Orthogonal right matrix
$V^{\pi}(s)$	V-value function or State-Value function
$V^{*}(s)$	Optimal V-value function
Ws	Specific weight
x, y	Generic number of degree of freedom
X	Direction of the reference axis of the numerical model
Y	Direction of the reference axis of the numerical model
Z	Direction of the reference axis of the numerical model

BIBLIOGRAPHY

Farrar, C. R., and Worden, K. (2007). An introduction to structural health monitoring. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 365(1851), 303-315. https://doi.org/10.1098/rsta.2006.1928

Giagopoulos, D. et al. (2019). Structural health monitoring and fatigue damage estimation using vibration measurements and finite element model updating. *Structural Health Monitoring*, *18*(4), 1189-1206. https://doi.org/10.1177/1475921718790188

Brincker, L. et al. (2020) 'Modal identification from ambient responses using frequency domain decomposition' in: Proceedings of the International Modal Analysis Conference (IMAC), pp. 625–630. San Antonio, Texas, USA, February 7-10

Rainieri, C. and Fabbrocino, G. (2014)'Operational Modal Analysis of Civil Engineering Structures', vol. 142, Springer, New York, p. 143.

Frangopol, D. M., and Messervey, T. B. (2009) 'Maintenance principles for civil structures', *Encyclopedia of structural health monitoring*. https://doi.org/10.1002/9780470061626.shm108

Aloisio, A. et al. (2022). Indirect assessment of concrete resistance from FE model updating and Young's modulus estimation of a multi-span PSC viaduct: Experimental tests and validation. *Structures*, *37*, 686–697. https://doi.org/10.1016/j.istruc.2022.01.045

Avci, O. et al. (2022). Operational modal analysis and finite element model updating of a 230 m tall tower. *Structures*, *37*, 154–167. https://doi.org/10.1016/j.istruc.2021.12.078

Aloisio, A. et al. (2020). Dynamic identification and model updating of an eight-storey CLT building. *Engineering Structures*, *213*, 110593. https://doi.org/10.1016/j.engstruct.2020.110593

Kurent, B., Brank, B. and Ao, W. K. (2021). Model updating of seven-storey cross-laminated timber building designed on frequency-response-functions-based modal testing. *Structure and Infrastructure Engineering*, *19*(2), 178–196. https://doi.org/10.1080/15732479.2021.1931893

Cabboi, A., Gentile, C. and Saisi, A. (2017). From continuous vibration monitoring to FEM-based damage assessment: Application on a stone-masonry tower. *Construction and Building Materials*, *156*, 252–265. https://doi.org/10.1016/j.conbuildmat.2017.08.160

Lye, A., Cicirello, A. and Patelli, E. (2021). Sampling methods for solving Bayesian model updating problems: A tutorial. *Mechanical Systems and Signal Processing*, *159*, 107760. https://doi.org/10.1016/j.ymssp.2021.107760

Behmanesh, I. et al. (2015). Hierarchical Bayesian model updating for structural identification. *Mechanical Systems and Signal Processing*, *64–65*, 360–376. https://doi.org/10.1016/j.ymssp.2015.03.026

Rocchetta, R. et al. (2017). On-line Bayesian model updating for structural health monitoring. *Mechanical Systems and Signal Processing*, *103*, 174–195. https://doi.org/10.1016/j.ymssp.2017.10.015

Yuan, Y. et al. (2022). Active learning structural model updating of a multisensory system based on Kriging method and Bayesian inference. *Computer-Aided Civil and Infrastructure Engineering*, *38*(3), 353–371. https://doi.org/10.1111/mice.12822

Padil, K. H., Bakhary, N. and Hao, H. (2016). The use of a non-probabilistic artificial neural network to consider uncertainties in vibration-based-damage detection. *Mechanical Systems and Signal Processing*, *83*, 194–209. https://doi.org/10.1016/j.ymssp.2016.06.007

Sabamehr, A., Lim, C. and Bagchi, A. (2018). System identification and model updating of highway bridges using ambient vibration tests. *Journal of Civil Structural Health Monitoring*, *8*(5), 755–771. https://doi.org/10.1007/s13349-018-0304-5

Naranjo-Pérez, J. et al. (2020). A collaborative machine learning-optimization algorithm to improve the finite element model updating of civil engineering structures. *Engineering Structures*, *225*, 111327. https://doi.org/10.1016/j.engstruct.2020.111327

Fu, B., Gao, Y. and Wang, W. (2024). A physics-informed deep reinforcement learning framework for autonomous steel frame structure design. *Computer-Aided Civil and Infrastructure Engineering*. https://doi.org/10.1111/mice.13276

