

Intelligent control systems for wind-induced vibration mitigation in tall buildings

by

Khalid Mohammad Abdelaziz Mohammad Abdelaziz

B.S., Cairo University, 2009

M.S., Cairo University, 2014

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the requirements for the degree

DOCTOR OF PHILOSOPHY

Alan Levin Department of Mechanical and Nuclear Engineering
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2022

Abstract

Wind-induced vibration (WIV) is a major contributor to the high investment cost of tall buildings. WIV can render a building unserviceable and strongly affect occupant comfort. It can also damage internal building components and may even cause catastrophic failure in severe cases. According to most design codes, tall buildings must be designed to endure severe wind events with return periods as long as 50 years, which requires additional construction material and design effort. Tuned mass/liquid dampers (TMDs/TLDs) have been designed and employed to reduce WIV effects. They rely on using energy dissipation mechanisms, such as oil flow through constrictions, to absorb WIV energy from the host structure. TMDs and TLDs are narrowband devices which are not adaptable to different wind characteristics. They also tend to consume space in higher floors where real estate is usually more expensive. To inhibit even the initial formation of WIV, aerodynamic modifications (AMs) have been used both independently and in conjunction with TMDs and TLDs. By changing the external shape of the building, the wind system attacking the building can be altered so the building becomes less susceptible to WIV. Major AMs change the overall shape of the building by introducing tapers, setbacks and helix features. Alternatively, minor AMs introduce corner or edge changes such as rounding, fillets or cut-offs. The literature review presented herein concludes that AMs have the potential to reduce WIV accelerations by 30-60%. However, AMs that are effective at reducing WIV accelerations at certain wind conditions may inadvertently increase those accelerations depending on the wind characteristics and the surrounding environment. This work focuses on designing and optimally controlling active aerodynamic modification systems that can change their orientation to guarantee consistent WIV reduction across different wind conditions and environmental changes. A central aim to the designs and techniques presented in this work is to

generate versatile controllers that can work across different building and AM designs, wind characteristics and surrounding environments while minimizing user input and deployment effort. To achieve these goals various machine learning models and techniques are used because they generally support a black box design approach. An AM design using four plates installed on the corners of the building is given special attention in this work. The plates can rotate and hold position around their own axes of symmetry. This design is emphasized for its simplicity, ease of implementation and the minimal number of control parameters. However, the techniques developed herein can also be used for different AM designs. To simulate the WIV response at different wind conditions and plate orientations, this work uses (1) a fluid-solid-interaction simulation model developed specifically to enable plate rotation and building motion in the simulation domain, (2) wind tunnel experiments of a 1:400 scaled robotic aeroelastic building model and (3) large-scale wind tunnel experiments of a 1:73 scaled robotic aeroelastic building model. These models are controlled using (1) a wind condition averaging control approach using a developed iterative optimum training technique proven herein to use a substantially reduced number of experiments to obtain optimal training and (2) The combination of a distributed sensor network and recurrent neural networks to fully define the instantaneous system state using time series inputs. The final version of the control scheme developed in this work uses a genetic reinforcement learning method that enhances the exploration capacity of reinforcement learning and makes it more likely to find global optimum solutions. The results indicate that utilizing the developed designs and control techniques can reduce WIV response by 40-90% depending on the wind condition. The developed controller was able to minimize WIV in randomly varying wind conditions even for conditions it did not specifically encounter during training.

Intelligent control systems for wind-induced vibration mitigation in tall buildings

by

Khalid Mohammad Abdelaziz Mohammad Abdelaziz

B.S., Cairo University, 2009

M.S., Cairo University, 2014

A DISSERTATION

submitted in partial fulfillment of the requirements for the degree

DOCTOR OF PHILOSOPHY

Alan Levin Department of Mechanical and Nuclear Engineering
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2022

Approved by:

Major Professor
Jared D. Hobeck

Copyright

© Khalid Abdelaziz 2022.

Abstract

Wind-induced vibration (WIV) is a major contributor to the high investment cost of tall buildings. WIV can render a building unserviceable and strongly affect occupant comfort. It can also damage internal building components and may even cause catastrophic failure in severe cases. According to most design codes, tall buildings must be designed to endure severe wind events with return periods as long as 50 years, which requires additional construction material and design effort. Tuned mass/liquid dampers (TMDs/TLDs) have been designed and employed to reduce WIV effects. They rely on using energy dissipation mechanisms, such as oil flow through constrictions, to absorb WIV energy from the host structure. TMDs and TLDs are narrowband devices which are not adaptable to different wind characteristics. They also tend to consume space in higher floors where real estate is usually more expensive. To inhibit even the initial formation of WIV, aerodynamic modifications (AMs) have been used both independently and in conjunction with TMDs and TLDs. By changing the external shape of the building, the wind system attacking the building can be altered so the building becomes less susceptible to WIV. Major AMs change the overall shape of the building by introducing tapers, setbacks and helix features. Alternatively, minor AMs introduce corner or edge changes such as rounding, fillets or cut-offs. The literature review presented herein concludes that AMs have the potential to reduce WIV accelerations by 30-60%. However, AMs that are effective at reducing WIV accelerations at certain wind conditions may inadvertently increase those accelerations depending on the wind characteristics and the surrounding environment. This work focuses on designing and optimally controlling active aerodynamic modification systems that can change their orientation to guarantee consistent WIV reduction across different wind conditions and environmental changes. A central aim to the designs and techniques presented in this work is to

generate versatile controllers that can work across different building and AM designs, wind characteristics and surrounding environments while minimizing user input and deployment effort. To achieve these goals various machine learning models and techniques are used because they generally support a black box design approach. An AM design using four plates installed on the corners of the building is given special attention in this work. The plates can rotate and hold position around their own axes of symmetry. This design is emphasized for its simplicity, ease of implementation and the minimal number of control parameters. However, the techniques developed herein can also be used for different AM designs. To simulate the WIV response at different wind conditions and plate orientations, this work uses (1) a fluid-solid-interaction simulation model developed specifically to enable plate rotation and building motion in the simulation domain, (2) wind tunnel experiments of a 1:400 scaled robotic aeroelastic building model and (3) large-scale wind tunnel experiments of a 1:73 scaled robotic aeroelastic building model. These models are controlled using (1) a wind condition averaging control approach using a developed iterative optimum training technique proven herein to use a substantially reduced number of experiments to obtain optimal training and (2) The combination of a distributed sensor network and recurrent neural networks to fully define the instantaneous system state using time series inputs. The final version of the control scheme developed in this work uses a genetic reinforcement learning method that enhances the exploration capacity of reinforcement learning and makes it more likely to find global optimum solutions. The results indicate that utilizing the developed designs and control techniques can reduce WIV response by 40-90% depending on the wind condition. The developed controller was able to minimize WIV in randomly varying wind conditions even for conditions it did not specifically encounter during training.

Table of Contents

List of Figures	xi
List of Tables	xiv
Acknowledgements	xv
Dedication	xvii
Chapter 1 - Introduction.....	1
1.1. Literature Review	1
1.1.1. Wind-induced vibration	1
1.1.2. Tuned mass/liquid dampers	3
1.1.3. Aerodynamic modification	4
1.1.4. Surrogate-guided and expensive optimization.....	6
1.1.5. Gated recurrent units	8
1.1.6. Reinforcement Learning	9
1.2. Scope and Organization	10
Chapter 2 - Numerical and Experimental Methods	13
2.1 Standard CAARC Building	13
2.2. Fluid-Solid-Interaction Simulation	14
2.2.1. Fluid Model.....	14
2.2.2. Rigid Body Model.....	16
2.2.3. Fluid-Structure-Interaction.....	18
2.2.4. Validation.....	19
2.2.5. Results and Discussion	22
2.3. Small-scale Wind Tunnel Experiments	25
2.3.1. GA-assisted aeroelastic model design.....	27
2.3.2. Active AM plates	29
2.3.3. Cyber-physical instrumentation	30
2.3.4. Results and Discussion	31
2.4. Ongoing development of a large-scale aeroelastic model	31
2.4.1. Model Design.....	33
2.4.2. Active AM designs	34

2.4.3. Results and Discussion	35
2.5. Conclusions.....	37
Chapter 3 - Steady State Optimum Control	39
3.1 Controller Design.....	39
3.2. Genetic Algorithm Training.....	41
3.2.1. Offline optimization.....	41
3.2.2. Optimization results	44
3.2.3. Simulation-based training	45
3.2.4. Controller Simulation.....	47
3.3. Iterative Optimum Training	49
3.3.1. Implementation	52
3.3.2. Numerical benchmark	54
3.3.3. Experiment-based training	57
3.3.4. Results and Discussion	60
3.3.4.1. IOT Performance	60
3.3.4.2 Optimum performance	62
3.3.4.3. Control response time	63
3.4. Conclusions.....	65
Chapter 4 - Genetic Reinforcement Learning Control Using GRUs and Distributed Sensor Networks.....	69
4.1. Experimental Setup.....	71
4.1.1. Sensor network.....	73
4.1.2. Cyber-physical setup.....	74
4.2. Controller Design.....	74
4.2.1. Encoder-decoder GRU	74
4.2.2. Genetic reinforcement learning algorithm	76
4.3. Results and Discussion	78
4.3.1. Encoder-decoder GRU prediction.....	78
4.3.2. Controller performance	79
4.4. Conclusions.....	85
Chapter 5 - Conclusions and Recommendations	88

5.1. Recommendations for Future Work	92
5.1.1. Economic analysis	92
5.1.2. Alternate AM designs	93
5.1.3. Genetic reinforcement learning (GRL)	93
5.1.4. Full-scale training	94
5.1.5. Integration with other WIV mitigation methods.....	94
References	95
Appendix A - Code Highlights	102
Appendix A.1. Fluid-Solid-Interaction Simulation	102
Appendix A.2. Mesh Generator	104
Appendix A.3. Genetic Algorithms	104
Appendix A.4. Iterative optimum training.....	106
Appendix A.5. Cyber-physical system	106
Appendix A.5.1. Large-scale experiments.....	109
Appendix B - List of Acronyms.....	111

List of Figures

Figure 2.1. Layout of the CAARC building model showing the full-scale floor dimensions, DOFs of interest, the convention for measuring the wind AOA and the rotation angles $\alpha 1 - 4$ of the four plates attached to the corners.....	15
Figure 2.2. The 2D computational mesh of the four-plate CAARC building case showing the computational mesh region of the corner plates and the rotating interpolation interface for plate 1	17
Figure 2.3. Motion of mesh points relative to the displacement of the moving region (R).....	20
Figure 2.4. Comparison between the normalized cross-wind displacement PSDs generated by the developed two-dimensional simulation (this study) and by experimental data from (Tang and Kwok 2004).....	23
Figure 2.5. PSD of cross-wind forces obtained from the FSI simulation of the CAARC scaled model.....	24
Figure 2.6. Vortex street formed behind the CAARC building at 0.8 m/s wind speed for (a) horizontal plates and (b) plates tilted 45°	24
Figure 2.7. A detailed drawing of the scaled aeroelastic building model showing the main components of the model in addition to a photo of the assembled model with the skins removed at the top floors	28
Figure 2.8. First order sway mode shapes for the scaled aeroelastic model obtained using GA optimization.	29
Figure 2.9. Components of the plate rotation mechanism used to implement active AM.....	30
Figure 2.10. Cyber-physical instrumentation of the scaled CAARC model consisting of data collection and plate control systems.	32
Figure 2.11. A picture of the large-scale CAARC building model with 9 lumped masses connected by four columns. The back envelope panels are removed to show internal details.	36
Figure 2.12. One floor of the large-scale CAARC model showing the floor dimensions and the horizontal fin-louver active AM design and the convention for measuring its orientation angle α	36

Figure 3.1. Steady state control architecture for the wind-building-plate system assuming a performance function Q	41
Figure 3.2. Convergence trends for the genetic algorithm optimization process showing the optimum plate configurations for different generations	44
Figure 3.3. Comparison between the suboptimal and the optimal deflections and forces for all 3 DOFs	46
Figure 3.4. Comparison of the along-wind building deflection and wind force PSDs before and after plate angle optimization.....	46
Figure 3.5. Result of predictor MLP training showing actual and predicted along-wind deflection amplitudes for various combinations of wind velocity and plate angles	48
Figure 3.6. Along-wind deflection changes as a result of employing the developed controller...	50
Figure 3.7. The MLP Q_w, a_i network used to implement the IOT procedure.	53
Figure 3.8. Flow chart of the IOT technique for minimizing Equation 3.2 while simultaneously training the MLP.	55
Figure 3.9. Performance comparisons of the tested online training techniques.	58
Figure 3.10. Layout of the upstream buildings added to increase the RMS accelerations at the diagonal critical direction ($AOA=56^\circ$)	59
Figure 3.11. RMS acceleration trends for all tested wind conditions obtained after two IOT iterations ($\star$ with the upstream buildings illustrated in Figure 3.10).....	61
Figure 3.12. RMS acceleration prediction performance of the MLP trained using the IOT procedure.....	62
Figure 3.13. RMS acceleration prediction performance of the MLP trained using the IOT procedure.....	64
Figure 3.14. Comparison between along-wind acceleration PSDs of best (green) and worst (blue) plate configurations for the three wind directions considered ($\star$ case with upstream buildings as illustrated in Figure 3.10).....	65
Figure 3.15. Reduction of building accelerations due to changing the plate angles to the optimum values ($AOA = 90^\circ, Vr = 9.7$). The IOT-based controller was inactive until $t=56s$ for this demonstration.....	66
Figure 4.1. A top view of the wind tunnel experiment setup showing: (1) the 1:400 scaled aeroelastic model centered in the width of the tunnel with four rotating plates installed at its	

corners; (2) the measurement conventions for the DOFs of interest (x , y and θ), plate angles ($\alpha_1 \dots \alpha_4$) and the wind direction U ; and (3) the upstream building and the pressure sensor locations and shape.	72
Figure 4.2. A schematic showing the various components of the cyber-physical setup for the wind tunnel experiments. The python script controls all aspects of the online training/control experiment.....	75
Figure 4.3. A schematic diagram of the encoder-decoder GRU model indicating the sensors inputs $\delta_0 \dots \delta_n$, outputs $\delta_{n+1} \dots \delta_{n+m}$ and the hidden state S	75
Figure 4.4. The modified RL procedure indicating how the previous plate angles A_p , GRU state S_p and acceleration sum Q_p (Equation 4.2) are used to train the policy MLPs (π_k). The policy MLPs are then used to determine the future plate angles A_f using the future GRU state S_f	78
Figure 4.5. A comparison between the predicted and actual building rotational acceleration θ for 16 minutes and 7.5 seconds of data.	80
Figure 4.6. Evolution of the controller WIV mitigation performance during 10 iterations of the GRL algorithm subjected to randomly generated wind speed profiles compared to baseline data created with 90 combinations of static (not controlled) plate angles and random wind speed profiles	82
Figure 4.7. Two different cases where the trained controller automatically responded to the increase in the rotational acceleration θ (resulting from the change in wind speed U) by changing the plate angles which immediately reduced the rotational acceleration	83
Figure 4.8. A comparison of the RMS rotational acceleration (20-second window) for the same randomly varying wind speed profile when the controller is turned off and on.....	84
Figure 4.9. The effect of changing the wind speed profile on the controller performance Q compared with the performance of the evaluated random plate angles.....	85

List of Tables

Table 2.1. Parameters of the FSI simulation model validation case involving the CAARC model without the plates	22
Table 2.2. Dynamic similarity parameters for the scaled aeroelastic model (subscripts $\mathbf{m}$ = model and $\mathbf{f}$ = full-scale building).....	27
Table 2.3 List of natural frequencies and damping ratios of the scaled model.	32
Table 2.4. Dynamic similarity parameters for the large-scale aeroelastic model (subscripts $\mathbf{m}$ = model and $\mathbf{f}$ = full-scale building).....	35
Table 2.5 Natural frequencies of the large-scale model.	37
Table 3.1. Parameters of the compared techniques training and optimization techniques	58
Table 3.2. IOT technique parameters employed for the wind tunnel training process.	60
Table 4.1: GRL algorithm parameters for WIV control of the 1:400 scaled building model.....	80

Acknowledgements

Pursuing my Ph.D. was a journey like no other. The knowledge I gained and the self-growth that I went through was beyond my wildest thoughts and dreams. I'm very thankful for everyone who lead me to and through this path.

Many thanks to my advisor, Dr. Jared D. Hobeck, for all the ways in which he inspired and helped me widen my gaze and gain deeper understanding. Before I worked with Dr. Hobeck, I had limited experience in fluid simulation and scientific experiments, but he made these aspects of my experience grow and flourish with his guidance, inquiries, encouragement, patience, support and funding. Many of the ideas in this dissertation are the result of our constructive discussions and meetings along the years. He was patient when I wouldn't implement his advice as timely as I should and gave me the space to discover my shortcomings and evolve my solutions. I'm also thankful for the fun times, the conference travels and the Florida WoW experiments and all the little times.

Many thanks to the advisory committee members for their continued support, advise and teaching effort, and for their kind words praising my effort and the overall quality of my work. Their input helped me develop and present this work with better form and content. The classes I took with each of the committee members helped shape my knowledge benefitting both me and the development of this dissertation. Many thanks to my past advisory committee member, Dr. Warren N. White, who passed away while my dissertation was still in progress, he was an incredible teacher who made infinite effort to widen my gaze and inspire me to investigate and develop my knowledge.

Many thanks to my master's degree advisor, Dr. Sayed M. Metwalli for all his guidance, support and teaching. Thanks for how he helped me shape my confidence and hone my skill. Many thanks to all my other professors, advisors and teachers along the years of my academic studies during my bachelors, masters and doctoral studies all the same.

Many thanks to my father for his support and motivation, for his prayers and unconditional love and appreciation, for all he taught me throughout my life and how he believed me and made me believe in myself. Many thanks to my mother who passed away a long time ago and left me with the memory of her infinite effort to help me learn and provide me with the best opportunities she could afford. I'm still reaping the benefits of her advice to join mechanical engineering to this day and forward. Many thanks to my wife and my son for their presence in my life, for their love and support, for all the laughs, the hikes and the beautiful memories. Many thanks to my sisters for their infinite love and prayers. How they had me in their minds and their hearts and supported me along the years of my studies, and how they made me feel home. Many thanks to my lab mates, friends and colleagues who cared for me and listened to me when I was frustrated or worried or lost or broken. You lent me listening ears and kind words that kept me going in my worst moments.

Many thanks to any who taught me or inspired me to learn and grow. To any who told me to not be too hard on myself and accepted me for who I am. For all who gave me second chances and found the excuses for my mistakes. I could not have done it without any of you. I'm so thankful for having you all in my life.

Dedication

To my parents.

Chapter 1 - Introduction

The number of tall (> 150 m) and super tall (> 300 m) buildings is increasing rapidly around the world to accommodate increasing population densities and land value. Wind-induced vibration (WIV) is a major consideration in tall building design. The Tacoma Narrows bridge collapse in 1940 is a prominent example of the destruction that WIV can cause when it is not considered sufficiently in the design. Most building codes require tall buildings to endure severe wind events that may happen once every 50 years (Love et al. 2018; Jafari and Alipour 2021). The additional materials and design effort required to comply with the codes constitutes a major contribution to the increased investment cost. In addition to major catastrophic failure, damage to building components, such as walls and façade panels, can also result from WIV (Love et al. 2018; Micheli et al. 2018). In recent years, occupant comfort and building serviceability due to WIV has become an area of major research interest. Human-centered design methods have been employed to establish acceleration serviceability and perception limits (Jafari and Alipour 2021).

In this introduction, Section 1.1 provides a literature review that covers the causes and mechanisms as well as the state-of-the-art solutions of WIV. The review also covers the computational methods used throughout this work to develop intelligent WIV mitigation controllers. Section 1.2 defines the scope and organization of this work guided by the findings and challenges discussed in the literature review.