Bleker, L., Tam, K-M. M. and D'Acunto, P. (2024) "A Graph-Based Grammar for Structural Design using Deep Reinforcement Learning." Proceedings of the IASS 2024 Symposium. https://app.iass2024.org/files/IASS_2024_Paper_299.pdf

Morato, P. et al. (2023). Inference and dynamic decision-making for deteriorating systems with probabilistic dependencies through Bayesian networks and deep reinforcement learning. *Reliability Engineering & System Safety*, *235*, 109144. https://doi.org/10.1016/j.ress.2023.109144

Arcieri, G. et al. (2024). POMDP inference and robust solution via deep reinforcement learning: an application to railway optimal maintenance. *Machine Learning*. https://doi.org/10.1007/s10994-024-06559-2

Ruggieri, S. et al. (2024). Assessing the dredging vibrational effects on surrounding structures: The case of port nourishment in Bari. *Journal of Building Engineering*, *96*, 110385. https://doi.org/10.1016/j.jobe.2024.110385

UNI 9614, Misura delle vibrazioni negli edifici e criteri di valutazione del disturbo, 2017 (in Italian).

ISO 2631/1, Mechanical vibration and shock Evaluation of human exposure to whole-body vibration, Part 1: General requirements (1997).

ISO 2631/2, Mechanical vibration and shock Evaluation of human exposure to whole-body vibration, Part 2: Vibration in buildings (1 Hz to 80 Hz) (2003).

UNI 9916, Criteri di misura e valutazione degli effetti delle vibrazioni sugli edifici, 2014 (in Italian).

ISO 4866, Mechanical vibration and shock Vibration of fixed structures, Guidelines for the measurement of vibrations and evaluation of their effects on structures (2010).

DIN 4150-3. Le Vibrazioni Nelle Costruzioni: Effetti Sui Manufatti, 2016 (in Italian, translated from German).

Bruno, G. et al. (2025). An intelligent agent-based method for model updating of structures with multiple uncertain parameters via deep reinforcement learning. *Mechanical Systems and Signal Processing*, *234*, 112832. https://doi.org/10.1016/j.ymssp.2025.112832

Barsocchi, P. et al. (2020). Wireless sensor networks for continuous structural health monitoring of historic masonry towers. *International Journal of Architectural Heritage*, *15*(1), 22–44. https://doi.org/10.1080/15583058.2020.1719229

Zonno, G. et al. (2018). Automated long-term dynamic monitoring using hierarchical clustering and adaptive modal tracking: validation and applications. *Journal of Civil Structural Health Monitoring*, *8*(5), 791–808. https://doi.org/10.1007/s13349-018-0306-3

Kita, A. et al. (2021). A new method for earthquake-induced damage identification in historic masonry towers combining OMA and IDA. *Bulletin of Earthquake Engineering*, *19*(12), 5307–5337. https://doi.org/10.1007/s10518-021-01167-0

Betti M. et al. (2022) European Workshop on Structural Health Monitoring. Springer; Cham, Switzerland22. Long-Term Structural Monitoring of a Steel Jacket Offshore Platform. Validation of Meteo-Marine Data and Implications for Maintenance; pp. 1038–1047

Morello R., De Capua C. and Meduri A. (2010) Remote monitoring of building structural integrity by a smart wireless sensor network; Proceedings of the IEEE Instrumentation & Measurement Technology Conference Proceedings; Austin, TX, USA. 3–6 May 2010; pp. 1150–1154

Jornet-Monteverde, J. A., Galiana-Merino, J. J., and Soler-Llorens, J. L. (2021). Design and implementation of a wireless sensor network for seismic monitoring of buildings. *Sensors*, *21*(11), 3875. https://doi.org/10.3390/s21113875

Sindhuja S. and Kevildon J.S.J. (2015) MEMS-based wireless sensors network system for post-seismic tremor harm evaluation and building monitoring; Proceedings of the International Conference Circuits, Power Computing Technologies; Nagercoil, India. 19–20 March 2015; pp. 1–4.

Sivasuriyan, A. et al. (2021). Practical implementation of structural health monitoring in Multi-Story buildings. *Buildings*, *11*(6), 263. https://doi.org/10.3390/buildings11060263

Riggio, M., and Dilmaghani, M. (2019). Structural health monitoring of timber buildings: a literature survey. *Building Research & Information*, *48*(8), 817–837. https://doi.org/10.1080/09613218.2019.1681253

Johnson, T. J. et al. (2003). Distributed structural health monitoring with a smart sensor array. *Mechanical Systems and Signal Processing*, *18*(3), 555–572. https://doi.org/10.1016/s0888-3270(03)00002-5

Uva, G. et al. (2014). Structural monitoring using fiber optic sensors of a pre-stressed concrete viaduct during construction phases. *Case Studies in Nondestructive Testing and Evaluation*, *2*, 27–37. https://doi.org/10.1016/j.csndt.2014.06.002

Bado, M. F., and Casas, J. R. (2021). A review of recent distributed optical fiber sensors applications for civil Engineering structural health monitoring. *Sensors*, *21*(5), 1818. https://doi.org/10.3390/s21051818

Sabato, A., Niezrecki, C., and Fortino, G. (2016). Wireless MEMS-Based Accelerometer Sensor Boards for Structural Vibration Monitoring: A review. *IEEE Sensors Journal*, *17*(2), 226–235. https://doi.org/10.1109/jsen.2016.2630008

Matarazzo, T. J. et al. (2018). Crowdsensing framework for monitoring bridge vibrations using moving smartphones. *Proceedings of the IEEE*, *106*(4), 577–593. https://doi.org/10.1109/jproc.2018.2808759

Guzman-Acevedo, G. M. et al. (2019). GPS, accelerometer, and smartphone fused smart sensor for SHM on Real-Scale bridges. *Advances in Civil Engineering*, *2019*(1). https://doi.org/10.1155/2019/6429430

Sun, H., and Büyüköztürk, O. (2015). Optimal sensor placement in structural health monitoring using discrete optimization. *Smart Materials and Structures*, *24*(12), 125034. https://doi.org/10.1088/0964-1726/24/12/125034

Ostachowicz, W., Soman, R., and Malinowski, P. (2019). Optimization of sensor placement for structural health monitoring: a review. *Structural Health Monitoring*, *18*(3), 963–988. https://doi.org/10.1177/1475921719825601

Kammer, D.C. (1991) Sensor placement for on-orbit modal identification and correlation of large space structures. J. Guid. Control Dyn. 14 251–9

Papadimitriou, C. (2004). Optimal sensor placement methodology for parametric identification of structural systems. *Journal of Sound and Vibration*, *278*(4–5), 923–947. https://doi.org/10.1016/j.jsv.2003.10.063

Castro-Triguero, R. et al. (2013). Robustness of optimal sensor placement under parametric uncertainty. *Mechanical Systems and Signal Processing*, *41*(1–2), 268–287. https://doi.org/10.1016/j.ymssp.2013.06.022

Meo, M., and Zumpano, G. (2005). On the optimal sensor placement techniques for a bridge structure. *Engineering Structures*, *27*(10), 1488–1497. https://doi.org/10.1016/j.engstruct.2005.03.015

Zini, G., Betti, M. and Bartoli, G. (2022). A pilot project for the long-term structural health monitoring of historic city gates. *Journal of Civil Structural Health Monitoring*, *12*(3), 537–556. https://doi.org/10.1007/s13349-022-00563-7

Bendat, J. S. and Allan, G. P. (2010). Random Data: Analysis and Measurement Procedures, Fourth edition. In *Wiley eBooks*. http://oa.lib.ksu.edu.tw/OA//handle/987654321/183418

Rainieri, C. and Fabbrocino, G. (2014). Operational modal analysis of civil engineering structures. In *Springer eBooks*. https://doi.org/10.1007/978-1-4939-0767-0

Brincker, R. and Larsen, J. A. (2007). Obtaining and Estimating Low Noise Floors in Vibration Sensors. *Proc XXV International Modal Analysis Conference*. Orlando, FL. https://vbn.aau.dk/da/publications/obtaining-and-estimating-low-noise-floors-in-vibration-sensors

Gharehbaghi, V. R. et al. (2022). A critical review on structural health monitoring: Definitions, methods, and perspectives. *Archives of computational methods in engineering*, *29*(4), 2209-2235.

Zonta, D., Pozzi, M. and Glisic, B., (2014). Preprocessing Techniques for SHM Acceleration Data: Filtering and Detrending Approaches in Large Datasets. Sensors, 14(4), pp. 7196–7214.

Kim, B. H., Stubbs and N., & Park, T. (2004). A new method to extract modal parameters using output-only responses. *Journal of Sound and Vibration*, *282*(1–2), 215–230. https://doi.org/10.1016/j.jsv.2004.02.026

Van Overschee, P. and De Moor, B. (1996). Continuous-time frequency domain subspace system identification. *Signal Processing*, *52*(2), 179–194. https://doi.org/10.1016/0165-1684(96)00052-7

Peeters, B. and De Roeck, G. (1999). REFERENCE-BASED STOCHASTIC SUBSPACE IDENTIFICATION FOR OUTPUT-ONLY MODAL ANALYSIS. *Mechanical Systems and Signal Processing*, *13*(6), 855–878. https://doi.org/10.1006/mssp.1999.1249

Qin, S., Kang, J. and Wang, Q. (2016). Operational Modal Analysis Based on Subspace Algorithm with an Improved Stabilization Diagram Method. *Shock and Vibration*, *2016*, 1–10. https://doi.org/10.1155/2016/7598965