1.1. Literature Review

1.1.1. Wind-induced vibration

When the wind attacks a bluff body, such as a tall building, separation of the wind flow happens at the sharp corners or edges of the body (Sarpkaya 2004; Besem et al. 2016). These

separations may cause many fluid-solid-interaction (FSI) phenomena, e.g., turbulent buffeting, flutter and vortex shedding to occur (Parkinson 1989). The total WIV response of the building depends on the characteristics of the attacking wind (e.g., turbulence intensity, speed and direction distributions) and the building geometry (Sarpkaya 2004; Hobeck and Inman 2012). The environment surrounding the building, such as other structures and geographical features, strongly affects the wind flow by inducing vortex formations and altering the wind characteristics, which has a strong impact on the WIV response (Tang and Kwok 2004).

Vortex shedding is one of the most notable FSI phenomenon. The separation of the flow causes the formation of circular flow patterns or *vortices*. These vortices eventually *shed* as they pass the building at a frequency that is proportional to the wind speed (Hobeck and Inman 2015; Hobeck and Inman 2016). Besem et al. (2016) conducted a study that focuses on elastic vortex-induced vibration (VIV) response. The study used an FSI simulation model to estimate the VIV response of an elastically supported cylinder. A resonance condition where the vortex-induced vibration (VIV) response is greatly amplified occurs when the vortex shedding frequency synchronizes with the structural natural frequency of the cylinder (Hobeck and Inman 2016; Besem et al. 2016). The resulting one-frequency system is said to be in *lock-in* state and the cylinder experiences aeroelastic limit cycle oscillation (Besem et al. 2016). Tall buildings are shown in the literature to show similar response characteristics (Abdelaziz et al. 2021). As will be noted in Section 1.1.3, the implication of these findings is that successful WIV mitigation techniques inhibit the formation of synchronized vortices to avoid lock-in (Sharma et al. 2018).

Momtaz et al. (2017) used a mathematical model combining the structure of the ACT skyscraper in Japan, the wind flow and the tuned mass damper (TMD) to study the effects of vortices on the skyscraper's WIV response (Momtaz et al. 2017). The model was solved for different excitation profiles and frequencies with and without taking the vortex effects into account. The authors concluded that (1) vortex effects can increase the WIV response by up to 40% and (2) TMDs perform well when the excitation force frequency is close to the natural frequency of the structure. For excitations with frequencies far away from the natural frequency, the TMD performs poorly and can even increase the WIV response. These findings mean that an effective WIV mitigation system has to (1) account for complex wind flow patterns (such as vortex shedding) and (2) offer a form of adaptability to provide consistent performance across different wind conditions.

Tang and Kwok (2004) conducted a wind tunnel experimental study to investigate the interference excitation mechanisms between two identical tall buildings. The models were placed at various separation distances and in different positions in the wake region relative to each other. The study concluded that the dynamic WIV response of both buildings was generally enhanced by the presence of the other. This interference effect was found to be more significant when the buildings were closely spaced, which is a common case in most high-density urban environments.

1.1.2. Tuned mass/liquid dampers

Tuned Mass Damper (TMD) systems have been thoroughly analyzed and tested in various deployments (Tuan and Shang 2014; Momtaz et al. 2017). It involves hanging a large mass from the building like a pendulum and attaching it to a set of dampers which absorb energy

from the host structure as it vibrates. Another similar approach is the Tuned Liquid Damper (TLD) system where a rigid water tank is placed on top of the building so as the sloshing motion of the water through constrictions in the tank will absorb the vibration energy (Love et al. 2018; Samanta and Banerji 2010). TLDs operate more efficiently at higher amplitudes and require considerable roof space (Samanta and Banerji 2010). Therefore, various enhancements and modifications were considered to improve their efficiency (Gardarsson et al. 2001; Tait et al. 2005). TMD/TLD systems generally consume space at upper floors where real estate is most expensive. As discussed in Section 1.1.1, Momtaz et al. (2017) found that TMD/TLD systems lack adaptability, so they can increase the vibration response when the excitation force frequencies are located away from the structural natural frequencies. This emphasizes the need for an adaptable WIV mitigation system. Another drawback of TMD/TLD systems that was observed in the results from Momtaz et al. (2017) is the large amount of time the TMD system requires to dissipate the vibration energy, which can render the system ineffective for mitigating fast-varying wind effects.

1.1.3. Aerodynamic modification

Various works in the literature establish that aerodynamic modifications (AMs) of a building can have a profound effect on its wind-induced vibration (WIV) response (Sharma et al. 2018; Abdelaziz et al. 2021). Major AMs consist of changes to the overall geometry of the building such as taper, setbacks or helix shapes. Minor AMs consist of chamfers, cuts or rounding at the building corners without changing the overall geometry. Sharma et al. (2018) presented a comprehensive review of AMs applied to tall buildings. The authors concluded that minor AMs could result in 30-60% reduction in WIV. It was also noted that minor AMs can have adverse effects depending on the local environmental conditions such as wind flow

characteristics and surrounding structures (Sharma et al. 2018). A common trait of successful AMs listed by the authors was the inhibition of synchronized vortex formation to prevent aeroelastic lock-in.

Elshaer et al. (2017) performed a simulation-based aerodynamic corner minimization of the drag and lift coefficients while considering wind directionality and turbulence. The authors reported up to a 30% reduction in both the along- and cross-wind responses using two-surface chamfering at the corners (Elshaer et al. 2017). Xie and Yang (2019) presented a case study of a 270 m-high building with plate fairings attached to its corners at a fixed orientation of 45° (Xie and Yang 2019). The fairings were deployed manually when the weather forecast predicted severe wind conditions. The authors reported an 18% reduction on structural design wind loads when using the fairings.

Hu et al. (2017) performed a wind tunnel experimental study to evaluate the performance of a double-skin façade system with different configurations. All the proposed configurations were found to have negligible effects on the along-wind response of the building. However, some of the configurations were found to significantly reduce the cross-wind response (Hu et al. 2017). Yang et al. (2020) studied buildings with various vertical plate configurations on their perimeter. The authors studied 8 different configurations which vary the vertical plate number, location and continuity. The authors reported a maximum reduction of 28.6% in the along-wind peak force when using one of the tested configurations. A different configuration resulted in a maximum reduction of 40% in the cross-wind peak force (Yang et al. 2020). It is noted that

different vertical plate configurations have different benefits for different WIV directions, which emphasizes the need for a configuration or a system that can combine these benefits.

These studies establish the effectiveness of passive aerodynamic modifications in WIV control for tall buildings, but also emphasize in different contexts that different flow characteristics and environments require different modifications to maintain reduced WIV and prevent adverse effects (Sharma et al. 2018; Xie and Yang 2019; Yang et al. 2020). This calls for the development of dynamic or active AMs to provide consistent WIV mitigation performance.

1.1.4. Surrogate-guided and expensive optimization

Approximating the relationship between the wind conditions, building external geometry and surrounding environment on one side and the building accelerations on the other is a core component to the steady-state control presented in Chapter 3. These relationships are difficult to predict because they display strong nonlinearity and non-monotonicity where, in certain conditions, small changes to the geometry can cause large changes to the wind-building system (Abdelaziz et al. 2021; Abdelaziz and Hobeck 2022). Many physical systems can display such behavior, which called for the development of methods to model, optimize and control them. Population-based optimization techniques (e.g., genetic algorithms, particle swarm, ant colony, etc.) are widely used in the literature because they typically do not require special knowledge of the system being optimized (e.g., mathematical form, derivatives, etc.). These techniques are also known to consume a large number of experiments or simulations, which may be impractical for many reasons including excessive monetary and/or time requirements. Methods are developed in the literature to tackle such *expensive optimization* problems, including both population-based and direct methods (Sergeyev et al. 2018; Paulavičius et al. 2020). The basic concept is to

consume the least possible number of function evaluations even at the expense of increased computational overhead of the method itself (Haftka et al. 2016). However, these methods usually focus on optimizing the system rather than generating a prediction model for it, which is needed in the current study to perform the control (Regis and Shoemaker 2007). As discussed in later sections, this work uses genetic algorithms (GAs) and DNNs to perform both functions simultaneously, while using the DNN as a surrogate model to guide the dispatch of additional function evaluations.

GAs are inspired by biological genetic processes found in nature. They have been used with great success in many studies in the literature (Abdelaziz et al. 2016; Li et al. 2019). This optimization strategy is generally able to operate on discontinuous and multimodal systems; however, GAs typically require a large number of experiments. Artificial neural networks (ANNs) are inspired by the operation of the human brain. DNNs are ANNs that have more than one hidden layer between the input and the output layers. They are widely utilized in performing cognitive functions (e.g., image recognition and speech recognition) normally performed by humans as well as predicting and controlling the behavior of physical systems (Morales et al. 2020). DNNs are widely integrated with GAs in the literature to reduce the number of experiments required for optimization. Initial experiments are used to train an MLP DNN that predicts the system behavior. The GAs then dispatch the predictor MLP instead of requiring additional experiments, thus greatly reducing the overall experiments required. Dasgupta et al. (2017) utilized this technique to model and optimize a polymer-enhanced ultrafiltration system (Dasgupta et al. 2017). Márquez-Nolasco et al. (2017) also used the technique to optimize and estimate the thermal energy of an absorber with graphite disks (Márquez-Nolasco et al. 2017).

Conde-Gutiérrez et al. (2018) used a similar technique to optimize an absorption heat transformer (Conde-Gutiérrez et al. 2018). In these works, the experiments performed for training the MLP are arbitrary and unguided, except by human experience. This may pose a problem for an arbitrary complex system whose behavior is not completely understood, because there is no guarantee that these experiments will train the MLP DNN so it can render a result sufficiently close to the global optimum. For such systems, a simple ‘grid’ of experiments may not be adequate or efficient because highly nonlinear areas of the parameter domain may need to be more densely probed relative to other areas.

1.1.5. Gated recurrent units

Gated recurrent units (GRUs) are a form of recurrent neural networks (RNNs) that can ‘forget’ irrelevant information in the hidden state and only keep information that affects the GRU’s prediction capability (Cho et al. 2014). GRUs have been utilized for wind speed forecasting, wind turbine condition monitoring, unmanned underwater vehicle state estimation, among other applications (Wei et al. 2021; Kong et al. 2020; Lin et al. 2021). GRUs are less computationally expensive than long-short term memory (LSTMs), and their performance is comparable (Rehmer and Kroll 2019).

Kong et al. (2020) developed a system for condition monitoring of wind turbines using spatiotemporal fusion of supervisory control and data acquisition (SCADA) data. The spatial features in the SCADA data are extracted by a convolutional neural network (CNN) and the temporal features in the sequence of the temporal features is extracted using a GRU network. Experiments were used to prove the effectiveness of the proposed method. The findings of this research justify utilizing GRUs for condition monitoring or wind condition prediction.

Lin et al. (2021) developed a GRU-based particle filter that improves the state estimation of unmanned underwater vehicles (UUVs). This allows UUVs to more efficiently perform tasks such as target tracking and collision avoidance. The developed framework relies on GRUs to use previous measurements and estimate the target states. Through the training process, the GRU is shown to learn the complex nonlinear dynamic behavior of the UUV and recognize the measurement noise. The effectiveness of the developed system is shown using simulations. This research supports using GRUs for condition prediction of complex nonlinear systems.

1.1.6. Reinforcement Learning

Reinforcement learning (RL) offers a way of programming agents through trial-and-error in a dynamic environment. For each state the agent encounters, RL training presents the agent with a set of possible actions. After the agent performs one of the actions, the RL training rewards it in proportion to how favorable the resulting system state was. Deep RL algorithms attracted increasing interest from machine learning researchers. It was successfully used for autonomous driving, learning to walk, human-level performance in 3d multiplayer games, path planning for unmanned aerial vehicles, among others (Kiran et al. 2021; Haarnoja et al. 2019; Jaderberg et al. 2019). Despite its success, efficient exploration of complex domains remains challenging for RL algorithms (Hong et al. 2018; Conti et al. 2018). This limitation can make the RL-generated policy unable to provide global optimum solutions especially for systems with infinite possible states and actions, such as the wind-building-plates system at hand. As discussed in Section 1.1.1, greater WIV acceleration response is coincident with the formation of vortices that eventually synchronize with the building motion (i.e., aeroelastic lock-in). Therefore, it is desirable to generate an agent able to obtain global minimum WIV accelerations using a minimum number of actions to minimize the possibility of aeroelastic lock-in.

1.2. Scope and Organization

The literature review presented in the previous section highlights that major and minor AMs of the building geometry constitute an effective means of reducing the building's WIV response. However, it was also pointed out repeatedly that changing the wind condition or surrounding environment can make a certain AM configuration increase rather than decrease the building's WIV response. Even for the same condition, different AM configurations may decrease one response component but increase another. In a similar manner, TMD/TLD systems can also increase the building's WIV response if the excitation force frequencies are located far away from the structural natural frequencies. TMD/TLD systems also lack the speed of actuation that fast-varying wind conditions may require.

Adaptive or kinetic building envelopes are currently utilized for shading and energy harvesting as well as for aesthetic purposes (Nagy et al. 2016; Miranda et al. 2020). They are generally composed of louvers, plates or origami modules that can actively alter their orientations or shapes (Nagy et al. 2016; Miranda et al. 2020). This study focuses on adaptive active AM systems whose orientations or shapes can be controlled based on online readings of the wind conditions. This approach aims to guarantee consistent reduction of the WIV response across different wind conditions. The control systems developed in this study have an adaptation mechanism, which guarantees that the WIV reduction performance remains constant even with long-term changes of the surrounding environment. The control systems also operate in a predictive manner, so they change the AM orientations in anticipation of the incoming wind conditions. This predictive operation enables the controllers to inhibit or greatly reduce even the

initial formations of the building's WIV response, which makes the system more suitable for fast-varying wind conditions (unlike TMD/TLD systems).

Chapter 2 of this dissertation focuses on the numerical and experimental tools that were used to understand and model the structural WIV response. The Commonwealth Advisory Aeronautical Research Council (CAARC) standard building model is used as a base case throughout this study. The chapter begins by introducing this building model along with the proposed four-corner-plates active AM system. This system was given special focus in this study because of its simplicity and reduced number of variables.

Chapter 3 of this dissertation presents a steady-state optimum controller whose input is the average readings of the wind speed and directions, and whose outputs are the corner plate orientations that minimize the building's WIV response. The chapter then discusses the controller training procedures which make use of the models and tools introduced in Chapter 2. Most notable among those, is the developed iterative optimum training (IOT) procedure which is proven to require a reduced number of experiments to achieve the training. The chapter finally introduces the WIV reduction results obtained by using the trained controllers.

Chapter 4 begins by highlighting the practical drawbacks introduced by using the wind speed and direction to define the wind condition. An alternate approach relying on multiple sensor systems and a GRU machine learning model is introduced to address these drawbacks and provide full unique instantaneous definition of the wind-building-plate system. The chapter also introduces a genetic reinforcement learning (GRL) technique that was developed to streamline

and automate the controller training process. The technique is a modification of the typical RL process which uses GAs and multiple trainable policy functions. The technique allows for training the controller in a dynamic environment while increasing the likelihood of obtaining global optimum solutions by using a pseudo-random search technique. This approach is intended to improve the exploration ability of the typical RL algorithm.

Chapter 2 - Numerical and Experimental Methods

This chapter focuses on the modeling methods that were utilized to estimate the structural WIV response in different conditions, which paves the way for WIV controller development in the next chapters. The chronological development of the research work necessitated that proof-of-concept numerical simulation be used in the early stages to avoid building poorly performing experimental models. Therefore, the simulation model illustrated in Section 2.2 was only used to perform steady-state optimum control (Chapter 3), while the aeroelastic robotic building models discussed in Section 2.3 and 2.4 were used to perform both steady-state and genetic reinforcement learning (GRL) control (Chapter 4).

2.1 Standard CAARC Building

The CAARC standard building is used as a base case in this work because it is thoroughly represented in the literature. Figure 2.1 illustrates the full-scale CAARC building along with all key parameters including dimensions which are $180 \times 45 \times 30$ m (Height $H \times$ Width $B \times$ Depth D). The average mass density of the building is 160 kg/m^3 (Tang and Kwok 2004; Hou and Sarkar 2020). A 1:400 scale of these dimensions is used in the FSI simulation model (Section 2.2) and the small-scale experimental model (Section 2.3), while a 1:75 model is used in the large-scale wind tunnel experiments (Section 2.4). As illustrated in Figure 2.1, the building exhibits two sway (bending) and 1 torsional (twisting) vibration degrees of freedom (DOFs) that are of primary interest for this study. The figure also shows a top view of the building with four plates attached at the corners having orientation angles ($\alpha_i = \alpha_1 \dots \alpha_4$). The plate width is approximately 10% of the building width, which is an acceptable ratio for real-life deployment. Since flow separation is strong at the corners, varying the plate orientations at these locations is expected to have a significant effect on the WIV response (Abdelaziz and Hobeck

2019; Abdelaziz et al. 2021). The range of 0 to 180 degrees for the plate rotation α_i is sufficient because the corner plates are chosen to be aerodynamically symmetric. If nonsymmetric plates are used, then a 360 degree range should be considered. The figure also shows the convention for measuring the wind speed U and direction AOA . These two variables define the wind condition vector $w = [U, AOA]$ or $\bar{w} = [\bar{U}, AOA]$ if the average wind speed $\bar{U}$ is used instead. Even though the building's WIV response depends on both the wind condition and the plate angles, only the latter can be manipulated by the controller. This separates the system parameters into uncontrollable ($\bar{w}$) and controllable parameters (α_i). The control systems discussed in Chapter 3 and Chapter 4 change the plate orientation angles α_i to minimize the building's WIV response assuming no control over the changes in the uncontrollable parameters $\bar{w}$, which constitutes an active AM implementation.

2.2. Fluid-Solid-Interaction Simulation

Evaluating the effect of active AMs is challenging for computational simulations because they normally require repetitive reconstruction of the meshing of the simulation domain, which is computationally expensive (Li et al. 2019). This section discusses a simplified 2D FSI simulation model, which enabled the AM geometry to change without reconstruction of computational mesh to allow for more efficient optimization (Abdelaziz and Hobeck 2019; Abdelaziz et al. 2021).

The model is validated using experimental data published in the literature for the CAARC standard building model (Tang and Kwok 2004).

2.2.1. Fluid Model

For the wind speeds considered (less than 10 m/s), air can be assumed to be incompressible. The optimization discussed in Section 3.2 involves a large number of simulations (~490), so the Reynolds averaged Navier-Stokes (RANS) equations were utilized

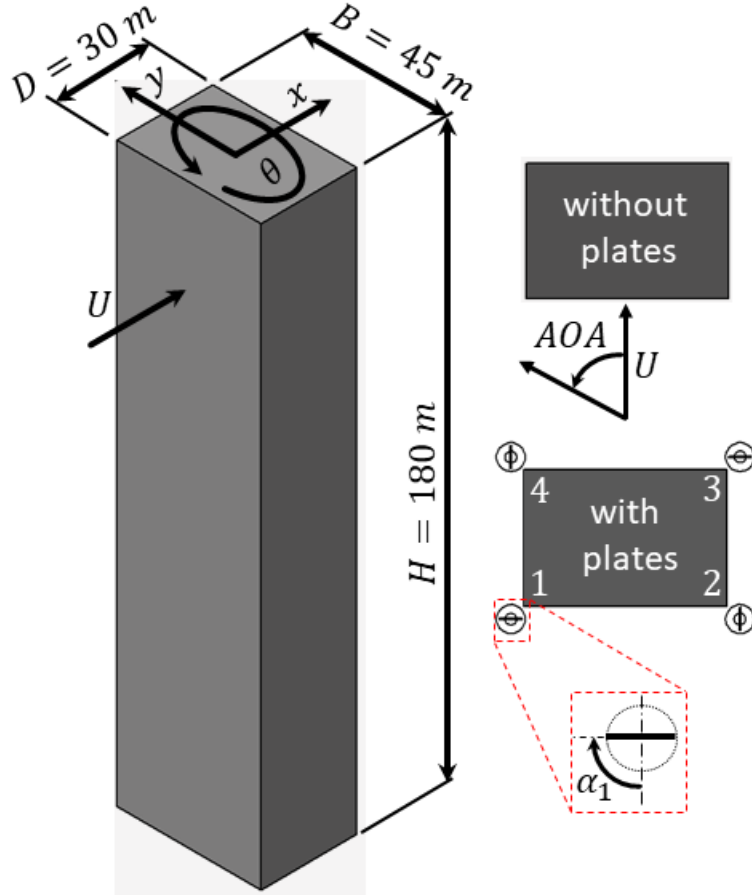


Figure 2.1. Layout of the CAARC building model showing the full-scale floor dimensions, DOFs of interest, the convention for measuring the wind AOA and the rotation angles α_{1-4} of the four plates attached to the corners

because of their low computational cost relative to other methods such as Large Eddy Simulations (LES) (Antonini et al. 2019). Antonini et al. showed that by using wind direction averaging, the RANS $k-\omega$ Shear Stress Transport (SST) turbulence model provides accurate predictions for wind speed in the wake of wind turbines (Antonini et al. 2019; Holzmann 2016). This motivated the adoption of the RANS $k-\omega$ SST model in the current study, which performs a considerable amount of building response time averaging while employing separate simulations for different conditions (see Section 3.2). The equations are discretized using appropriate schemes for each term on a 2D finite volume mesh created using SALOME® as shown in Figure

2.2 (Ribes and Caremoli 2007). The OpenFOAM® (Weller et al. 1998) implementation of the hybrid PIMPLE (PISO+SIMPLE) algorithm is used for time integration as well as pressure-velocity coupling with a dynamic time step that changes to satisfy the Courant-Friedrichs-Lewy (CFL) condition. The condition generally guarantees the time step, the flow velocity and the mesh size are all appropriately proportioned (Holzmann 2016; Courant et al. 1928). A maximum CFL number of 3 was used, because some of the simulation time steps failed to resolve at higher values, which could negatively affect the validity of the solutions. In the CFD mesh, each plate is centered inside of a circular mesh boundary which has an interpolation interface with the surrounding mesh (see Figure 2.2). When the circular mesh boundary rotates while the simulation is running, the mesh nodes become incoincident at the boundary, so the fluid field variables (speed and pressure) are interpolated between the neighboring nodes (Jasak 2009). This allows the plates and the circular mesh to rotate freely without re-meshing. ParaView software was used to visualize and analyze the results (Ayachit 2015).