Reynders, E. et al. (2015). Uncertainty quantification in operational modal analysis with stochastic subspace identification: Validation and applications. *Mechanical Systems and Signal Processing*, *66–67*, 13–30. https://doi.org/10.1016/j.ymssp.2015.04.018

Li, W. et al. (2017). Extraction of modal parameters for identification of time-varying systems using data-driven stochastic subspace identification. *Journal of Vibration and Control*, *24*(20), 4781–4796. https://doi.org/10.1177/1077546317734670

Figueiredo, E. et al. (2010). Influence of the autoregressive model order on damage detection. *Computer-Aided Civil and Infrastructure Engineering*, *26*(3), 225–238. https://doi.org/10.1111/j.1467-8667.2010.00685.x

Gul, M. and Catbas, F. N. (2009). Statistical pattern recognition for Structural Health Monitoring using time series modeling: Theory and experimental verifications. *Mechanical Systems and Signal Processing*, *23*(7), 2192–2204. https://doi.org/10.1016/j.ymssp.2009.02.013

Zahid, F. B., Ong, Z. C. and Khoo, S. Y. (2020). A review of operational modal analysis techniques for in-service modal identification. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, *42*(8). https://doi.org/10.1007/s40430-020-02470-8

Brincker, R. and Zhang, L. (2009). Frequency Domain Decomposition Revisited. *Proceedings of the 3rd International Operational Modal Analysis Conference -*

IOMAC,. http://findresearcher.sdu.dk/portal/da/publications/frequency-domain-decomposition-revisited(0d17d4f0-5658-11de-b5e8-000ea68e967b).html

Brincker, R., Ventura, C. E. and Andersen, P. (2021). Damping Estimation by Frequency Domain Decomposition. *Proceedings of IMAC 19: A Conference on Structural Dynamics, Orlando Florida*, 698–703. https://vbn.aau.dk/da/publications/damping-estimation-by-frequency-domain-decomposition(2bcee690-9c2e-11db-8ed6-000ea68e967b)/export.html

Allemang, R.J. and Brown, D.L. (1982). A correlation coefficient for modal vector analysis, in: Proceedings of the 1st International Modal Analysis Conference, pp. 110–116. Orlando, 8-10 November.

Ewins, D. J. (2009). *Modal testing: theory, practice, and application*. John Wiley & Sons.

Pappa, R. S., Elliott, K. B. and Schenk, A. (1993). Consistent-mode indicator for the eigensystem realization algorithm. *Journal of Guidance Control and Dynamics*, *16*(5), 852–858. https://doi.org/10.2514/3.21092

Heylen, W. and Sas, P. (2005). Modal Analysis Theory and Testing. *PMA KU-Leuven*

Svendsen, B. T. et al. (2021). Improved finite element model updating of a full-scale steel bridge using sensitivity analysis. *Structure and Infrastructure Engineering*, *19*(3), 315–331. https://doi.org/10.1080/15732479.2021.1944227

Lee, Y. et al. (2023). Structural damage detection using deep learning and FE model updating techniques. *Scientific Reports*, *13*(1). https://doi.org/10.1038/s41598-023-46141-9

Fernández-Godino, M. G. (2023). Review of multi-fidelity models. *Advances in Computational Science and Engineering*, *1*(4), 351–400. https://doi.org/10.3934/acse.2023015

Conti, P. et al. (2024). Multi-fidelity reduced-order surrogate modelling. *Proceedings of the Royal Society a Mathematical Physical and Engineering Sciences*, *480*(2283). https://doi.org/10.1098/rspa.2023.0655

Agathos, K. et al. (2021). Parametric reduced order models for output-only vibration-based crack detection in shell structures. *Mechanical Systems and Signal Processing*, *162*, 108051. https://doi.org/10.1016/j.ymssp.2021.108051

Vlachas, K. et al. (2025). Reduced Order Modeling conditioned on monitored features for response and error bounds estimation in engineered systems. *Mechanical Systems and Signal Processing*, *226*, 112261. https://doi.org/10.1016/j.ymssp.2024.112261

Deshmukh, A. P. and Allison, J. T. (2017). Design of dynamic systems using surrogate models of derivative functions. *Journal of Mechanical Design*, *139*(10). https://doi.org/10.1115/1.4037407

Torzoni, M., Manzoni, A. and Mariani, S. (2023). A multi-fidelity surrogate model for structural health monitoring exploiting model order reduction and artificial neural networks. *Mechanical Systems and Signal Processing*, *197*, 110376. https://doi.org/10.1016/j.ymssp.2023.110376

Ereiz, S., Duvnjak, I. and Jiménez-Alonso, J. F. (2022). Review of finite element model updating methods for structural applications. *Structures*, *41*, 684–723. https://doi.org/10.1016/j.istruc.2022.05.041

Hofmeister, B., Bruns, M. and Rolfes, R. (2019). Finite element model updating using deterministic optimisation: A global pattern search approach. *Engineering Structures*, *195*, 373–381. https://doi.org/10.1016/j.engstruct.2019.05.047

Aloisio, A. and Alaggio, R. (2021). Experimental Estimation of the Elastic Modulus of Concrete Girders from Drive-By Inspections with Force-Balance Accelerometers. *Shock and Vibration*, *2021*(1). https://doi.org/10.1155/2021/1617526

Yuan, Z. et al. (2018). Parameter selection for model updating with global sensitivity analysis. *Mechanical Systems and Signal Processing*, *115*, 483–496. https://doi.org/10.1016/j.ymssp.2018.05.048

Zhu, T. et al. (2017). Sensitivity-Based finite element model updating using dynamic condensation approach. *International Journal of Structural Stability and Dynamics*, *18*(08), 1840004. https://doi.org/10.1142/s0219455418400047

Titurus, B. and Friswell, M. I. (2007). Regularization in model updating. *International Journal for Numerical Methods in Engineering*, *75*(4), 440–478. https://doi.org/10.1002/nme.2257

Kamali, S. et al. (2024). Inverse surrogate model for deterministic structural model updating based on random forest regression. *Mechanical Systems and Signal Processing*, *215*, 111416. https://doi.org/10.1016/j.ymssp.2024.111416

Bruno, G. et al. (2024) SAP.py: automated model updating using SAP2000 OAPI and python for structural health monitoring. Proceeding of the 18th World Conference on Earthquake Engineering (WCEE).

Lye, A., Cicirello, A. and Patelli, E. (2021b). Sampling methods for solving Bayesian model updating problems: A tutorial. *Mechanical Systems and Signal Processing*, *159*, 107760. https://doi.org/10.1016/j.ymssp.2021.107760

Jia, X. et al. (2022). Nonlinear model updating through a hierarchical Bayesian modeling framework. *Computer Methods in Applied Mechanics and Engineering*, *392*, 114646. https://doi.org/10.1016/j.cma.2022.114646

Beck, J. L. and Yuen, K. (2004). Model selection using response measurements: Bayesian Probabilistic approach. *Journal of Engineering Mechanics*, *130*(2), 192–203. https://doi.org/10.1061/(asce)0733-9399(2004)130:2(192

Ierimonti, L. et al. (2020). An innovative continuous Bayesian model updating method for base-isolated RC buildings using vibration monitoring data. *Mechanical Systems and Signal Processing*, *139*, 106600. https://doi.org/10.1016/j.ymssp.2019.106600

Beck, J. L. and Au, S. (2002). Bayesian Updating of Structural Models and Reliability using Markov Chain Monte Carlo Simulation. *Journal of Engineering Mechanics*, *128*(4), 380–391. https://doi.org/10.1061/(asce)0733-9399(2002)128:4(380

Jang, J. and Smyth, A. (2017). Bayesian model updating of a full-scale finite element model with sensitivity-based clustering. *Structural Control and Health Monitoring*, *24*(11), e2004. https://doi.org/10.1002/stc.2004

Monchetti, S. et al. (2023). Comparison between Bayesian updating and approximate Bayesian computation for model identification of masonry towers through

dynamic data. *Bulletin of Earthquake Engineering*, *22*(7), 3491–3509. https://doi.org/10.1007/s10518-023-01670-6

Chowdhury, R. and Adhikari, S. (2012). Fuzzy parametric uncertainty analysis of linear dynamical systems: A surrogate modeling approach. *Mechanical Systems and Signal Processing*, *32*, 5–17. https://doi.org/10.1016/j.ymssp.2012.05.002

Haag, T., González, S. C. and Hanss, M. (2011). Model validation and selection based on inverse fuzzy arithmetic. *Mechanical Systems and Signal Processing*, *32*, 116–134. https://doi.org/10.1016/j.ymssp.2011.09.028

Sakiyama, F. I. et al. (2022). Quantifying the extent of local damage of a 60-year-old prestressed concrete bridge: A hybrid SHM approach. *Structural Health Monitoring*, *22*(1), 496–517. https://doi.org/10.1177/14759217221079295

Tran-Ngoc, H. et al. (2021). Finite element model updating of a multispan bridge with a hybrid metaheuristic search algorithm using experimental data from wireless triaxial sensors. *Engineering With Computers*, *38*(S3), 1865–1883. https://doi.org/10.1007/s00366-021-01307-9