2.2.2. Rigid Body Model

From a design perspective, the in-plane deformations of a floor are negligible when compared with the bending and twisting deformations along the building's height. As a cantilevered structure, the largest vibration amplitudes of the lowest frequency modes generally occur at the free end. Therefore, the simulation in this study considers the top building floor as a 2D, 3-DOF rigid body (see Figure 2.1). The along- and cross-wind movements of the floor correspond to the bending of the building in two directions, while its rotation corresponds to the torsion or twisting of the building. This simplified model cannot account for excitation of the building's higher frequency modes, which would require more than one floor to be simulated and

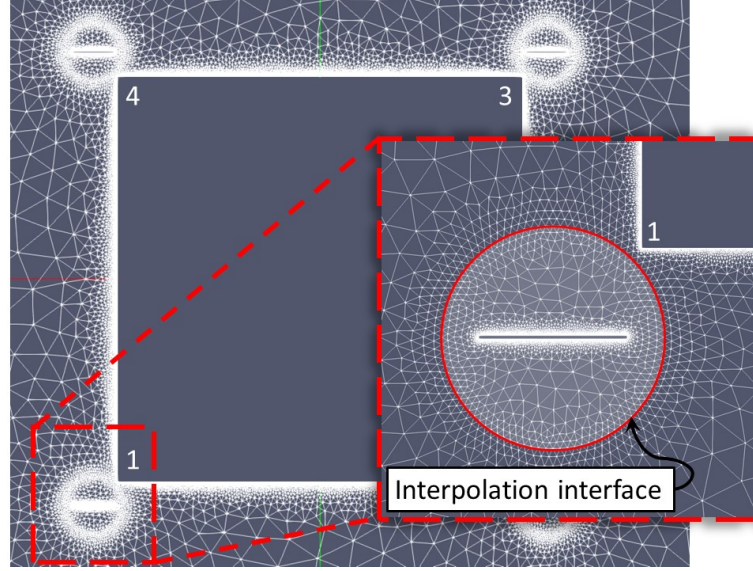


Figure 2.2. The 2D computational mesh of the four-plate CAARC building case showing the computational mesh region of the corner plates and the rotating interpolation interface for plate 1

coupled together. It will later be shown in Sections 2.2.4 that the dominant wind force frequency is either found to be or chosen to be around the lowest natural bending frequency. This minimizes the effects of higher vibration modes, thus justifying the use of the proposed simplified model. After the pressure and the velocity fields are obtained from the fluid solution, the building height and the perimeter length are used to estimate the wind forces. The forces acting on the building and the plates are summed at the geometric center of the floor. The dynamic response of the top floor as a rigid body with the 3 DOFs explained above is given by the following equation (Inman 2013):

$$M\ddot{X} + C\dot{X} + KX = HF(t)$$

$$X = [x \quad y \quad \theta]^T \quad (2.1)$$

where x , y and θ are the along-wind bending, cross-wind bending, and rotational displacements of the floor, respectively. M is the mass matrix, C is the damping matrix, K is the stiffness matrix, $\dot{X}$ is the velocity, $\ddot{X}$ is the acceleration, H is the building height and $F(t)$ is a vector of

forces per unit height of the building extracted from the CFD simulation at each time step.

Equation 2.1 is solved using the Householder algorithm for symmetric matrices (Householder 1958) to provide the instantaneous values of the accelerations $\ddot{X}$ which are integrated using the simulation time step to obtain $\dot{X}$ and X (Abdelaziz and Hobeck 2019). The coupling between the three DOFs is assumed to be negligible which results in diagonal K , M and C matrices (Momtaz et al. 2017). For the cases being considered, the natural frequency ω for each DOF is known a priori from the literature. System mass and stiffness values were determined by assuming or estimating the stiffness, k , for each DOF, then calculating the mass m with the relationship: $\omega = \sqrt{(k/m)}$. The damping matrix C is assumed to be a linear combination of the M and K matrices (proportional damping) (Inman 2013):

$$C = \alpha M + \beta K \quad (2.2)$$

For a DOF i , α and β are determined by solving the system of equations:

$$\zeta_i = \frac{\alpha}{2\omega_i} + \frac{\beta\omega_i}{2} \quad (2.3)$$

where ω_i is the undamped natural frequency and ζ_i is the damping ratio, which was used to a typical value for buildings (1%) (Love et al. 2018).

2.2.3. Fluid-Structure-Interaction

After the instantaneous deflections, X , are obtained from the moving rigid body model, the floor is moved in the CFD mesh accordingly. In reference to Figure 2.3, a rectangular region

R of the mesh surrounding the building and the plates is assumed to have a constant shape so each point of it is moved according to X . The region R typically extends for about 5 times the building width so that the smallest cells in the mesh (adjacent to the solid body) will not endure any volume changes. The outermost mesh surrounding the region R is the only part of the mesh that is allowed to deform. Its cells expand and contract up to the fixed boundary to accommodate translation and rotation of the building. The motion of each node in this deforming mesh region is determined using linear interpolation between nodes on the boundary of R (full motion) and nodes on the outermost boundary of the computational domain (zero motion). Using linear interpolation allows the mesh to deform gradually to reduce chances of mesh distortion. All mesh point motions are performed by utilizing the OpenFOAM dynamic mesh library (Jasak 2009) which adjusts the mass flow through the cell faces automatically, effectively imposing position and velocity boundary conditions on the fluid domain for the next time step, as per the widely-used partitioned FSI scheme (Banks et al. 2018).

2.2.4. Validation

Validation of the proposed FSI model was performed by comparing its displacement power spectral density (PSD) to experimental data from the literature. A PSD can show the structural response, natural frequencies, and dominant vortex shedding response – all of which were used for model validation. Tang *et al.* (2004) studied the interference effect between adjacent buildings using wind tunnel experiments (Tang and Kwok 2004). Their study includes results for a standard CAARC building as shown in Figure 2.1 (without the plates) with no other buildings surrounding it, which was used herein to validate the previously described FSI simulations. Tang also presented vertical wind velocity and turbulence intensity profiles with maximum values of 8.3 m/s and 9%, respectively, measured at the top of the building.

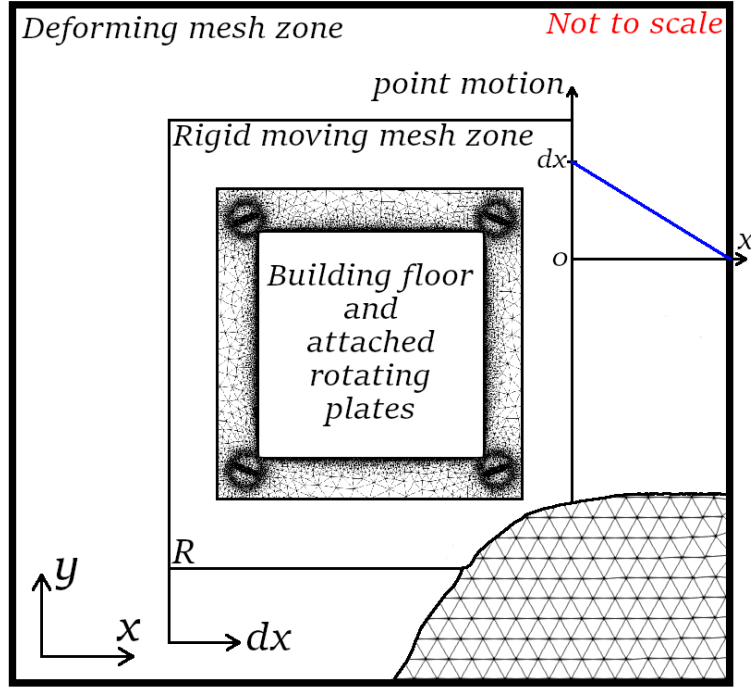


Figure 2.3. Motion of mesh points relative to the displacement of the moving region (R)

This velocity corresponds to approximately 54 m/s for the full-scale building. The natural frequencies for the along-wind bending, cross-wind bending and twisting motions were extracted from the normalized PSD charts provided by Tang et al. (Tang and Kwok 2004). The wind conditions at the top floor were used as inputs to the CFD simulation. Because the proposed model is a 2D representation of a 3D building, it is not able to account for changes in flow conditions along the building height. It is worth mentioning that for this initial model, it is assumed that considering 3D flow effects is not required to demonstrate the active AM concept and its controller development. The experimental models discussed in Section 2.3 and 2.4 were later used to more accurately estimate the full-scale building WIV response. The final parameters considered by the simulation are summarized in Table 2.1.

Tang et al. utilized a normalization in their PSD charts that is used in the current study to enable a fair comparison of results (Tang and Kwok 2004). This normalization is given by the following equation:

$$S_{y_n}(n) = \frac{nS_y(n)}{\left[\frac{1}{2k_{y_r}}\rho U^2 B H^2\right]^2} \quad (2.4)$$

where $S_{y_n}(n)$ is the normalized cross-wind deflection PSD, n is the first mode cross-wind natural frequency of the building, $S_y(n)$ is the cross-wind deflection PSD, ρ is the density of air, k_{y_r} is the rotational stiffness, which equals the applied moment around the base of the building that causes a unit displacement at the top. This stiffness was calculated by multiplying the assumed linear stiffness by the building height. The normalized PSD (Equation 2.4) obtained from the wind tunnel experiment performed by Tang et al. and the simulation performed in the current study are plotted versus the reduced frequency ($n_r = nB/U$) in Figure 2.4 (Tang and Kwok 2004). The peak wind force frequency obtained from the experiment is only 4% higher than the simulation, which constitutes an acceptable amount of error for this study. The relative differences between the amplitude levels at different frequencies are also in good agreement. However, the overall amplitude resulting from the simulation developed in this work is higher than the wind-tunnel experiment. This discrepancy exists because the simulation considers only the wind speed at the top of the building (maximum value), and projects this effect along the total height of the building. This results in higher oscillation amplitudes relative to the experiment, which measures the response of a wind velocity profile that shows a reduction in wind speed until it reaches 0 m/s at the ground level (power law). Another potential source of error is the unrealistic turbulence modeling in two-dimensional domains, which eliminates all chaotic fluid motions in the depth dimension. Minor differences in the frequency values may be the result of the digitization process of the plots provided by Tang *et al.* (Tang and Kwok 2004).

For the purpose of optimization, which relies on comparing different plate configurations to each other, the model results are still usable despite the presence of minor numeric differences.

Table 2.1. Parameters of the FSI simulation model validation case involving the CAARC model without the plates

Parameter	Value	Unit
Along-wind frequency ω_x	12.3	Hz
Cross-wind frequency ω_y	12.8	Hz
Rotational frequency ω_r	18.1	Hz
Damping ratios $\zeta_x, \zeta_y, \zeta_r$	1%	-
Wind speed U	8.3	m/s
Turbulence intensity I_u	9%	-
Angle of attack AOA	0°	-
Reynolds number RE	60,000	-
Depth D	0.075	m
Width B	0.1125	m
Height H	0.375	m
CFD domain width	3	m
CFD domain height	3	m
Plate width	6×10^{-3}	m
Plate thickness	2.5×10^{-4}	m
Cell count	241,143	

2.2.5. Results and Discussion

This section uses the developed simulation model to investigate the relationship between wind speed, plate angles, and building vibration to support the feasibility of the proposed active AM system. Data taken in the United States shows that the average wind speed at 30 m above sea level is in the range of 5-9 m/s, which corresponds to 0.8-1.4 m/s for the 1:400 scale CAARC model illustrated in Section 2.3 (Pryor et al. 2009). Figure 2.5 compares the simulated power spectral density (PSD) of cross-wind forces on the CAARC building floor at 0.8 and at 1.4 m/s. It can be observed that the peak frequency was shifted from 0.2 to 0.6 Hz because of the wind speed change without any changes in the floor geometry or the plate angles.

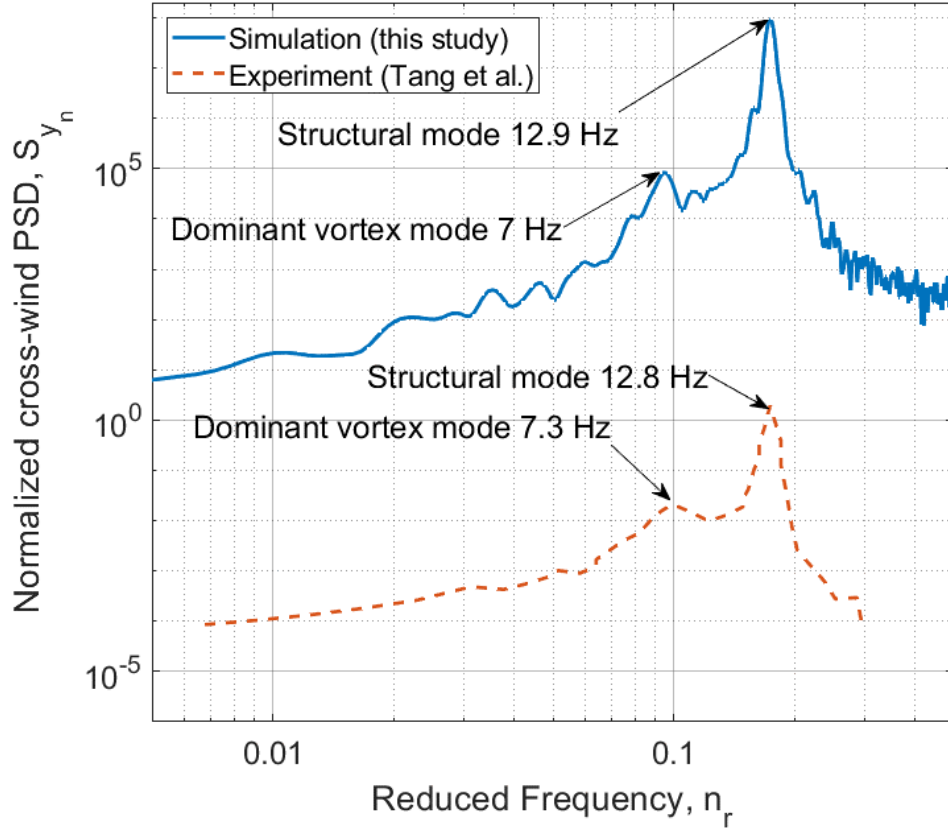


Figure 2.4. Comparison between the normalized cross-wind displacement PSDs generated by the developed two-dimensional simulation (this study) and by experimental data from (Tang and Kwok 2004)

The Strouhal Number, S_t , is used to define the well-known linear relationship between flow velocity, U_H , and vortex shedding frequency, f_s as (Okajima 1982):

$$U_H = \frac{f_s L}{S_t} \quad (2.5)$$

where L is the characteristic length of a rectangular cylinder. The proposed active AM system allows this relationship to be controlled rather than being a fixed function with no controllable terms. Figure 2.5 also shows the PSD of cross-wind forces at a wind speed of 0.8 m/s when all plate angles are set to 0° (i.e. neutral) and when they are set to 45° . The peak frequencies at 0.19 and 7 Hz were shifted to 0.33 and 9.5 Hz, respectively. This effect was obtained without changing the wind speed. Figure 2.6 compares the streamlines of the vortex streets that are

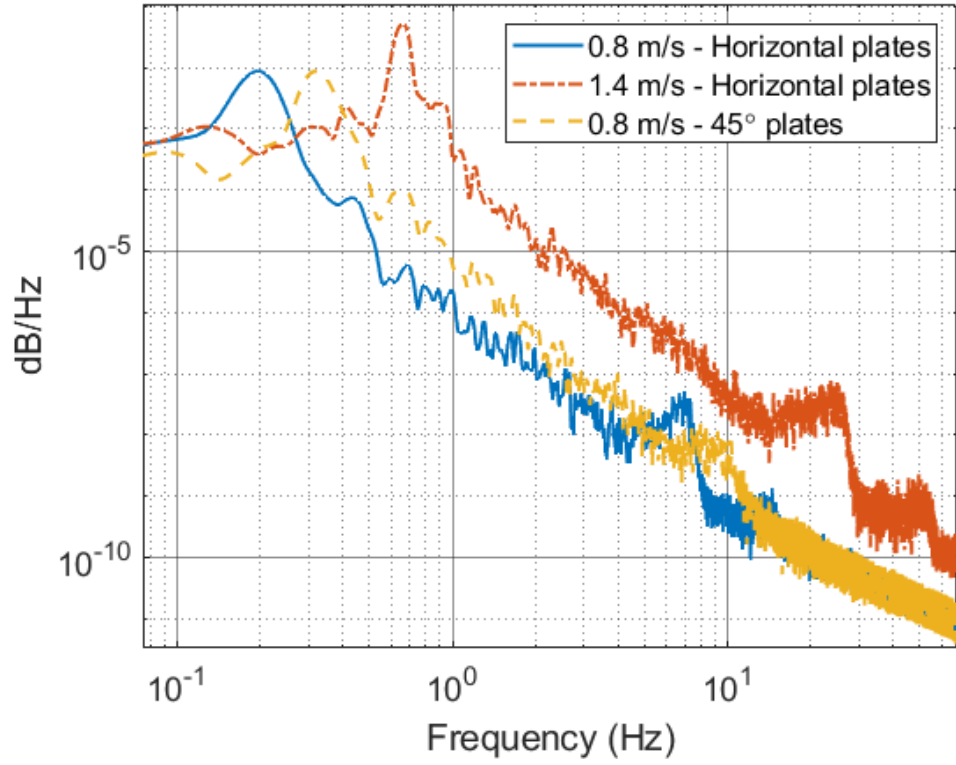


Figure 2.5. PSD of cross-wind forces obtained from the FSI simulation of the CAARC scaled model

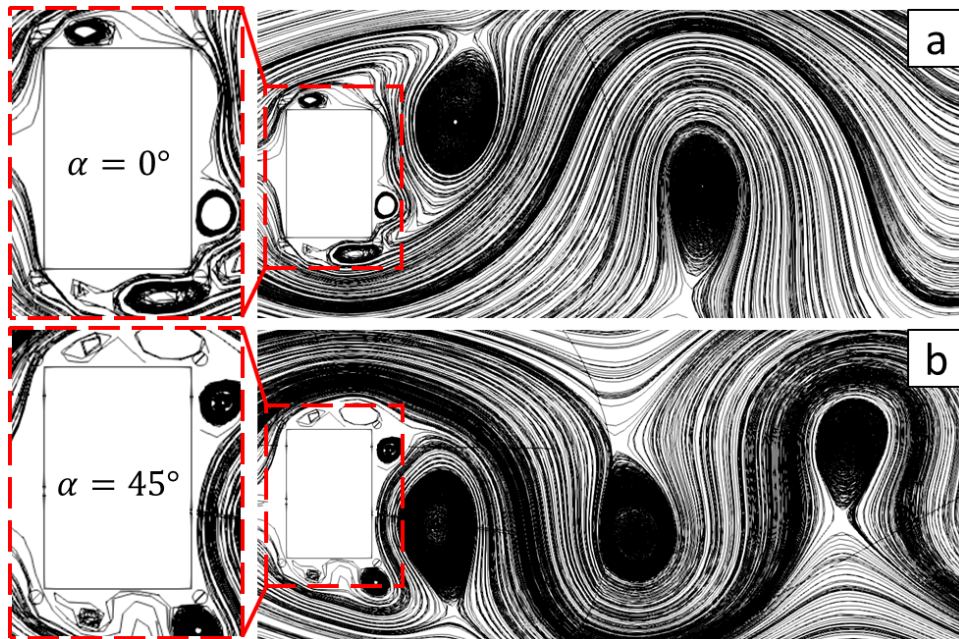


Figure 2.6. Vortex street formed behind the CAARC building at 0.8 m/s wind speed for (a) horizontal plates and (b) plates tilted 45°

formed behind the building for plate orientations of 0° and 45° with an inlet velocity of 0.8 m/s (the same conditions used in Figure 2.5). It is clear that the distance between vortex centers decreases when the plates are positioned at 45° . This decrease indicates an increase in vortex shedding frequency f_s as confirmed by the resulting force PSD in Figure 2.5. The ratio of the average distance between vortex centers (~ 1.7) is equal to the shift in the maximum peak frequency ($0.33/0.19 \cong 1.7$). The results of these preliminary simulations indicate that plate orientations strongly influence the vortex shedding behavior, which is a significant contributor to the WIV response (see Section 1.1.1). This finding is a strong justification for developing the controllers in Chapter 3 and Chapter 4, both of which rely on changing the plate orientations to obtain minimized structural WIV response.