Malekloo, A. et al. (2021). Machine learning and structural health monitoring overview with emerging technology and high-dimensional data source highlights. *Structural Health Monitoring*, *21*(4), 1906–1955. https://doi.org/10.1177/14759217211036880

Zeng, J., Xue, K. and Chen, H. (2025). Real-time probabilistic model updating and damage detection using Machine Learning-based likelihood-free inference. *Mechanical Systems and Signal Processing*, *230*, 112612. https://doi.org/10.1016/j.ymssp.2025.112612

Ghahramani, Z. (2015). Probabilistic machine learning and artificial intelligence. *Nature*, *521*(7553), 452–459. https://doi.org/10.1038/nature14541

Esteghamati, M. Z. et al. (2025). Beyond development: Challenges in deploying Machine-Learning models for structural engineering applications. *Journal of Structural Engineering*, *151*(6). https://doi.org/10.1061/jsendh.steng-13301

Vrtač, T. et al. (2023). A hybrid modeling strategy for training data generation in machine learning-based structural health monitoring. *Mechanical Systems and Signal Processing*, *207*, 110937. https://doi.org/10.1016/j.ymssp.2023.110937

Acunzo, G. et al. (2017). Modal mass estimation from ambient vibrations measurement: A method for civil buildings. *Mechanical Systems and Signal Processing*, *98*, 580–593. https://doi.org/10.1016/j.ymssp.2017.05.014

Structural Vibration Solutions, ARTeMIS Modal Pro. User Manual, 2023. Denmark. https://www.svibs.com/artemis-modal-pro/

Computer and Structures, Inc (2024) SAP2000, Advanced 25.Structural Analysis Program-Manual – Computer and Structures, Inc, Berkeley, California, USA. Available at: https://www.csiamerica.com/products/sap2000

MIT, Ministero delle Infrastrutture e dei Trasporti (2018) *Aggiornamento delle <<Norme tecniche per le costruzioni>>*.

François-Lavet, V. et al. (2018). An introduction to deep reinforcement learning. *Foundations and Trends® in Machine Learning*, *11*(3–4), 219–354. https://doi.org/10.1561/2200000071

Sutton, R.S. and Barto., A.G. (2018) Reinforcement Learning, Second Edition: An Introduction. *MIT Press*, Cambridge, MA.

Fuentes, R. et al. (2020). Structural health monitoring and damage identification. In *Springer eBooks* (pp. 1–72). https://doi.org/10.1007/978-1-4939-6503-8_23-1

Li, H. et al. (2014). Reviews on innovations and applications in structural health monitoring for infrastructures. *Structural Monitoring and Maintenance*, *1*(1), 1–45. https://doi.org/10.12989/smm.2014.1.1.001

Bathe, K. (2008). Finite element method. *Wiley Encyclopedia of Computer Science and Engineering*, 1–12. https://doi.org/10.1002/9780470050118.ecse159

Aliabadi, F. M. H. (2020). Boundary element Methods. In *Springer eBooks* (pp. 182–193). https://doi.org/10.1007/978-3-662-55771-6_18

Bićanić, N. (2004). Discrete Element Methods. *In Encyclopedia of Computational Mechanics (Eds E. Stein, R. Borst and T.J.R. Hughes)*. https://doi.org/10.1002/0470091355.ecm006.pub2

Aradi, S. (2020). Survey of Deep Reinforcement Learning for Motion Planning of Autonomous Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, *23*(2), 740–759. https://doi.org/10.1109/tits.2020.3024655

Andrieu, C. et al. (2003). An introduction to MCMC for Machine learning. *Machine Learning*, *50*(1/2), 5–43. https://doi.org/10.1023/a:1020281327116

Abdi, J. and Moshiri, B. (2013). Application of temporal difference learning rules in short-term traffic flow prediction. *Expert Systems*, *32*(1), 49–64. https://doi.org/10.1111/exsy.12055

Wen, S. et al. (2022). A Hierarchical Path Planning Approach with Multi-SARSA Based on Topological Map. *Sensors*, *22*(6), 2367. https://doi.org/10.3390/s22062367

Schulman, J. et al. (2017). Proximal Policy optimization Algorithms. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1707.06347

Mnih, V. et al. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533. https://doi.org/10.1038/nature14236

Lillicrap, T. P. et al. (2015). Continuous control with deep reinforcement learning. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1509.02971

Haarnoja, T. et al. (2018). Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1801.01290

Mangalathu, S. et al. (2021). Explainable machine learning models for punching shear strength estimation of flat slabs without transverse reinforcement. *Journal of Building Engineering*, *39*, 102300. https://doi.org/10.1016/j.jobe.2021.102300