2.3. Small-scale Wind Tunnel Experiments

The FSI simulation model discussed in the previous section served as a proof-of-concept and was also used to develop initial control techniques as illustrated in Chapter 3. However, the 2D simplification included in the FSI simulation model introduced considerable approximation into the model, especially regarding turbulence modeling and 3D flow effects. Additionally, despite the simplification efforts, the simulation model was still computationally expensive compared to wind tunnel experimental models. In the literature, cyber-physical experimental wind tunnel models have recently been used to design aerodynamic modifications for aviation and tall building applications at a reduced experimental cost (Fagley et al. 2017; Whiteman et al. 2018; Fernández-Cabán et al. 2020). In this section, a cyber-physical scaled building model is designed and manufactured to enable the development of online controllers (Abdelaziz and Hobeck 2022). The model has four actively controlled corner plates as illustrated in Section 2.1 (Abdelaziz et al. 2021). The plates function as active AM surfaces, which the controller can

independently position in response to time-averaged measurements of the wind condition $\bar{w}$ to minimize the building's WIV response.

Starting from the dimensional scale of 1:400, Table 2.2 lists the scaling parameters of other quantities of interest, which were required to ensure transferability to the full-scale CAARC building (Wind Tunnel Studies of Buildings and Structures 1999). As will be shown in Section 2.3.4, the damping ratios were approximately the same as the full-scale building, which yields the unity scale for the Scruton number and eventually the accelerations, as shown in the table (Hou and Sarkar 2020).

The standard CAARC building discussed in Section 2.1 has an average mass density of 160 kg/m^3 , which is uniformly distributed along its height (Tang and Kwok 2004). Also, Isyumov (1982) reported that four lumped masses are enough to study most tall buildings (Wind Tunnel Studies of Buildings and Structures 1999). Figure 2.7 illustrates details of the 1:400 scale aeroelastic CAARC model. The height of the scaled model is 451.7 mm. Four aluminum plates representing the floors were equally spaced along the height ($451.7/5 = 90.3 \text{ mm}$). In order to evenly distribute wind forces, the amount of mass at each floor was proportional to the envelope height assigned to it. The top floor was assigned double the envelope height of other floors to make room in the mass budget for the active AM plates and their control mechanisms. This resulted in a model with four adjustable lumped masses representing five floors two of which were combined at the top floor. The columns were cut from 6061 aluminum threaded rods and were attached to the floors using self-locking nuts. The envelope of the model was composed of

balsa wood panels laser cut from 1/32" (0.8 mm) stock and attached to the floors. The weight of the final model excluding the base was 0.62 kg, which was 2% over the target weight of 0.61 kg.

Table 2.2. Dynamic similarity parameters for the scaled aeroelastic model (subscripts **m** = model and **f** = full-scale building).

Quantity	Parameter	Scale value
Length	$\lambda_L = l_m/l_f$	1/400
Frequency	$\lambda_f = f_m/f_f$	33
Time	$\lambda_t = t_m/t_f$	1/33
Displacement	$\lambda_L = l_m/l_f$	1/400
Mass	$\lambda_m = m_m/m_f$	1/400 ³
Velocity	$\lambda_v = \lambda_L \lambda_f$	1/12.1
Damping ratio	$\zeta_m = \zeta_f$	1
Acceleration	$a_m = a_f$	1

2.3.1. GA-assisted aeroelastic model design

For the standard CAARC building, the first order sway mode shapes are linear along the height of the building, while the torsional mode shape is uniform. The torsional/sway first order natural frequency ratio ranges from 1.2-1.5 in the literature (Tang and Kwok 2004; Frison et al. 2019).

Column diameters at each floor were used to manipulate the mode shapes and frequency ratios of the model (Frison et al. 2019). A GA optimization procedure was formulated and used to determine the column cross-sections that (1) minimize the deviation from linear mode shapes and (2) obtain appropriate torsional/sway frequency ratios. The objective of this optimization was defined as,

$$\min. f(d_i) = \sum_{i=1}^4 \Delta_i + \left| \frac{\omega_\theta}{\omega_x} - 1.2 \right| \quad (2.6)$$

where d_i are the column diameters for floors $i=1$ to 4, which were chosen from a discrete list of readily available off-the-shelf diameters, Δ_i is the deviation from the linear mode shape at floor i , ω_θ and ω_x are the torsional and along-wind sway natural frequencies respectively.

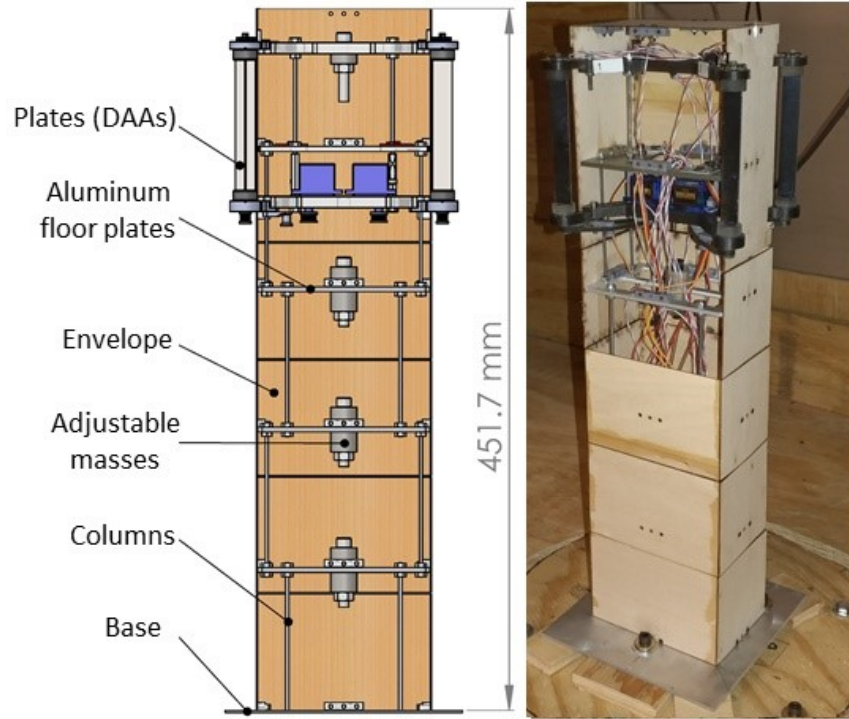


Figure 2.7. A detailed drawing of the scaled aeroelastic building model showing the main components of the model in addition to a photo of the assembled model with the skins removed at the top floors

A finite element (FE) model of the scaled building was used to calculate Equation 2.6. The model consisted of $\sim 300,000$ nodes for $\sim 160,000$ 3D solid elements and accounted for every component of the aeroelastic model assembly including fasteners, hardware, and the envelope boards. The mass density of each component was estimated during the initial design phase and updated with measured values after components were acquired prior to assembly. The FE model was created using the Solidworks® frequency analysis module. Figure 2.8 illustrates the first order sway mode shapes and natural frequencies (ω_x and ω_y) obtained using the GA optimization. The mode shapes show very little deviation from linearity. These deviations are primarily due to limiting the column diameter selection to discrete sizes. The torsional mode shape occurs at 8.0 Hz, so the torsional/sway ratio is 1.17 and 1.16 for the along- and cross-wind

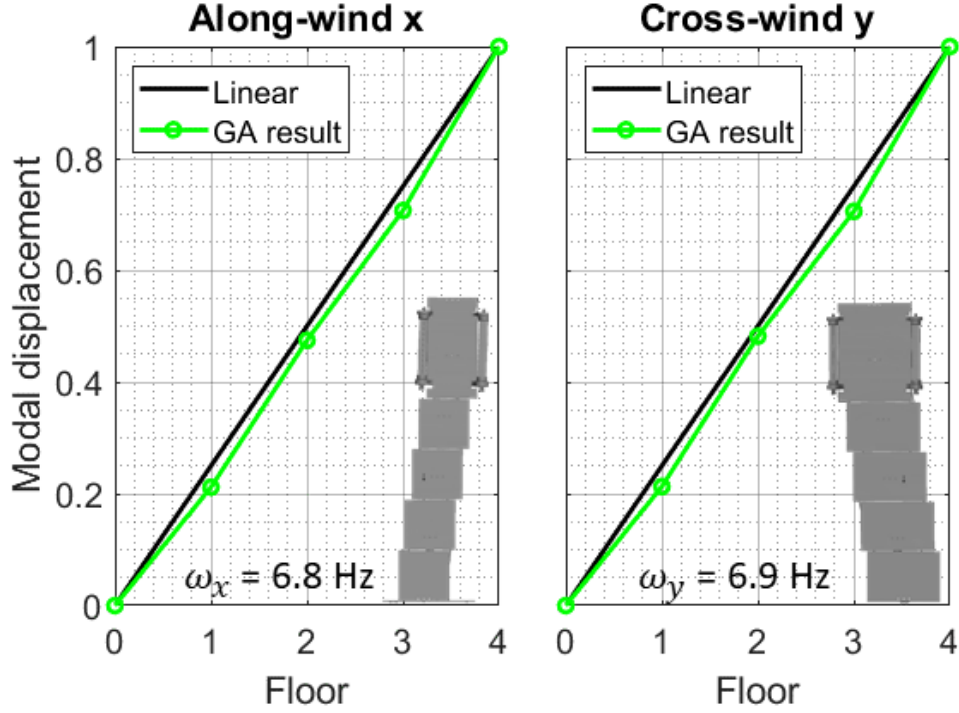


Figure 2.8. First order sway mode shapes for the scaled aeroelastic model obtained using GA optimization.

sway directions. This ratio is only 2.5% lower than the range of ratios found in the literature (Hou and Sarkar 2020).

2.3.2. Active AM plates

The four corner plates were cut from 0.8×12.8 mm steel to a length of 90 mm each. The width of the plates was chosen to be approximately 10% of the building width B to ensure practical plate dimensions at the full scale. The plates extended for approximately 30% of the total building height to maintain their rigidity and minimize the investment cost. They were placed towards the top of the model to maximize their effect because this is where wind velocities, wind forces, and building deflections are typically the largest. Figure 2.9 illustrates the plate rotation mechanism. Each plate attaches to a rotational encoder at the top, which measures actual plate position. The plate motors have a range of 0-180° and weigh ~9 grams each.

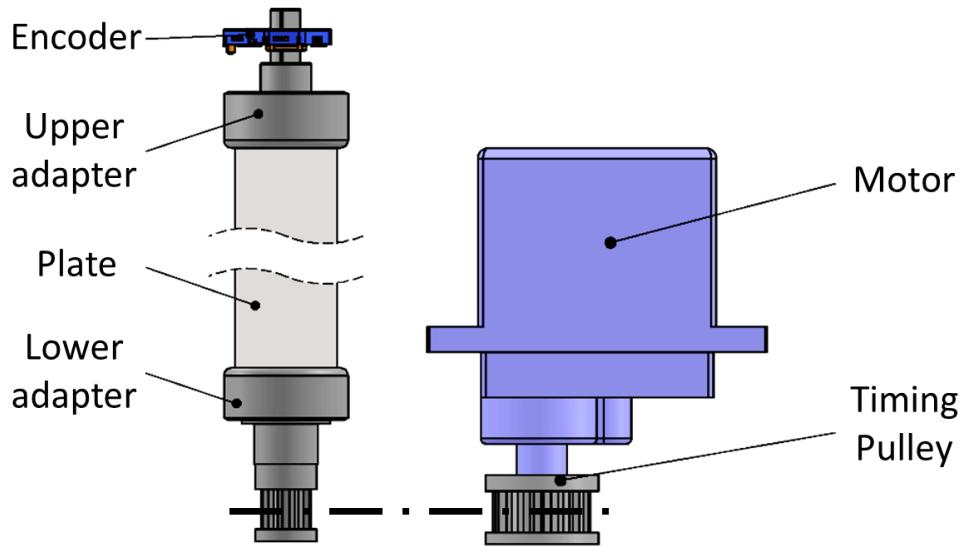


Figure 2.9. Components of the plate rotation mechanism used to implement active AM.

They can perform full travel in ~ 0.2 seconds with a stall torque of ~ 0.1 N.m and are compatible with Arduino®.

2.3.3. Cyber-physical instrumentation

Figure 2.10 illustrates the data acquisition and plate control instrumentation that were used to complete the full cyber-physical system. Python scripts utilized the nidaqmx software library to control the data acquisition unit and enable serial communication with an Arduino which was used to control the motors. The tests were performed in a mobile wind tunnel built by the Kansas State University Wind Power Team. The tunnel has a $1.1 \text{ m} \times 1.1 \text{ m}$ test section powered by 4 industrial fans each measuring $0.5 \text{ m} \times 0.5 \text{ m}$. The tunnel generates wind speeds ranging from 0-14 m/s. The speed control on the tunnel is open-loop and human operated with discrete integer ‘level’ settings. Average free-stream windspeed $\bar{U}$ was determined using 3-minute averaged readings with a Testo 405i hot-wire anemometer placed ~ 200 mm upstream of the model. While all wind speed readings are taken at a single point, the 3-minute averaging

generates distinct average wind speed values for each tunnel speed setting, which is necessary for controller training.

2.3.4. Results and Discussion

Experiments were performed to determine the first mode natural frequencies and damping ratios for the scaled model. The model was given an initial displacement smaller than 3 mm and left to vibrate freely while the motion was measured with a Polytec® OFV-5000 laser doppler vibrometer (LDV) unit as well as the onboard accelerometers. The power spectral density (PSD) charts were plotted for each DOF to obtain natural frequencies. Damping ratios were estimated using the logarithmic decrement method. Table 2.3 lists the characterization results which show minimal deviation from the design natural frequency ratios as explained in Section 2.3.1. These small deviations were primarily due to minor manufacturing and assembly inaccuracies. The obtained damping ratios for the linear modes are also in line with the full-scale model, so no special modifications were made to change the damping ratios.

2.4. Ongoing development of a large-scale aeroelastic model

The small-scale aeroelastic building model illustrated in the previous section provided for more accurate and efficient means of estimating the WIV response when compared with FSI simulations. However, the 1:400 dimensional scale of the model limited the number of control surfaces that can be included due to weight and space limitations.

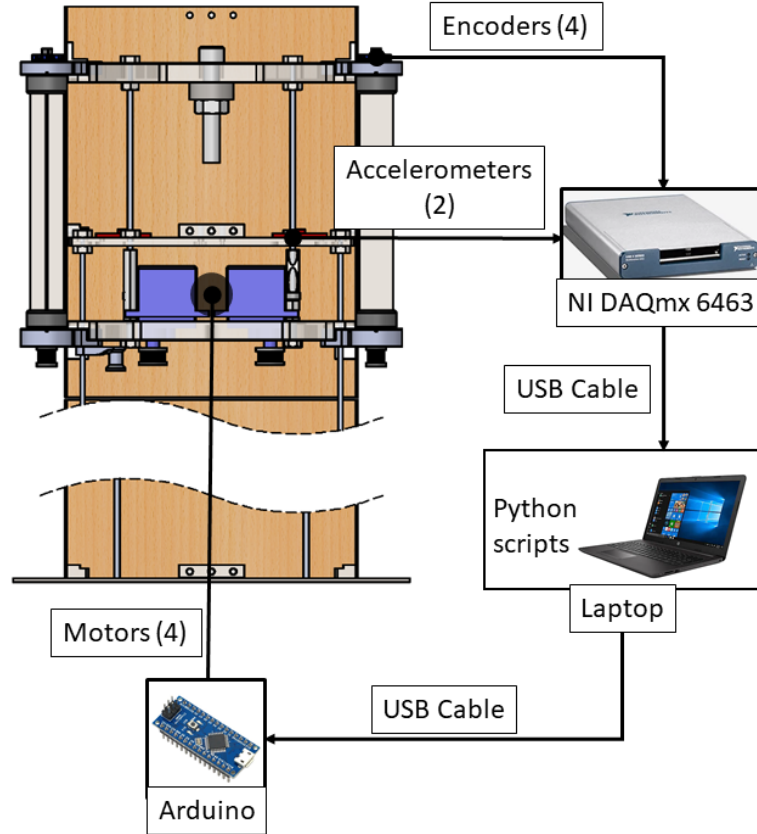


Figure 2.10. Cyber-physical instrumentation of the scaled CAARC model consisting of data collection and plate control systems.

Table 2.3 List of natural frequencies and damping ratios of the scaled model.

Parameter	Design	Actual	Difference
Natural frequency ratios			
ω_{θ}/ω_x	8.0 Hz / 6.8 Hz = 1.17	7.4 Hz / 6.5 Hz = 1.13	3.4%
ω_{θ}/ω_y	8.0 Hz / 6.9 Hz = 1.16	7.4 Hz / 6.7 Hz = 1.10	5.2%
Damping ratios			
ζ_x	1%	1.1%	10%
ζ_y	1%	1%	0.3%
ζ_{θ}	-	0.8%	-

As discussed in Section 3.3.4.2, additional control surfaces increase the control authority which can improve the effectiveness of the WIV controllers discussed in Chapter 3 and Chapter 4. The model also had only four independent floors while the full-scale CAARC building has about 60 floors. The CAARC building is not considered a super tall building, so the reduced number of

four floors does not considerably affect the validity of the results (Wind Tunnel Studies of Buildings and Structures 1999). However, additional floors gives the model the ability to more accurately represent higher modes of oscillation.

The National Science Foundation (NSF) research project that funded most of this research included large-scale wind tunnel experiments performed at the Natural Hazards Engineering Research Infrastructure (NHERI) Wall-of-Wind (WoW) facility in Florida International University (FIU). The test section of the WoW is 14 ft. high $\times$ 20 ft. wide (4.3 m $\times$ 6.1 m), which enables testing full-scale buildings or large-scale tall buildings. The WoW is powered by a 12-fan system capable of repeatable testing in up to 157 mph wind atmospheric boundary layer wind.