Aronhime, S. et al. (2013). DCE-MRI of the liver: Effect of linear and nonlinear conversions on hepatic perfusion quantification and reproducibility. *Journal of Magnetic Resonance Imaging*, *40*(1), 90–98. https://doi.org/10.1002/jmri.24341

Python Software Foundation (2024) *Python: A Dynamic, Open-Source Programming Language*. Available at: https://www.python.org/

Kerschen, G. et al. (2005). The Method of Proper orthogonal Decomposition for dynamical characterization and order reduction of mechanical Systems: An Overview. *Nonlinear Dynamics*, *41*(1–3), 147–169. https://doi.org/10.1007/s11071-005-2803-2

Khodaparast, H. H., Mottershead, J. E. and Friswell, M. I. (2008). Perturbation methods for the estimation of parameter variability in stochastic model updating. *Mechanical Systems and Signal Processing*, *22*(8), 1751–1773. https://doi.org/10.1016/j.ymssp.2008.03.001

Brincker, R., Zhang, L. and Andersen, P. (2001). Modal identification of output-only systems using frequency domain decomposition. *Smart Materials and Structures*, *10*(3), 441–445. https://doi.org/10.1088/0964-1726/10/3/303

Fujimoto, S., Van Hoof, H. and Meger, D. (2018). Addressing function approximation error in Actor-Critic methods. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1802.09477

Portelas, R. et al. (2020) Automatic curriculum learning for deep rl: A short survey. *arXiv preprint arXiv:2003.04664*. https://doi.org/10.48550/arXiv.2003.04664

Xu, Z. et al. (2019). Learning an adaptive learning rate schedule. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1909.09712

Giglioni, V. et al. (2024). Transfer learning in bridge monitoring: Laboratory study on domain adaptation for population-based SHM of multispan continuous girder bridges. *Mechanical Systems and Signal Processing*, *224*, 112151. https://doi.org/10.1016/j.ymssp.2024.112151

Yano, M. O. et al. (2023). Foundations and applicability of transfer learning for structural health monitoring of bridges. *Mechanical Systems and Signal Processing*, *204*, 110766. https://doi.org/10.1016/j.ymssp.2023.110766

Haywood-Alexander, M. et al. (2025). Response estimation and system identification of dynamical systems via physics-informed neural networks. *Advanced*

Modeling and Simulation in Engineering Sciences, *12*(1). https://doi.org/10.1186/s40323-025-00291-9

Yamaguchi, T. and Mizutani, T. (2024). A Physics-Informed neural network for the nonlinear damage identification in a reinforced concrete bridge pier using seismic responses. *Structural Control and Health Monitoring*, *2024*(1). https://doi.org/10.1155/2024/5532909

11. CURRICULUM



Gianluca Bruno

Date of Birth: 08/12/1996
Nationality: Italian
Phone: +39 3280817730
Email: g.bruno15@phd.poliba.it; bruno.gianluca.gb@gmail.com
LinkedIn: linkedin.com/in/gianluca-bruno-aa7662b0
ResearchGate: https://www.researchgate.net/profile/Gianluca-Bruno-2
ORCiD: https://orcid.org/0009-0009-6965-3126

EDUCATION

2022 - Current	Ph.D., Structural Engineering, Ph.D. Course of Risk and Environ-mental, Territorial and Building Development, Polytechnic Uni-versity of Bari
2025	Visiting Ph.D., Eidgenössische Technische Hochschule Zürich, Department of Civil, Environmental and Geomatic Engineering, Supervisor: Prof Eleni Chatzi, Zürich, Switzerland
2022	Master of Science in Civil Engineering, Structures, 110/110 Cum Laude, Polytechnic University of Bari
2019	Bachelor of Science in Civil Engineering, 100/110, Polytechnic University of Bari
2015	High School Diploma, Technical and Economic Institute “Cassandro-Fermi-Nervi”, Barletta, Italy

WORK EXPERIENCE

2022 - Current	Structural Heath Monitoring Analyst, Consorzio Fabre, Poly-technic University of Bari
May 2022 – Oct 2022	Research fellow: in Structural health monitoring of buildings and infrastructure, Design and simulation of control systems for the transport of dangerous goods, Polytechnic University of Bari

Nov 2021 – Dic 2021	Curricular training Technical Studio Eng. M. Verderosa, Bari, Italy. Inspections, surveys and measurements on building sites and graphic restitution using CAD software (autocad and revit). Thermophysical modelling using Logical Soft's Termolog Epix software and analysis of the results obtained
Jun 2021 – Dec 2021	Tutoring service, Polytechnic University of Bari. Student support activities, for reception, university events, teaching activities, administrative procedures, mediation between students and lecturers, support and orientation proposals.
Feb 2019 – Apr 2019	Laboratory training, Polytechnic University of Bari. Research and experimental activities for the mechanical characterisation and analysis of the stresssoftening problem of highly damped elastomers
Jul 2015 – Jan 2016	Training Technical Studio - Spazio A, Andria, Italy. Survey and graphic restitution activities. Support activities for property estimates and appraisals

PUBLICATIONS

Authorship convention: for student-authored publications, the student is the first author, and the advisor is the last author. For other authorship positions or larger collaborative pa-pers, authorship order denotes contribution level.