2.4.1. Model Design

A large-scale CAARC building model was designed and manufactured to perform experiments at the WoW facility and further demonstrate the effectiveness of the developed WIV controllers (see Chapter 3 and Chapter 4). Figure 2.11 presents an illustrated picture of the large-scale CAARC building model. The dimensional scale of the model is 1:73.3, which yields the scaled model dimensions of $2.5 \times 0.6 \times 0.4$ m (Height $H \times$ Width $B \times$ Depth D) (see Figure 2.1). The increase in available construction space allowed this model to have 9 floors, as shown in the figure. Table 2.4 lists the scaling parameters of the quantities of interest to maintain transferability to the full-scale CAARC building (Wind Tunnel Studies of Buildings and Structures 1999).

The dynamics of the model were designed using a lumped-mass programming code written in Python. Similar to the small-scale model design, the main objective was to determine the column diameters to obtain acceptable mode shapes and natural frequency ratios. Starting from identical column diameters, various trial and error iterations were performed to obtain the final column diameters. Compared to using Solidworks® frequency simulations for the small-scale model, the lumped-mass code offered great time savings in the design phase. The final resulting design was still verified using Solidworks® frequency simulations. The final obtained along wind structural natural frequency was 2 Hz compared to 0.2 Hz for the full-scale CAARC building, which yields the frequency ratio of 10 listed in Table 2.4 (Tang and Kwok 2004).

2.4.2. Active AM designs

The vertical corner plate configuration discussed in Section 2.3 was also employed for testing with the large-scale model. However, the increased size of the model allowed for testing the corner plates along the height of the whole building instead of only the top 30% for the small-scale model. Each floor has four corner plates that can be controlled independent from the other floors. The model is also designed to accommodate another active AM design that uses a plethora of horizontally rotating fin-louver assemblies (Fangwei 2020). Figure 2.12 illustrates a single floor of the large-scale CAARC building model showing the floor dimensions. The figure also shows the horizontally rotating fin-louver active AM design theorized and investigated by the NSF project collaborators at Iowa State University (Fangwei 2020). As shown in the figure, each side of the model's envelope is composed of two thin plates with a 6 mm separation distance between them. A pattern of circular ducts with triangular stators is cut into the front plate. At the center of each circular duct, a rectangular fin protrudes from the envelope for $\sim 4\%$ the building width B . A louver sandwiched between the front and the back plates rotates with the

Table 2.4. Dynamic similarity parameters for the large-scale aeroelastic model (subscripts m = model and f = full-scale building).

Quantity	Parameter	Scale value
Length	$\lambda_L = l_m/l_f$	1/73.3
Frequency	$\lambda_f = f_m/f_f$	10
Time	$\lambda_t = t_m/t_f$	1/10
Displacement	$\lambda_L = l_m/l_f$	1/73.3
Mass	$\lambda_m = m_m/m_f$	1/73.3 ³
Velocity	$\lambda_v = \lambda_L \lambda_f$	1/7.3
Damping ratio	$\zeta_m = \zeta_f$	1
Acceleration	$a_m = a_f$	1

external fin as a single fin-louver assembly to a position α measured as shown in the figure. The axis of rotation is at the center of each duct. The fin is intended to affect the air flow in front of the duct, while the louver is intended to affect the porosity of the building envelope. Both effects were shown to reduce the WIV response in other research (Hu et al. 2017; Fangwei 2020). In that alternate design, the controller changes the angle of rotation α (see Figure 2.12) of the fin-louver assembly instead of changing the vertical plate rotation angle to minimize the building's WIV response. The control techniques developed in Chapter 3 and Chapter 4 are generic in nature and can be used for either one of the active AM designs.

2.4.3. Results and Discussion

Free vibration experiments were conducted for the large-scale model, similar to Section 2.3.4. Table 2.5 below lists the natural frequencies obtained from the experiments. As of the time of this writing, the large-scale model is being prepared to be shipped to the FIU WoW facility to perform wind tunnel tests. The results are going to be submitted for publication after the testing is completed.

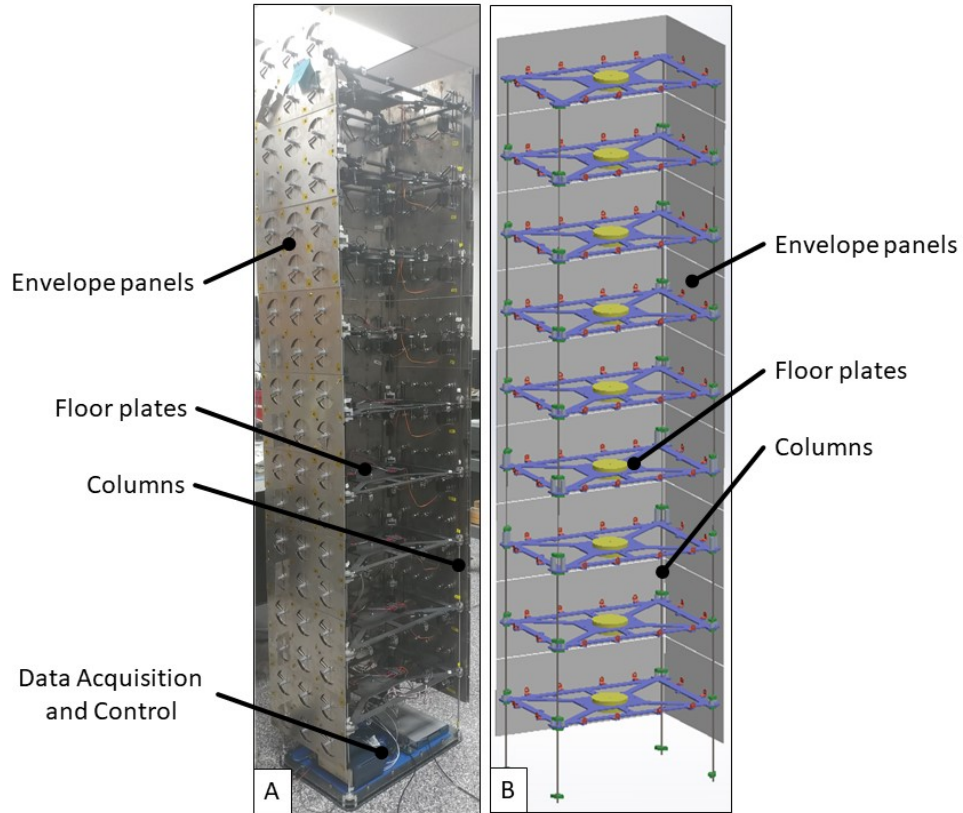


Figure 2.11. A picture of the large-scale CAARC building model with 9 lumped masses connected by four columns. The back envelope panels are removed to show internal details.

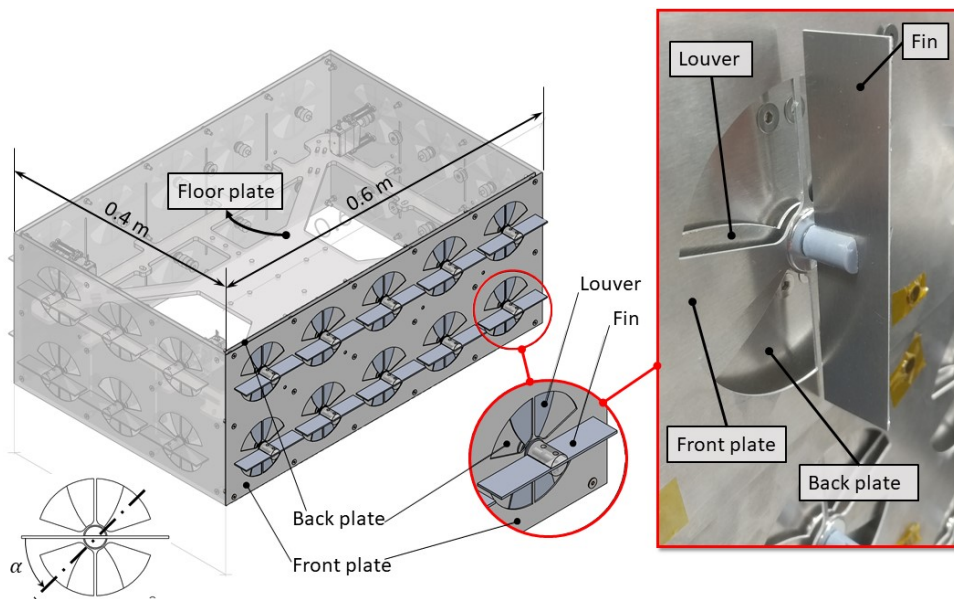


Figure 2.12. One floor of the large-scale CAARC model showing the floor dimensions and the horizontal fin-louver active AM design and the convention for measuring its orientation angle α

Table 2.5 Natural frequencies of the large-scale model.

Parameter	Actual
Natural frequency ratios	
ω_{θ}/ω_x	2.6 / 1.7 Hz = 1.5
ω_{θ}/ω_y	2.6 / 1.6 Hz = 1.6

2.5. Conclusions

This chapter presented two main approaches that are used in next chapters to develop WIV minimization controllers that use active aerodynamic modification (AM) surfaces: (1) Fluid-structure-interaction (FSI) simulations and (2) wind tunnel experiments of scaled robotic aeroelastic building models. The Commonwealth Aeronautical Advisor Research Committee (CAARC) standard building was used as a base case for both approaches. The FSI simulation model was designed to enable the rotation of the active AM plates without requiring reconstruction of the mesh. This was achieved using interpolation interfaces around the plate rotation zones. The initial results from the FSI simulation showed the great effect of the active AM plate orientations on the building's WIV response. This finding constitutes a cornerstone to the controller development presented in next Chapters.

The second approach presented in this chapter involved small and large-scale wind tunnel experiments conducted on robotic aeroelastic building models. The small-scale (1:400) building model was designed using a genetic algorithm (GA) procedure that matched its dynamic properties to the full-scale building. The characterization of the resulting manufactured model showed minor deviations from the design values which verifies the model's comparability to the full-scale building.

The large-scale model (1:73) allowed for the inclusion of 9 lumped masses, instead of 4 for the small-scale model. This is expected to enable the model to simulate higher modes of oscillation. The increase in available construction space also allowed the model to accommodate

different active AM designs and additional AM control surfaces, both of which are expected to better demonstrate the effectiveness of the controllers developed in next chapters. As of the time of this writing, the large-scale model is being prepared for shipping to the Natural Hazards Engineering Research Infrastructure (NHERI) Wall-of-Wind (WoW) facility for testing. The results of these tests will be submitted for publication as soon as they are completed.

Chapter 3 - Steady State Optimum Control

The previous chapter presented different methods used to model the WIV response of the building. This chapter utilizes these models to perform optimal control of the active AM system. The control techniques discussed in this chapter assume that the wind condition can be fully represented using average readings of the wind speed and direction $\bar{w} = [\bar{U}, AOA]$ (see Figure 2.1). Chapter 4 will present and address some practical limitations of this approach. Using the average wind condition $\bar{w}$, the controller provides plate orientations that minimize the WIV response as discussed in the following sections.

3.1 Controller Design

The proposed controller operates by defining a performance function $Q(\bar{w}, \alpha_i)$, which is calculated knowing the building's WIV deflections or accelerations. The exact mathematical representation of Q is presented for each controller implementation separately (see Section 3.2. Genetic Algorithm Training and 3.3. Iterative Optimum Training). The Q function treats the wind-building-plates system as a reduced order system defined only by the inputs (plate orientation angles α_i , average wind condition $\bar{w}$) and the outputs (building WIV response). This allows the controller to operate efficiently without requiring knowledge of the complex turbulent flow surrounding the plates and the building. It is understood that the position of a single plate has the potential to influence the flow around a nearby or downstream plate; however, instead of trying to observe and quantify those interactions the controller operates on parameters that are more practical to measure (e.g., building motion) using existing sensing techniques. Figure 3.1 illustrates the architecture of the developed controller, which uses Two multi-layer perceptron (MLP) DNNs to issue plate angle control actions and ensure long-term adaptability. The predictor MLP $\hat{Q}(\bar{w}, \alpha_i)$ acts as a regression model that provides the expected building WIV

performance $\hat{Q}$. The optimizer MLP $\alpha_i^*(\bar{w})$ provides the optimum plate orientations α_i^* for a given wind condition $\bar{w}$. Sufficient averaging of wind condition readings can be performed to reduce plate actuation speed requirements. After the controller issues plate angle commands using the optimizer MLP, the WIV performance Q is measured using accelerometers mounted on the building. The controller keeps track of the WIV performance Q that results at different plate angle combinations and wind conditions so they can be used to periodically train the predictor MLP. The background optimizer is a GA optimizer which uses the predictor MLP as a meta model. It constantly tries to find better plate angle combinations for the wind conditions being encountered by the controller. The findings of the background optimizer are used to periodically train the optimizer MLP when a better solution is found. The background optimizer does not have to be dispatched for every wind condition that the system encounters; therefore, it can operate locally using any remaining computational capacity, or remotely on separate computational machines. MLP DNNs are used to achieve WIV control because: (1) they can generalize from encountered to arbitrary conditions, which reduces the training effort; and (2) they can be used in real-time, because their evaluation is computationally inexpensive. While demonstrating the adaptation capabilities of the steady-state controller is not achieved in this study, predictor-optimizer DNN-based controllers are known in the literature to have such capabilities (Noriega and Wang 1998). Adaptation will enable the controller to respond to changes in the surrounding environment, such as erection of adjacent buildings. The following sections discuss two different approaches that were used to train the illustrated controller architecture

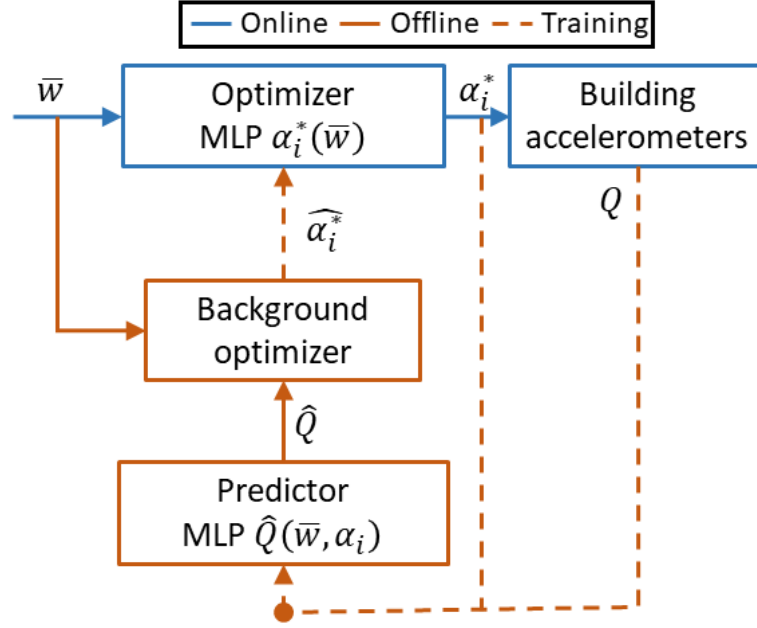


Figure 3.1. Steady state control architecture for the wind-building-plate system assuming a performance function Q

3.2. Genetic Algorithm Training

In this section, GAs are used to train the controller discussed in the previous section. The FSI simulation procedure discussed in Section 2.2 is used to estimate the WIV performance Q . GA optimization is used to find plate orientations α_i that improve the WIV performance Q for different wind conditions $\bar{w}$. The optimization results are used to initiate the two DNNs which are utilized by the controller.

3.2.1. Offline optimization

In reference to the CAARC building model description discussed in Section 2.1, The WIV performance Q represents an inverted sum of building deflection variances as follows:

$$\max. Q(\bar{w}, \alpha_i) = \frac{0.1}{\sqrt{\sigma_x^2 + \sigma_y^2 + 10^8 \sigma_\theta^2}} \quad 0 \leq \alpha_i \leq \pi \quad i = 1, 2, 3, 4 \quad (3.1)$$

where σ^2 is the statistical variance of building floor oscillations for all DOFs, which represents the deflection amplitude, and α_i is the rotation angle of plate i as illustrated in Figure 2.1. Because the building deflection angle is measured in radians, its variance σ_θ^2 is multiplied by 10^8 to make all three variances equally significant. The variance sum is evaluated using the FSI simulation model discussed in Section 2.2. The statistical variances σ^2 are defined using 30 seconds of deflection data. This time length is determined using an initial experiment where the data collection is run infinitely while the variance sum is being calculated for the accumulated data simultaneously. The final time length chosen (30 seconds) is the minimum amount of time it took the variances to become constant, which indicates that the building is showing stable steady-state response. Though the instantaneous forces acting on the building have higher inaccuracies with the RANS $k-\omega$ SST turbulence model for high flow separation cases, Antonini et al. (2019) showed that averaging the resulting structural response minimizes these inaccuracies; thus making the RANS model a more efficient valid option for predicting the time-averaged structural response as discussed in Section 2.2.1 (Antonini et al. 2019). To start the optimization, the objective function Q was evaluated for different randomly chosen plate angle combinations. The relationship was not found to be monotonic, which means that multiple local optima may exist. Therefore, a global optimization technique, such as GAs, was used instead of a direct search method to avoid convergence to local optimum solutions. A GA program was written in C++ that operates one generation at a time using an input/output “generation” file (comma-separated list). If there was no previous generation file provided, the GA program begins by generating a uniformly random population of 70 different plate angle combinations and writes them down to a newly created generation file. An independent simulation is performed for each plate angle combination, and the objective function is evaluated and inserted

into the same text file. The GA code reads the file again to determine the angles to be evaluated in the next generation. The 10 best performing plate angle combinations are copied to the next generation as elites. Fitness-proportionate selection (roulette wheel) is used to determine the remaining 60 combinations to be evaluated. Single-point cross over is performed with a probability of 0.8 as well as uniform mutation with a probability of 0.1. A new file is written for the new generation and the process is repeated until the GA stops finding better combinations. The population size and the mutation and cross-over rates were determined from recommendations in the literature (Angelova and Pencheva 2011). The developed GA was also tested against a few benchmark problems (e.g., 4-variable Schwefel function) to ensure proper implementation (Jamil and Yang 2013).

The optimization was run for a total of 7 generations, which involved performing approximately $7 \times 70 = 490$ simulations, each of which considered a different 4-plate rotation angle combination. It is expected that the interaction between the different values of plate rotation angles α_i will have a significant effect on the overall building response. The GA optimization and the control algorithm (see Section 3.1) account for these complex aeroelastic interactions indirectly by observing the time-averaged building response (Equation 3.1). The GA uses uniform crossover to recombine plate angle combinations where fitness proportionate selection makes it more likely to consider plate angle combinations that will reduce building vibration. Each simulation in the optimization utilizes 12 processor cores from the CRUNCH computational cluster at the Multifunctional Structures Laboratory (MSL) at Kansas State University, which allows up to 24 plate angle combinations to be simulated simultaneously. Figure 3.2 shows the GA convergence for 7 generations as well as the corresponding optimum

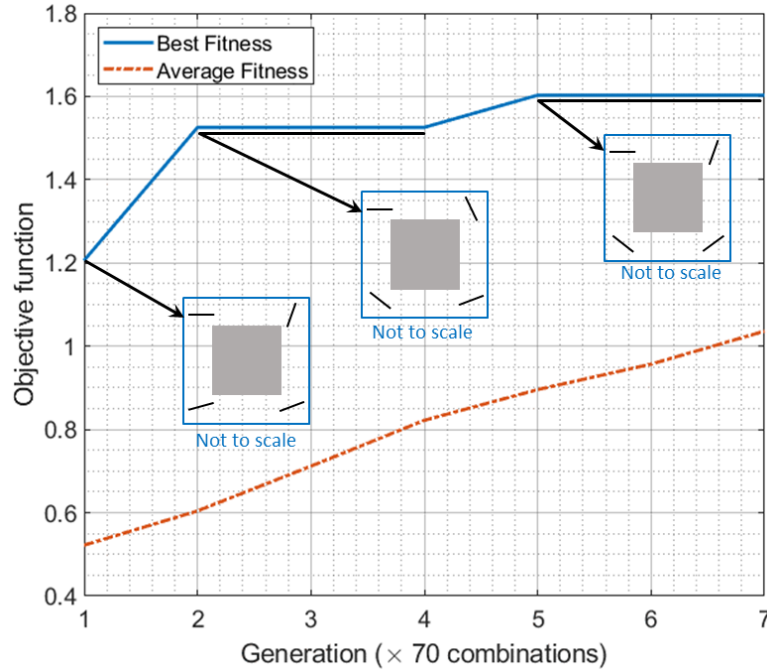


Figure 3.2. Convergence trends for the genetic algorithm optimization process showing the optimum plate configurations for different generations

plate combinations. The average fitness is shown to steadily increase from a generation to the next, which indicates the effectiveness of the optimization procedure. However, only two performance improvements were obtained relative to the initial generation. This suggests that running more generations may obtain better solutions, but the corresponding improvements are expected to be minimal compared to the additional computational effort. Further optimization using wind tunnel experiments is discussed in Section 3.3.

3.2.2. Optimization results

Figure 3.3 compares the time series for along-wind, cross-wind and torsional building oscillations, respectively, between the optimal and a suboptimal solution. The *suboptimal* solution is the average performance from the first GA generation (i.e., whose fitness approximately equals the average fitness in generation 1). The simulated steady state amplitudes of building oscillations show a decrease of 60-80%. It is important to note that these simulation

results may not quantitatively relate to a full-scale deployed system; however, they do indicate that the controller is able to operate as expected. This reduction can be explained in light of the PSD plot in Figure 3.4 where the optimum configuration decreases the force amplitude around the bending mode natural frequency of 0.6 Hz. This causes a more significant proportion of the available wind energy to avoid frequencies most sensitive to the building, which results in a lower amplitude response around that frequency as evident in the deflection PSD. As shown in the force PSD, the peak at the natural frequency (0.6 Hz) is not very prominent, which means that multiple phenomena may have been involved (e.g., turbulent buffeting). However, this minimal representation of the active AM system (only 4 plates) still considerably reduced building oscillations, which is a good indicator of robustness and prompts the wind tunnel experimental work discussed in the next sections.

3.2.3. Simulation-based training

According to the controller design discussed in Section 3.1, the current section illustrates the architecture and the training procedure for the predictor and the optimizer MLPs used by the controller. Various MLP architectures and configurations were considered with the goal of obtaining an accurate regression. The final architecture has two hidden layers, each consisting of 10 neurons which use the bipolar sigmoid activation function. This function was used because it is expected to be able to correctly map nonlinear behavior as well as positive and negative values. The predictor MLP was trained using ~500 different wind speed and plate combinations. Only 70% of this total was used to train the network using backpropagation learning. The learning was performed for minibatches of 16 data points. Stochastic gradient descent (SGD) was used as an optimization algorithm (Amari 1993). The learning rate, which controls the amount of change in the network weights per update, was set at 0.001.

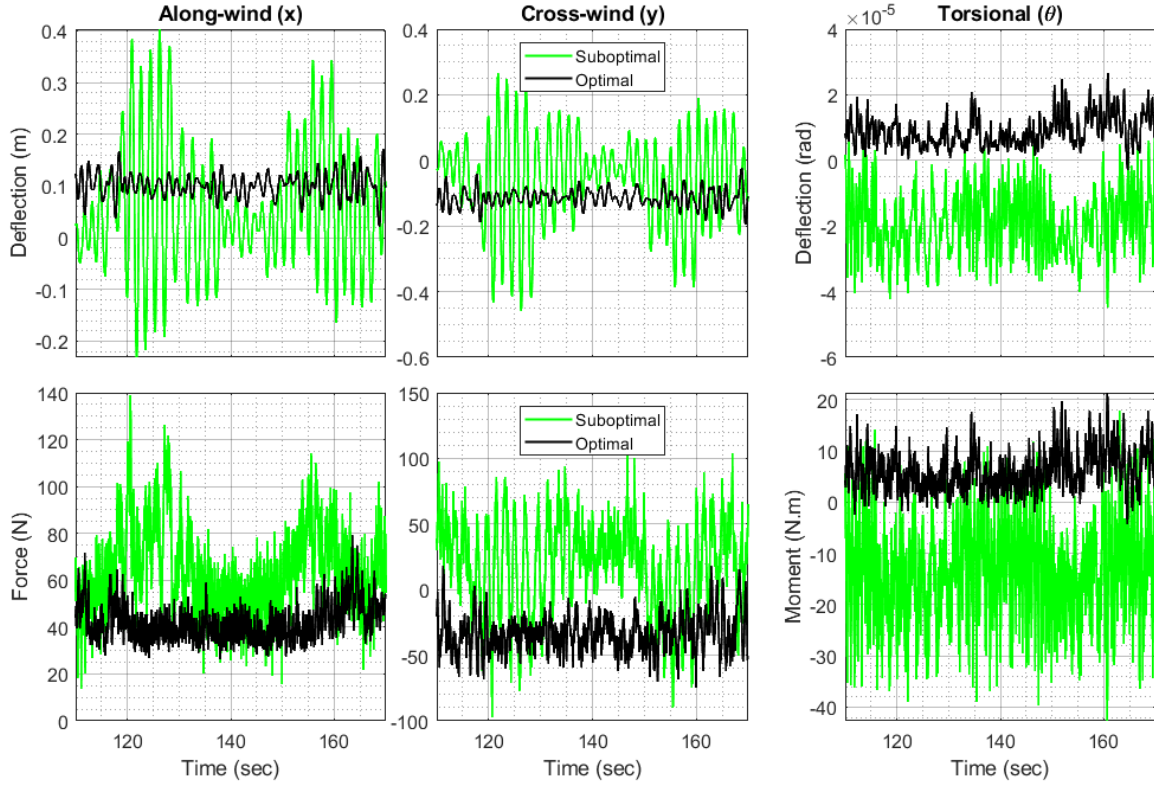


Figure 3.3. Comparison between the suboptimal and the optimal deflections and forces for all 3 DOFs

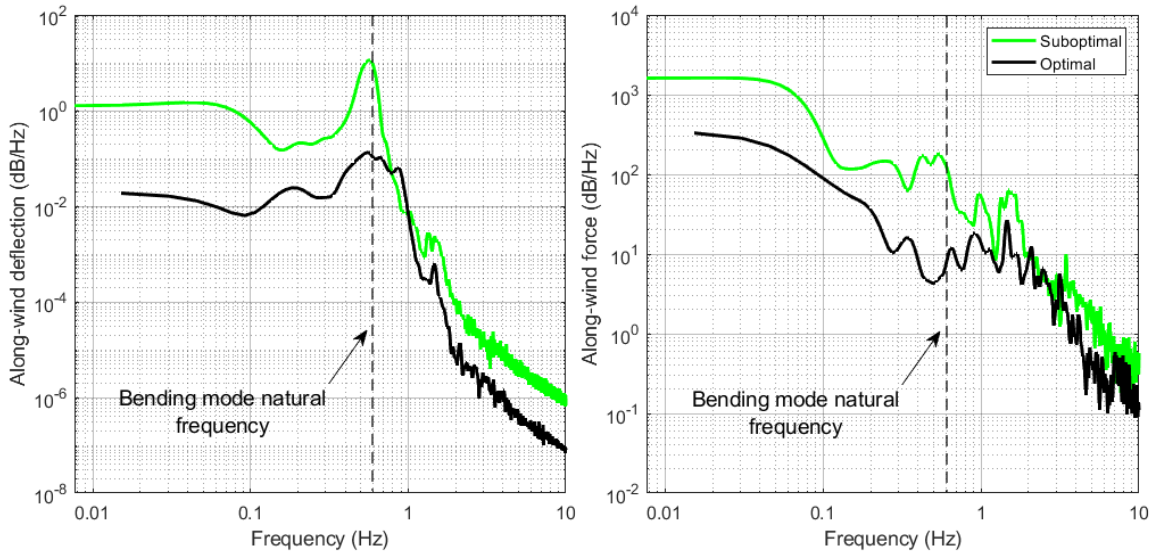


Figure 3.4. Comparison of the along-wind building deflection and wind force PSDs before and after plate angle optimization

The remaining 30% of the data was used only in the final testing stage to evaluate the root mean square error (RMSE) for the prediction. Figure 3.5 illustrates the performance of the predictor MLP for the range of plate angles simulated. An RMSE of 1 mm for the along-wind deflection was obtained, which is equal to an acceptable value of 40 mm for the full-scale building. Additionally, for a deployed system, as per the controller design (see Figure 3.1), the predictor MLP has a learning path (indicated by a dashed line in the figure) which can be performed periodically. This is expected to continuously improve the prediction accuracy and enable the predictor MLP to adjust to long term environmental or building changes. The optimizer MLP has the same architecture as the predictor MLP but has different inputs and outputs, as shown in Figure 3.1. Evaluating performance of the optimizer MLP using FSI simulations is a very time-consuming process, because the optimization discussed in Section 3.2.1 would need to be restarted for many different wind conditions. It was decided to perform this more exhaustive adaptive control evaluation via experiment, as shown in Section 3.3.

3.2.4. Controller Simulation

This section performs a simulation of the controller performance to gain insight on the amount of time it takes the plate angle changes to affect the wind system and obtain a tangible difference in the building oscillations. Figure 3.6 shows the along-wind time response of the building to the following simulation procedure: The wind speed at the inlet is set at 5 m/s, similar to the optimization case. The plates are set at a suboptimal configuration and the fluid domain is left to stabilize while the building deflection is held constant. This ensures the building won't be subjected to unrealistic fluid forces. After the fluid forces develop, the building is allowed to move, which causes building oscillations with approximately 10 mm amplitudes to ensue.

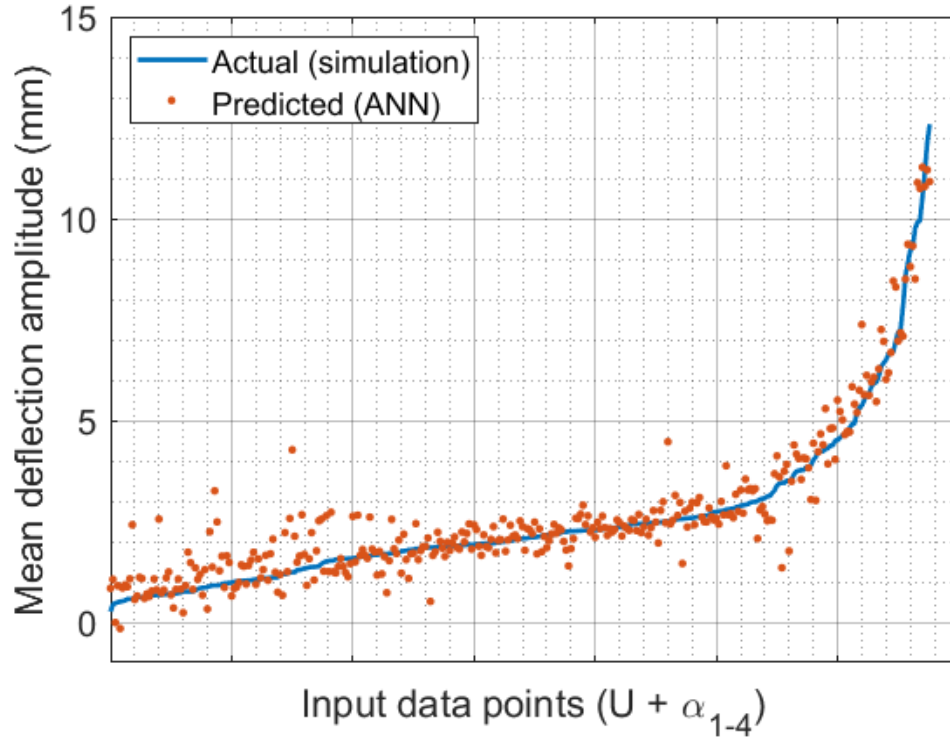


Figure 3.5. Result of predictor MLP training showing actual and predicted along-wind deflection amplitudes for various combinations of wind velocity and plate angles

When the controller changes plate angles to their optimum configuration for 5 m/s (see Section 3.2.2), the oscillations diminish to only 0.6 mm in approximately 7 seconds (94% reduction). A considerable reduction of 60% is obtained after only 3 seconds. Again, it is important to note that these results may not quantitatively represent real-life deployment performance due to the 2D simplifications made to the model. However, these results do bring validity to the proposed control scheme – the focus of this study – while motivating the scaled 3D building experiments discussed in Section 3.3.

These results demonstrate that the plates, building, and controller are able to respond fast enough to negate wind events that may last up to 20 seconds, or at least speed up the decay of the resulting oscillations after the events occur. Wind gusts usually last less than 20 seconds;

however, longer-lasting strong winds which are called squalls can continue for minutes. Tornadoes can last from several seconds to more than an hour, but are typically less than 10 minutes (U. S. Department of Commerce 2020; American Meteorological Society 2020). The life cycle of a hurricane can be anywhere from 12 hours to a month (Hurricane Science 2020). Thunderstorm winds may last tens of minutes, and local breezes generated by heating of land surfaces can last a few hours.

An important limiting factor is how fast the plates can be rotated to new locations. This can be considered when designing the full system by making the plates light enough to allow for speedy actuation. Reducing plate weight may be accomplished numerous ways including reducing plate height and making them from lightweight material (e.g. composites). Adding more plates (instead of only 4) and reducing their size may also help to improve plate rotation speeds. While these modifications are viable approaches that could be considered in the context of a parametric study to improve the performance of the system, they are outside the scope of this study which focuses on underlining the effectiveness of the control algorithm.

3.3. Iterative Optimum Training

The previous section proved the effectiveness of using GAs to generate training data for the WIV controller discussed in Section 3.1. However, the process involved performing many FSI simulations to evaluate even a single wind condition. A difficult or time-consuming training operation can greatly hinder the adoption of the developed WIV controller. This part of the study

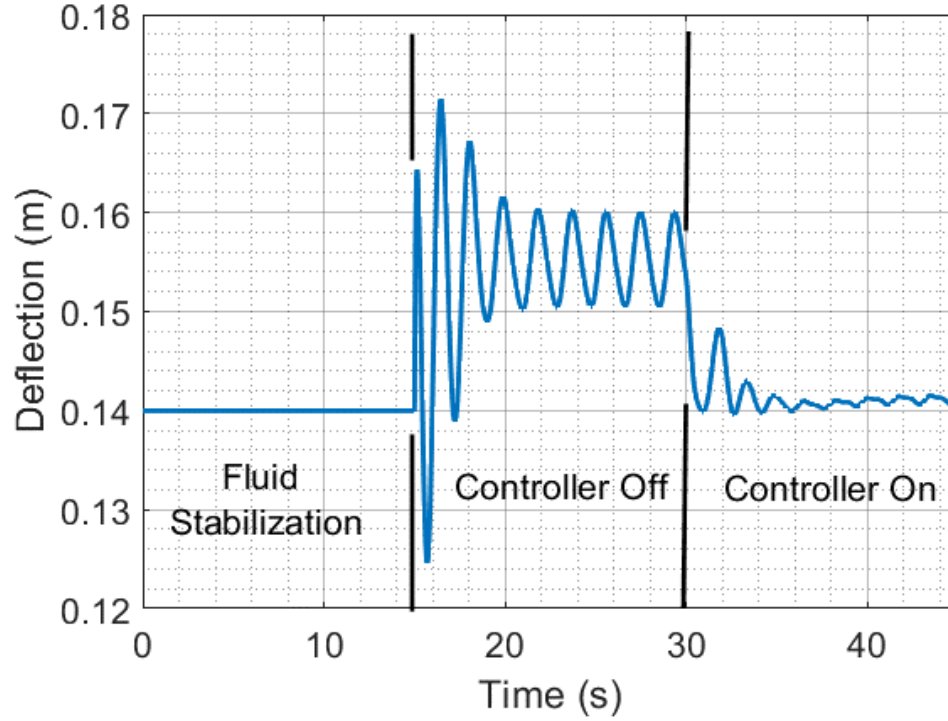


Figure 3.6. Along-wind deflection changes as a result of employing the developed controller

presents a simultaneous training and optimization technique that greatly improves the WIV controller training performance by overcoming some the typical surrogate-based optimization drawbacks discussed in Section 1.1.4 (Haftka et al. 2016). Additionally, wind tunnel experiments of the 1:400-scale robotic aeroelastic building model discussed in Section 2.3 were used instead of FSI simulations. This eliminates the inaccuracies resulting from the 2D simplification of the FSI simulation model and greatly reduces the total training time. The main strategy of the developed iterative optimum training (IOT) technique is to perform the experiments selectively based on the progression of the optimization instead of randomly or based on a grid of the parameters. This approach serves two purposes: (1) the training can be performed with a smaller number of experiments and (2) the resulting MLP is more densely trained around the optimum operating control parameters. Point (2) is especially relevant for optimally controlling a system because control parameters providing for better system performance are prioritized in the

training, which increases the likelihood of generalizing the *optimum modes* of performance for the system. The IOT technique presents a modification to the original MLP-GA procedure discussed in Section 1.1.4. At first, a relatively small initial set of experiments is used to train the MLP. The GA operates using the MLP instead of requiring additional experiments. A set of best solutions from the GA are evaluated using additional experiments and added to the training data set to improve the training around the current optima to enhance exploitation. Similarly, another set of uniformly random solutions are evaluated and added to the data set to enhance exploration. The expanded data set is used to retrain the MLP which is used again by the GA. The process is repeated until the stopping criteria are met. Each iteration only requires a small number of experiments, relative to the initial experiments, to evaluate the solutions obtained from the GA. Because IOT starts from a smaller initial sample set, it may locate the global optimum by using significantly fewer experiments. Also, because the training data set is expanded with random samples at each iteration, IOT does not rely on the initial sample being representative of the whole domain, so it has no inherent stall mechanism (as opposed to pure GAs). In the proposed framework, the GA can be replaced by any other population-based optimization technique (e.g., particle swarm, ant colony, etc.) to benefit from the iterative nature of the training. Huang *et al.* (2015) used a similar iteration technique to optimize the thickness of blow molded parts (Huang et al. 2015). In that work, the authors expand the data set by only one sample obtained from the GA optimization after each iteration without adding random samples. Therefore, the choice of this one sample is more critical for the progression of the search, so the authors (1) used design of experiments (DOE) to obtain the initial sample set as a non-iterative exploration mechanism and (2) integrated GAs and simulated annealing (SA) to improve the local search ability of the GA to enhance exploitation. In the current study, multiple samples are added from the GA and at

random to enhance both exploitation and exploration, which simplifies the implementation. Additionally, the current study uses the technique for online training of a data-driven optimum controller and evaluates the technique by comparing it to others using a benchmark function.

3.3.1. Implementation

For this version of the steady-state controller discussed in Section 3.1, the primary function of the controller is to obtain plate angle combinations α_i that minimize WIV accelerations at different average wind conditions $\bar{w}$. In reference to the CAARC building case illustrated in Section 2.1, the WIV performance Q is redefined in this version of the controller as,

$$\text{Min. } Q(\bar{w}, \alpha_i) = \text{RMS}(\ddot{x}) + \text{RMS}(\ddot{y}) + \frac{B}{2} \text{RMS}(\ddot{\theta}) \quad (3.2)$$

where α_i are the plate angles between 0 and 180 degrees, subscript i denotes the plate number, $\bar{w} = f(\bar{U}, AOA)$ is the time-averaged wind condition which consists of the average wind speed $\bar{U}$ and the angle of attack AOA , RMS is the root-mean-square for a defined interval of acceleration readings, and $\ddot{x}$, $\ddot{y}$, $\ddot{\theta}$ are the along-wind sway, cross-wind sway and torsional accelerations, respectively. The RMS of $\ddot{\theta}$ has units of rad/s^2 as opposed to m/s^2 for the sway components, so it is numerically scaled by half of the building width, which represents the radius of twist. Compared to the previous version of the controller (see Section 3.2), this version uses a sum of RMS accelerations instead of a sum of deflection variances/amplitudes. This change is supported by evidence in the literature that humans have a stronger perception of accelerations rather than deflections, which makes them more important for human comfort (Jafari and Alipour 2021).

Figure 3.7 illustrates the MLP network used by the IOT procedure to approximate $Q(\bar{w}, \alpha_i)$. The hidden layers of the network contain neurons using exponential linear unit (ELU) activation functions defined as,

$$ELU(x) = \begin{cases} x & x \geq 0 \\ e^x - 1 & x < 0 \end{cases} \quad (3.3)$$

where x is the neuron input, due to their low computational cost and proven performance (Clevert et al. 2016). The output of the MLP network is a sigmoid activation function ($\sigma(x) = 1/(1 + e^{-x})$) because its result is scaled between 0.0 and 1.0, which is suitable for representing all potential values of Equation 3.2.