Archival Journal Publications

- *Ruggieri, S., **Bruno, G**., Attolico, A., & Uva, G. (2024). Assessing the dredging vibrational effects on surrounding structures: The case of port nourishment in Bari. Journal of Building Engineering, 96, 110385. https://doi.org/10.1016/j.jobe.2024.110385*
- ***Bruno, G**., Ruggieri, S., & Uva, G. (2025). Structural Health Monitoring of an Existing Complex Structure: From Experimental Data. Seismic Behaviour and Design of Irregular and Complex Civil Structures V, 53, 226.*
- ***Bruno, G**., Parisi, F., Ruggieri, S., & Uva, G. (2025). An intelligent agent-based method for model updating of structures with multiple uncertain parameters*

via deep reinforcement learning. Mechanical Systems and Signal Processing, 234, 112832. https://doi.org/10.1016/j.ymssp.2025.112832

Other Publications

- ***Bruno, G**., Ruggieri, S., & Uva, G. (2023, October). Structural Health Monitoring of an Existing Complex Structure: From Experimental Data to Model Updating. In European Workshop on the Seismic Behaviour of Irregular and Complex Structures (pp. 226-236). Cham: Springer Nature Switzerland.*
- ***Bruno, G**., Ruggieri, S., Nettis, A., Parisi, F., & Uva, G. (2024). SAP. py: automated model updating using SAP2000 OAPI and python for structural health monitoring. In Proceeding of the 18th World Conference on Earthquake Engineering (WCEE).*
- ***Bruno, G**., Parisi, F., Ruggieri, S., & Uva, G. On the use of Deep Reinforcement Learning for online model updating of structures with multiple uncertain parameters. In Proceeding of the 10th International Conference on Computational Methods in Structural Dynamics and Earthquake Engineering (COMPDYN 2025)*
- *Ruggieri, S., **Bruno, G.**, Parisi, F., & Uva, G. (2025, August). Model updating of structures with uncertain multiple parameters: from traditional trial-and-error approaches to autonomous intelligent agents. In Multimodal Sensing and Artificial Intelligence for Sustainable Future (Vol. 13570, pp. 144-150). SPIE.*

PRESENTATIONS

- **"***SAP. py: automated model updating using SAP2000 OAPI and python for structural health monitoring", 18th World Conference on Earthquake Engineering. Mi-lan, Italy, June 30 – July 5 2024.*
- Seminar "NEW APPROACHES FOR MODEL UPDATING OF STRUCTURES USING AUTOMATED INTELLIGENT ALGORITHMS", D-BAUG Department, ETH Zurich, 18 March 2025.
- *"On the use of Deep Reinforcement Learning for online model updating of structures with multiple uncertain parameters", COMPDYN 2025, 10th*

International Conference on Computational Methods in Structural Dynamics and Earthquake Engineering, Rhodes, Greece, 15-18 June 2025.

TEACHING EXPERIENCE

Course: “Costruzioni in zona sismica”, SSD ICAR/09, Academic Year 2022-2023. Role: supervisor for the project part of the course.

- Course: “Tecnica delle costruzioni”, SSD ICAR/09, Academic Year 2022-2023. Role: supervisor for the project part of the course.
- Course: “Costruzioni in zona sismica”, Academic Year 2023-2024, Role: supervi-sor for the project part of the course.
- Course: “Tecnica delle costruzioni”, Academic Year 2023-2024. Role: supervisor for the project part of the course.

ADVISING

- Giosuè Liberato, MS, “DYNAMIC IDENTIFICATION OF AN EXISTING REINFORCED CONCRETE BRIDGE: A CASE STUDY”, 2024. Co-supervisor of the thesis project.
- Rossella Fasan, MS, “DYNAMIC STRUCTURAL HEALTH MONITORING BASED ON ENVIRONMENTAL VIBRATION MEASUREMENTS: A CASE STUDY”, 2023. Co-supervisor of the thesis project.

LANGUAGE SKILLS

- talian
- English, Level B1

DIGITAL SKILLS

- Structural engineering: SAP2000,ARTEMIS, CSI Bridge, PRO_SAP, Straus7, Midas Gen
- Programming skills: Python, Matlab
- GIS: QGIS
- Graphics: AutoCad 3d, Rhinoceros, Revit, Photoshop, Illustrator
- Other: Microsoft Office, Windows, macOS