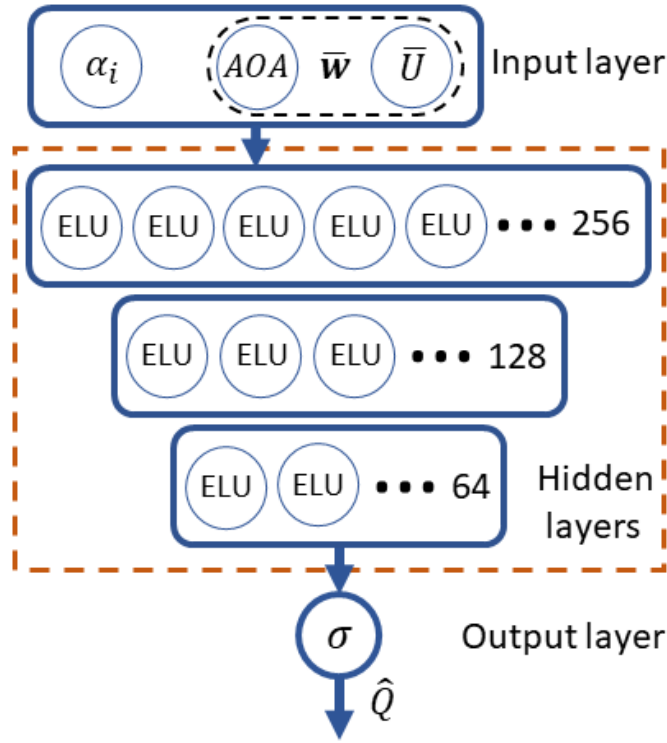


Figure 3.7. The MLP $\hat{Q}(\bar{w}, \alpha_i)$ network used to implement the IOT procedure.

Figure 3.8 is a flow chart illustrating the IOT procedure. For each wind condition, initial training for the MLP is performed using k initial experiments. The GA utilizes the trained MLP rather than requiring additional experiments (inverse ANN). The best $r_1 \times k$ solutions found by the GA are evaluated using additional experiments, where r_1 is the GA expansion ratio. If the stopping criteria are met, the procedure terminates for the current wind condition $\bar{w}_j$. Otherwise,

additional experiments are also performed for $r_2 \times k$ uniform random samples, where r_2 is the random expansion ratio. The original samples k are expanded by the summation of the new samples $(r_1 + r_2)k$. The MLP is retrained using the expanded set of samples and the process repeats until stopping criteria are met. The way in which the original samples k are expanded after each iteration greatly affects the performance of the IOT technique. The $r_1 \times k$ solutions found by the GA help improve future training accuracy around the current optima, which represents an exploitation mechanism, while the $r_2 \times k$ additional random solutions prevent the IOT technique from stalling if the initial samples k had no representation of the global optimum solution, which represents an exploration mechanism. The next section discusses how these two mechanisms improve the solution quality and reduce the number of experiments required to perform optimum training. In addition to generating the trained MLP $\hat{Q}(\bar{w}, \alpha_i)$, the procedure also stores the optimum plate angle combinations $[\alpha_i]_j^*$ associated with each wind condition encountered during training. These plate angles are used to initiate the controller illustrated in Section 3.1.

3.3.2. Numerical benchmark

This section uses a benchmark function to provide comparisons to other techniques and establish the training/optimization performance gains associated with the IOT technique. The Alpine 2 function was used to test, develop, and compare the IOT technique before using it with actual wind tunnel experiments (Clerc 1999). This function was chosen mainly because it is multimodal and non-separable. This behavior is expected in the wind-plate-building system because the wake forming behind a plate significantly affects the plate behind it, which causes non-separability. The Alpine 2 function is given as (Clerc 1999),

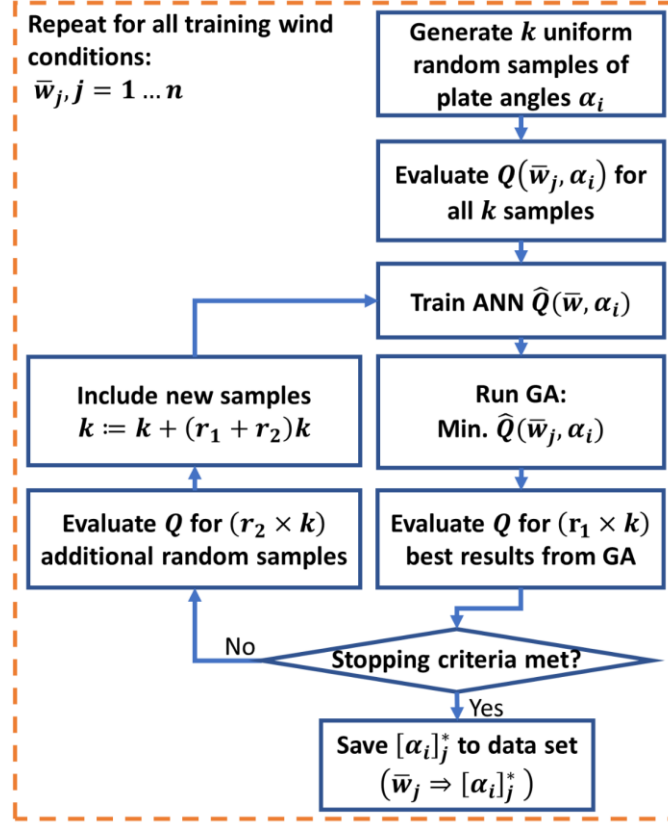


Figure 3.8. Flow chart of the IOT technique for minimizing Equation 3.2 while simultaneously training the MLP.

$$Q(X) = Q(x_1, \dots, x_p) = \prod_{i=1}^p \sqrt{x_i} \sin(x_i) \quad (3.4)$$

where p is the number of dimensions and $x_i \in [1, 10]$ are the design variables. The function is continuous and has a global maximum $f(X^*) = 2.808^p$ at $x_i = 7.917$.

Table 3.1 lists parameters of the compared online training techniques, including the developed IOT technique. The first technique considered is a direct GA that does not train or utilize an MLP. The GA generation size was varied among 62 different values, as shown in Table 3.1. For each generation size, 30 independent trials were performed. The maximum number of generations for each trial was set at 50 to prevent a stalling GA from halting the progress of the comparison. The second technique considered was a GA that utilizes an MLP

(MLP-GA) that is trained from a constant number of samples without iteration, similar to several studies in the literature (Dasgupta et al. 2017; Márquez-Nolasco et al. 2017; Conde-Gutiérrez et al. 2018). The training sample size was varied among 504 different values, as shown in Table 3.1. For each sample size, 6 independent trials were performed. Finally, the IOT technique developed in this work and illustrated in the previous section was compared to the other techniques. In reference to Figure 3.8, all parameters $x_1 \dots x_p$ are assumed controllable, so $n = 1$. The initial sample size k was varied among 56 different initial sample sizes, as shown in the table. For each initial sample size, 24 independent trials were performed. At each iteration, the sample size was expanded by 25% using the best solutions from the GA that uses the trained MLP and by another 25% using uniform random samples ($r_1 = r_2 = 0.25$ in Figure 3.8). Except for the iterative modification, all MLP architecture and training parameters are identical among MLP-GA and IOT. The stopping criterion and all other parameters of the three techniques are listed in Table 3.1. The generation size and number of samples and/or trials for each technique were determined to obtain a fair comparison between the total number of samples required by each technique. This is evident in Figure 3.9, which plots the normalized objective function (explained in Table 3.1) versus the total number of samples for all three techniques. In many instances, the GA technique is shown to achieve low relative objective values (0.7-0.8) at a relatively low number of samples (< 2000). This behavior seems to decrease as the number of samples increases until it disappears at ~ 4000 samples. This indicates stalled GA runs, in which the initial population fails to contain a representation of the global optimum solution. The MLP-GA technique shows a large variance of the obtained optima at small sample sizes. That variance decreases as the sample size increases. Additionally, the mean relative objective obtained by MLP-GA only manages to match the GA technique at ~ 2800 samples and the IOT technique at

~3500 samples. These drawbacks of MLP-GA happen because small sample sizes tend to fail to produce an MLP whose prediction correctly represents the behavior function around the global optimum. The IOT technique presented in this work has the means to alleviate these drawbacks. At each iteration, when the stopping criterion is not met, the original sample k is expanded in light of previous iterations as well as with additional random samples. This causes the IOT technique to have no stall mechanism such as the case with the GA technique; therefore, the IOT solution always satisfies the stopping criterion shown in Table 3.1. Also, because the IOT starts from a small initial sample k , it can still satisfy the stopping criterion early when smaller samples manage to properly train the MLP. These observations explain that the IOT fitted curve consistently provides the best relative objective for the total number of samples consumed.

3.3.3. Experiment-based training

This section discusses the wind tunnel training process for the WIV controller, which follows the IOT technique discussed in Section 3.3.1. The system was trained at three critical wind directions (AOA): 0° , 56° (diagonal) and 90° . The RMS accelerations observed for $AOA=56^\circ$ were considerably lower than those for $AOA=0^\circ$ and 90° , which coincides with findings in the literature (Hou and Sarkar 2020). Given such low accelerations, controller performance could not be demonstrated at that direction. Therefore, a special case was considered only for $AOA=56^\circ$ where two rigid scaled building models were positioned upstream of the principal model as illustrated in Figure 3.10. The presence of these upstream buildings created additional flow features (e.g., vortices and large-scale turbulence), which increased principal model accelerations thereby enabling demonstration of the WIV controller. A total of 9 average wind speeds were tested at each wind direction, so a total of 27 distinct wind conditions $\bar{w}$ were used for training. The wind speed values for each wind direction are illustrated in Figure 3.11.

Table 3.1. Parameters of the compared techniques training and optimization techniques

Parameter	GA	MLP-GA	IOT (this work)
Dimensions p	4		
Selection method	Fitness proportionate (roulette wheel)		
Crossover	Single point, 80%		
Mutation	Uniform, 10%		
Elitism	12.5% of population		
Training samples	N/A	[64, 72 ... 4088, 4096]	[64, 72 ... 504, 512]
Population	[16, 24 ... 504, 512]	1024	
Generations	50	50	20
MLP architecture	N/A	Same as Figure 3.7, but neuron counts are 512, 256, 128 to fit larger sample counts of benchmark	
MLP training	N/A	Backpropagation using Adam algorithm (Kingma and Ba 2017)	
Optimum fitness	$2.808^4 = 62.17$		
Expansion ratios	N/A	N/A	$r_1 = r_2 = 25\%$
Stopping criterion	$Q \geq 59.10$ ($95\% \times 62.17$)	N/A (no iteration)	$Q \geq 59.10$ ($95\% \times 62.17$)
Normalized objective	$\frac{Q - Min.}{Max. - Min.} = \frac{Q - (-62.17)}{2 \times 62.17}$		

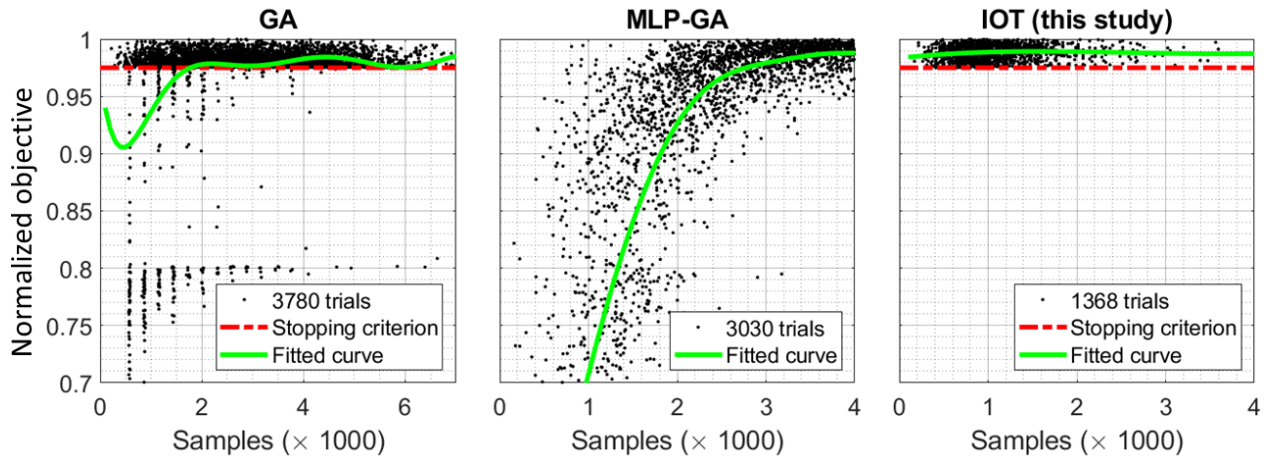


Figure 3.9. Performance comparisons of the tested online training techniques.

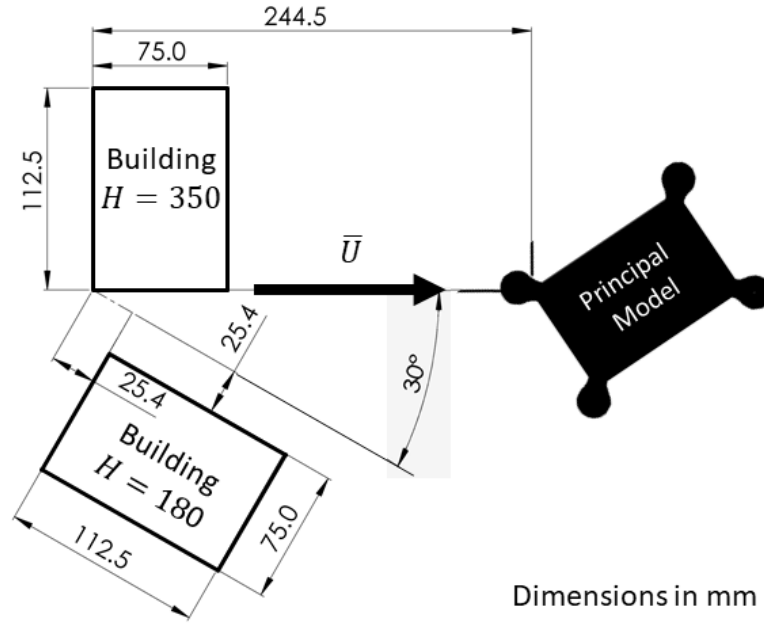


Figure 3.10. Layout of the upstream buildings added to increase the RMS accelerations at the diagonal critical direction ($AOA=56^\circ$)

In reference to Figure 3.8, Table 3.2 lists the IOT technique parameters that were employed in the wind tunnel training process. Two iterations were used as a stopping criterion because it was observed that the RMS reduction is negligible thereafter. These parameters translate to 60 experiments per wind condition, which equals ~ 1620 experiments in total for the 27 wind conditions considered. The experiment control program is provided with a different list of plate angle combinations to test at each wind condition. For each combination, the acceleration data is collected for 2 minutes and written to a comma-separated file. The files from all combinations are later post-processed to calculate the WIV performance Q and continue the IOT procedure.

Table 3.2. IOT technique parameters employed for the wind tunnel training process.

Parameter	n	k	r_1	r_2	iterations
Value	27	40	12.5%	12.5%	2

3.3.4. Results and Discussion

This section discusses the performance of the IOT-based WIV controller trained from wind tunnel experiments of the 1:400-scaled CAARC building model. A brief consideration of the expected real-life performance of the cyber-physical system is also included.

3.3.4.1. IOT Performance

Figure 3.11 plots the WIV performance Q in a normalized form (Q/f^2L) versus the reduced velocity ($V_r = \bar{U}/fL$) for the tested wind conditions. Where f is the along-wind sway natural frequency, L is the length scale ($L = \sqrt{DB}$) and $\bar{U}$ is the average wind speed. To isolate the effect of the plates, the figure also plots the neutral condition considered when the plates are parallel to the flow for minimum engagement. For $AOA=0^\circ$, the neutral position was found to generate relatively low RMS values, but some plate orientations maximized the RMS response considerably at $V_r \geq 9.7$. The best obtained plate orientations after two iterations are always at or below the neutral position, which indicates the effectiveness of the IOT optimization procedure. A similar trend is obtained at $AOA=56^\circ$ (with upstream buildings shown in Figure 3.10), but the neutral RMS response increases with V_r until the end of the test range. This case also shows that the environment surrounding the building can generate considerably high RMS responses even at wind conditions that are normally safe. Which further justifies the need for active AM control at all wind conditions. The maximum RMS response is obtained at $AOA=90^\circ$ for $V_r > 8.7$. This coincides with findings in the literature (Hou and Sarkar 2020). At $V_r = 9.7$, the neutral position

response surpasses the worst plate orientations found during training. These extreme vibration levels caused some of the model columns to fail from fatigue at $V_r = 10.1$. The failed columns were later replaced, and no plate optimization was performed at this velocity or beyond. Figure 3.11 also shows that in many cases, certain plate orientations maximize the RMS accelerations. This justifies the need for actively controlled AMs because static modifications may have a considerably negative contribution in unforeseen flow conditions. The maximum reduction obtained in the WIV performance Q is 81%, 41% and 95% for the 0° , 56° and 90° directions, respectively. This major reduction underlines the overall effectiveness of the cyber-physical building system.

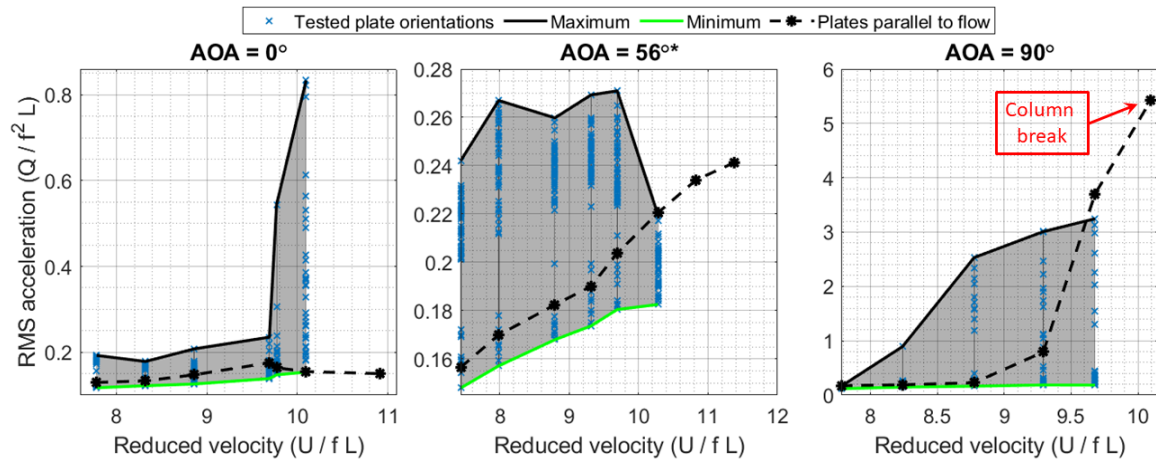


Figure 3.11. RMS acceleration trends for all tested wind conditions obtained after two IOT iterations (★ with the upstream buildings illustrated in Figure 3.10).

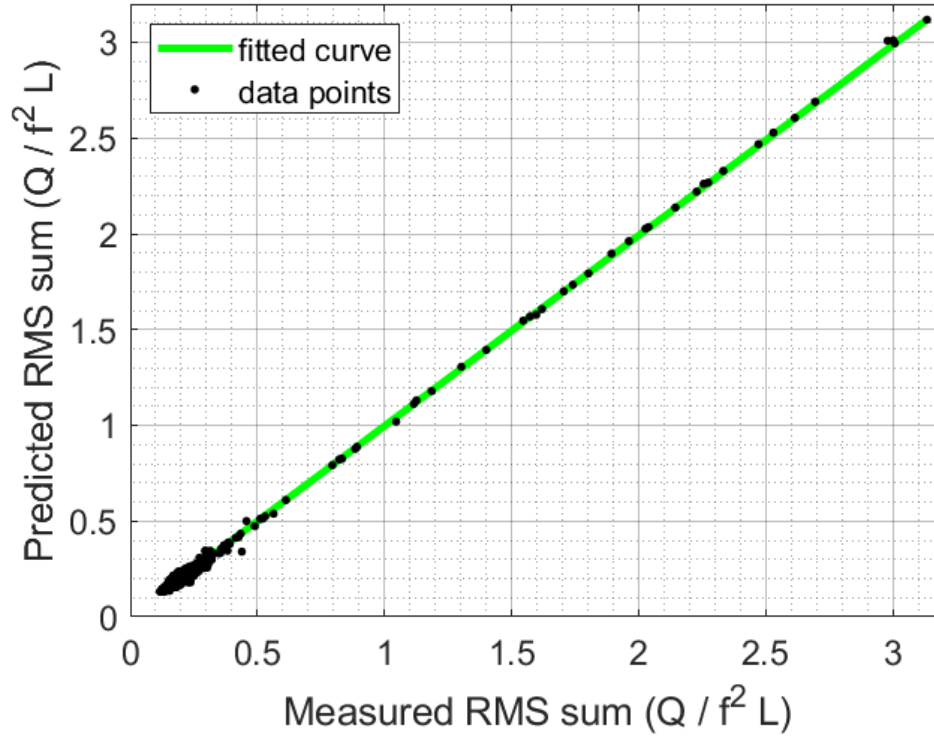


Figure 3.12. RMS acceleration prediction performance of the MLP trained using the IOT procedure.

Figure 3.12 illustrates the prediction performance of the predictor MLP which was trained using the IOT procedure. The average absolute error in the prediction is only 0.3% the maximum RMS sum and the R-squared value is 0.9985. The small relative error and the high R-squared value mean that the trained MLP is suitable for online control of the system, as discussed in the next section. Also, due to the simultaneous optimization and training nature of the IOT procedure, most of the training points exist at the low end of RMS sum values, as shown in the figure.

3.3.4.2 Optimum performance

Figure 3.13 compares the timeseries of the normalized 3-DOF accelerations of the scaled model obtained using the best and the worst plate angle combinations for each wind direction. The speed at which the plot is generated for each direction has the maximum RMS acceleration

in Figure 3.11 to clearly demonstrate the effect of the plates. The reduction is considerable at $AOA=56^\circ$ and major at both $AOA=0^\circ$, 90° which agrees with the RMS acceleration results in Figure 3.11. Additionally, Figure 3.14 shows the corresponding along-wind acceleration PSDs for each time series. For the cases without buildings upstream (0° and 90°), the minimum acceleration response is associated with a shift of the peak frequency away from the structural natural frequencies indicated on the figure. For the case with upstream buildings (56°), the turbulence-induced wind forces on the principal model are distributed over a broader band of frequencies compared to the relatively narrowband vortex-induced forces that are most prominent for 0° and 90° cases. Also, in this configuration, plate number 4 is completely blocked by the building, which reduces its wind control authority. These factors help explain why the controller reduces the RMS response by only 41% for this case as opposed to 80-90% for the other two cases. While the current active AM configuration may still be more effective for other surrounding environments, future studies involving a higher number of active AM plates on the building's perimeter are being considered.

3.3.4.3. Control response time

Figure 3.15 plots the acceleration time series of all 3 DOFs continuously before and after the plate angles were changed from the worst to the best configuration (at $AOA=90^\circ$ and $V_r = 9.7$). The figure was obtained by keeping the plates at a known poor configuration for a given wind condition and keeping the controller inactive for 60 seconds. This allowed large magnitude oscillations to develop. Then, the controller was activated, and it rotated the plates to the optimum configuration. Acceleration and plate position data was collected throughout the experiment. The figure also shows the running RMS accelerations calculated using a sliding window width of 2 seconds. The acceleration amplitude for rotation (dominant DOF) decreases

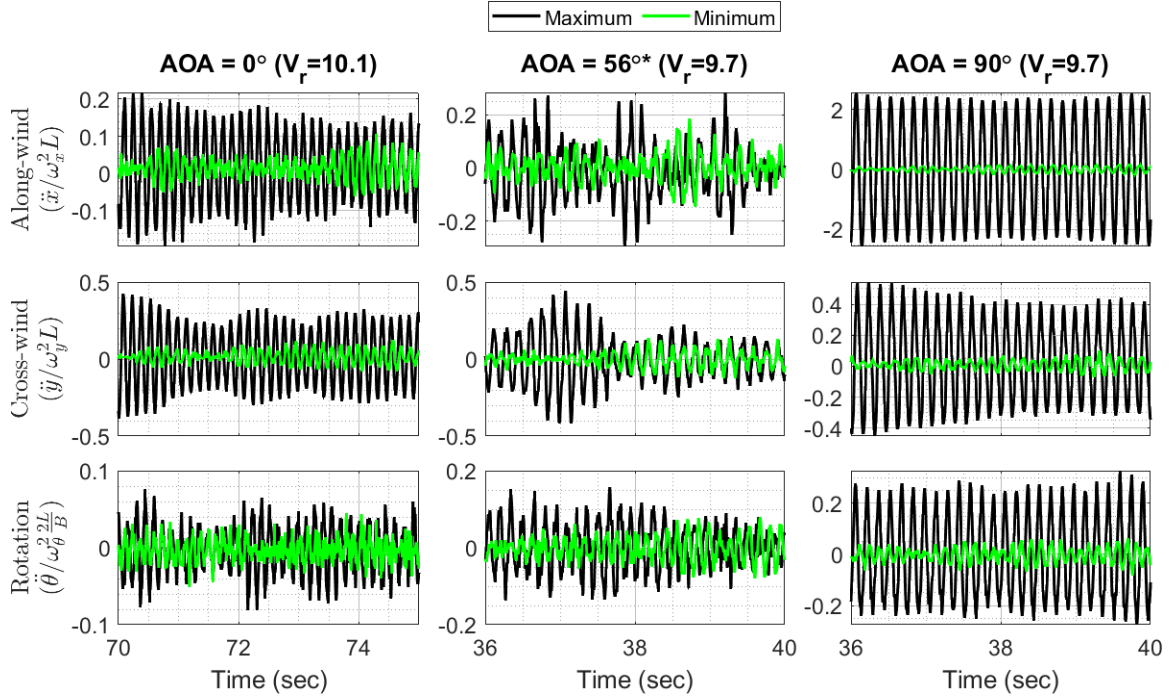


Figure 3.13. RMS acceleration prediction performance of the MLP trained using the IOT procedure.

by 33%, 61% and 90% after 1.1, 3.4 and 9.8 seconds, respectively. These reductions constitute major improvements in the model's WIV response. Additionally, for a deployed system, the plates will be actuated as soon as the adverse wind condition is encountered rather than waiting as was done for this demonstration. In other words, with the controller continuously activated, accelerations would not have a chance to reach these high values and the active AM actuation would require less time. At the 1/33 time scale listed in Table 2.2, the response time of 1.1 seconds corresponds to 36.3 seconds for the full-scale building. It is known that wind gusts typically last for less than 20 seconds, while squalls can continue for minutes at a time (U. S. Department of Commerce 2020; American Meteorological Society 2020). The proximity of the response time to the wind event time scales establishes the applicability of the proposed four-plate active AM system to mitigate WIV. However, to account for different environments and

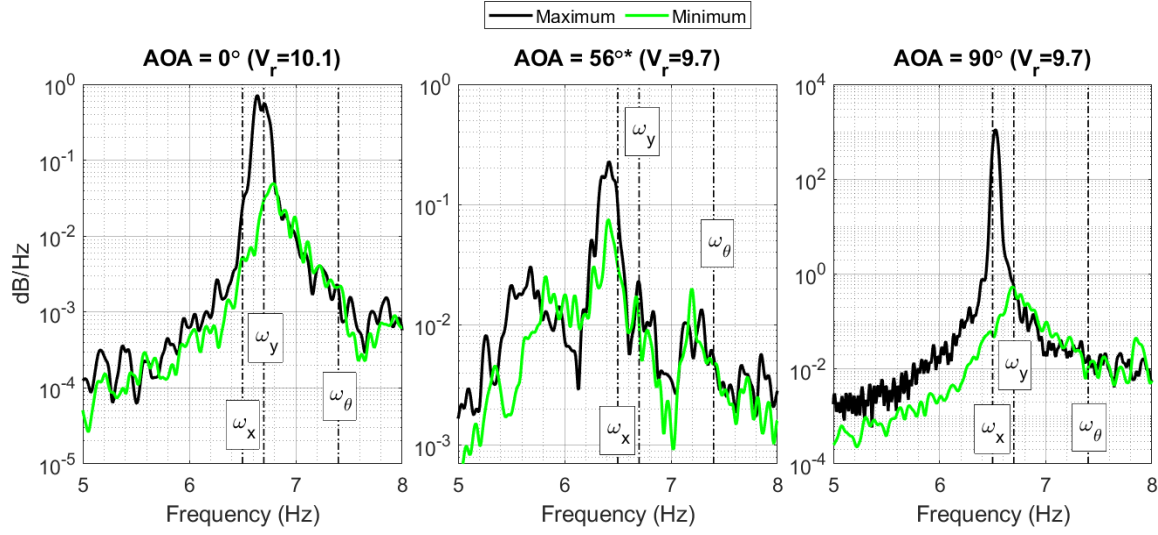


Figure 3.14. Comparison between along-wind acceleration PSDs of best (green) and worst (blue) plate configurations for the three wind directions considered (* case with upstream buildings as illustrated in Figure 3.10)

wind conditions, more advanced predictive techniques for wind condition monitoring are considered in Chapter 4.

3.4. Conclusions

This chapter presented a steady-state controller design that relies on average readings of the wind condition to determine corner plate angles to minimize the building's WIV response. The main component in the controller is a predictor DNN that estimates the building's WIV response for different average wind conditions and plate angle combinations. An optimizer DNN acts as a look up that the controller can use directly during operation. Background genetic algorithm (GA) optimizations are used to train the optimizer DNN. Two different approaches were used to train and evaluate the steady-state controller design: (1) offline GA optimization using the results of fluid-structure-interaction simulations and (2) online iterative optimum

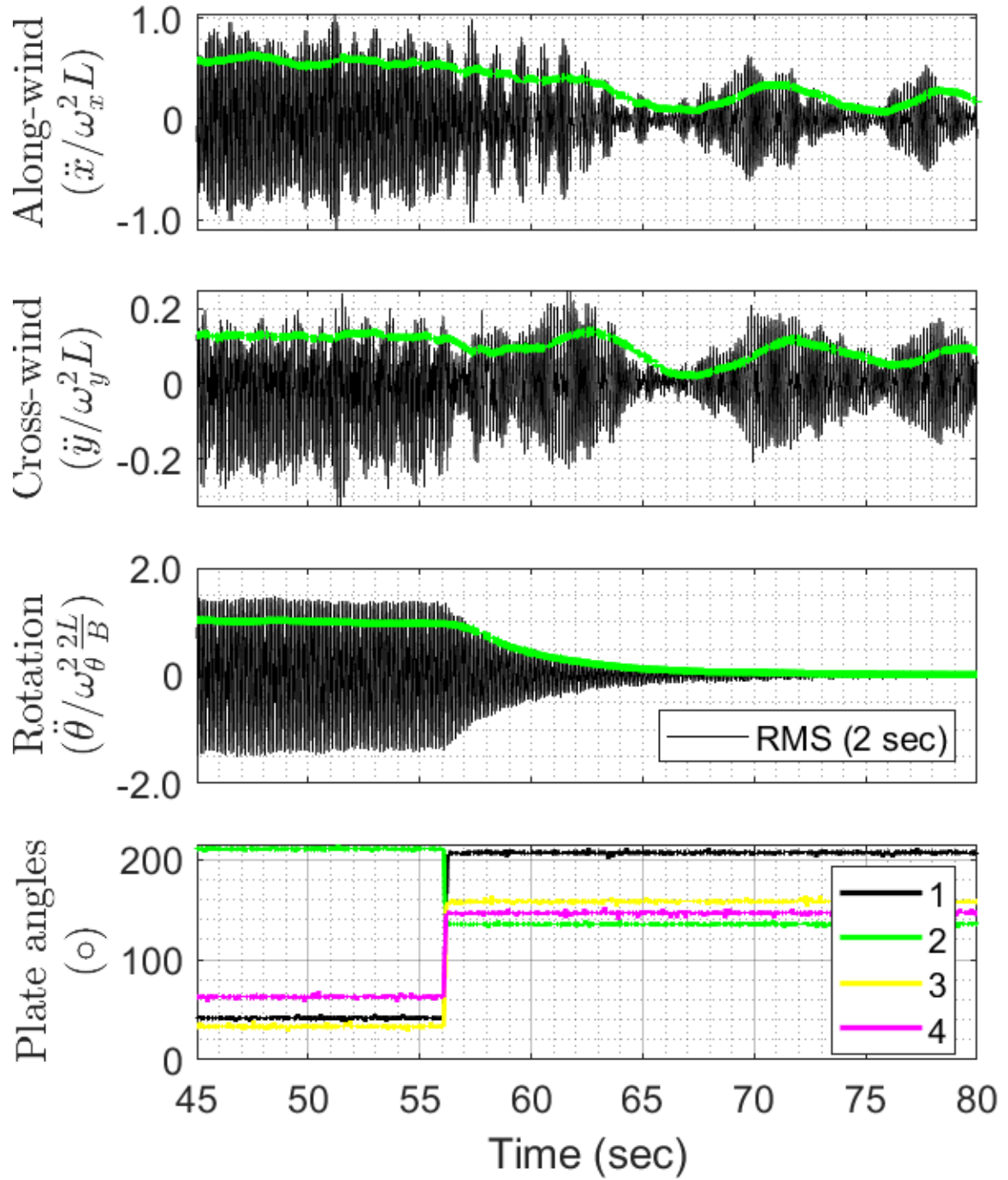


Figure 3.15. Reduction of building accelerations due to changing the plate angles to the optimum values ($AOA = 90^\circ$, $V_r = 9.7$). The IOT-based controller was inactive until $t=56s$ for this demonstration.

training (IOT) using the results of wind tunnel experiments of a scaled robotic aeroelastic model.

The following findings were obtained from the offline GA experiments:

- 1- An amplitude reduction of 60-80% was obtained as a result of the optimization.
- 2- The trained predictor DNN predicted the building deflections with a root-mean-square-error (RMSE) of only ~ 1 mm (7% of maximum deflections).
- 3- The actuation speed of the controller, which includes the speed at which the wind-building system changes, was evaluated using a separate simulation procedure. The controller was found to be able to reduce the oscillation amplitudes by 60% in 3 seconds and by 94% in 7 seconds.

These findings proved the feasibility of the WIV control technique presented and justified the use of wind experimental models. The following findings were obtained from the online IOT procedure using the wind tunnel experimental model:

- 1- A reduction of 41-95% in RMS acceleration was obtained by changing the plate orientations across various wind conditions. The maximum reduction was obtained at $AOA=90^\circ$.
- 2- Depending on the wind condition, some plate orientations maximized the RMS response which justifies the need for actively controlled rather than static aerodynamic modifications.
- 3- In the frequency domain, the RMS reduction was often associated with a diversion of the wind force dominant frequencies away from the structural natural frequencies of the model.

- 4- At 90° and a reduced velocity $V_r = 9.7$, changing from the worst to the best plate angles reduced the RMS rotational acceleration by 33% in as little as 1.1 seconds, which establishes the effectiveness of the system to mitigate wind events at the time scale of the full-scale building.
- 5- The MLP generated from the IOT procedure predicted the RMS acceleration response of the building with an average absolute error of only 0.3% of the maximum RMS acceleration sum.

The findings of this the online IOT training study justify further investigation involving advanced predictive wind condition monitoring techniques to further enhance the practicality of the system.

Chapter 4 - Genetic Reinforcement Learning Control Using GRUs and Distributed Sensor Networks

The steady-state controller discussed in the previous chapter relied on average readings of the wind condition $\bar{w} = [\bar{U}, AOA]$ to choose the corner plate angles to use. In FSI simulations and wind tunnel experiments, this approach was shown to be very effective in reducing the building's WIV response. However, the author noted several concerns that may greatly hinder its adoption in real-life deployments. First, the training process was time consuming and required wind conditions to be held constant in a repeatable manner, which is not feasible for full-scale buildings. Second, potentially destructive wind conditions may need to be encountered by the training process before it can learn how to mitigate their effects. Additionally, defining wind conditions around a building with only two parameters (speed U and direction AOA) ignores many other spatiotemporal features about the wind that may be valuable for effective control. Lastly, while the wind speed U and direction AOA are easily available in simulations and wind tunnel experiments, they are not naturally available to measure for a real-life building. Obtaining free-stream measurement of these quantities may require using advance structures (e.g., gradient towers) that measure the flow before it arrives to the building. The controller developed in this chapter addresses these concerns.

As a better means for defining the wind conditions, this work uses an encoder-decoder GRU model that constructs a unique system state (GRU hidden state) using timeseries readings from multiple sensor systems. The sensors include a distributed network of pressure probes installed on surrounding structures, accelerometers installed on the principal building, and an atmospheric conditions sensor. Overall, the GRU acts as a time-sensitive encoder for the wind-

building system. The use of the sensor systems in conjunction with the GRU model eliminates the need to directly measure the wind speed U and direction AOA while providing a prediction window in which the system can act to prevent incoming hazardous wind events. Finally, the inclusion of the temporal depth of the sensor readings exposes the GRU model to various patterns of system behavior that can be common among different wind conditions (e.g., vortex structures). This is expected to improve the controller's ability to generalize to wind conditions it did not specifically encounter during training.

One of the challenges to the steady-state controller discussed in Chapter 3 was the requirement to hold the wind conditions constant until the system learns how to mitigate the WIV effects, i.e., forcing the environment to be static. The controller developed in this chapter uses reinforcement learning (RL) to program the controller through trial-and-error in the real-life dynamic environment of the full-scale building. As discussed in Section 1.1.6, the exploration aspect of RL algorithms is considerably challenging, which may prevent control agents from quickly obtaining global optimum solutions. In the context of WIV control, this can be a major concern because suboptimal solutions can eventually allow synchronized vortices to form leading to aeroelastic lock-in and large amplitude building vibration.

This work presents a genetic reinforcement learning (GRL) algorithm which enhances RL exploration and enables it to more quickly obtain global optimum solutions. GRL employs genetic algorithms (GAs) integrated with a series of trainable MLP policy functions. The MLP policy functions take the current GRU hidden state as an input and provide the plate orientations dictated by the policy as an output. The GRL algorithm starts by sorting the current solution

(plate orientations) into the set of MLP policies according to how favorable the recorded building accelerations were. The first MLP policy is always trained using the best-found solutions for the encountered GRU hidden states, the second MLP policy is trained using the second-best solutions, and so on. To determine the plate orientations to explore next, the solutions suggested by all MLP policies for the current hidden state are combined to form a population of solutions to perform GA operations on. While GAs inherently provide means for controlling exploration and exploitation, a chance for trying fully random solutions is also added to further enhance this control. As the training progresses, each MLP policy represents a different ‘front’ of solution quality which provides a mechanism for elitism while maintaining solution diversity. GRL can still find better solutions for states it did not exactly encounter because both the GRU and the MLP policies have inherent interpolation ability. During operation, only the best MLP policy (trained from best-found solutions) is used to determine the plate orientations to use for the current GRU hidden state. The developed procedure eliminates the need to keep the uncontrollable wind conditions constant while performing a pseudo-random search capable of obtaining minimum WIV accelerations in one plate rotation action (i.e., reducing the possibility of aeroelastic lock-in).

4.1. Experimental Setup

The small-scale (1:400) robotic aeroelastic model of the CAARC building discussed in Section 2.3 is used to implement the WIV controller in this chapter. Figure 4.1 shows a top view of the scaled aeroelastic model centered in the width of the wind tunnel. The DOFs of interest and the wind direction U are reiterated in this figure, but match those displayed in Figure 2.1. This chapter considers only this direction of wind ($AOA=90^\circ$) speed because it was found to be the most severe in previous experiments (see Section 3.3.4). The same four rotating corner plate

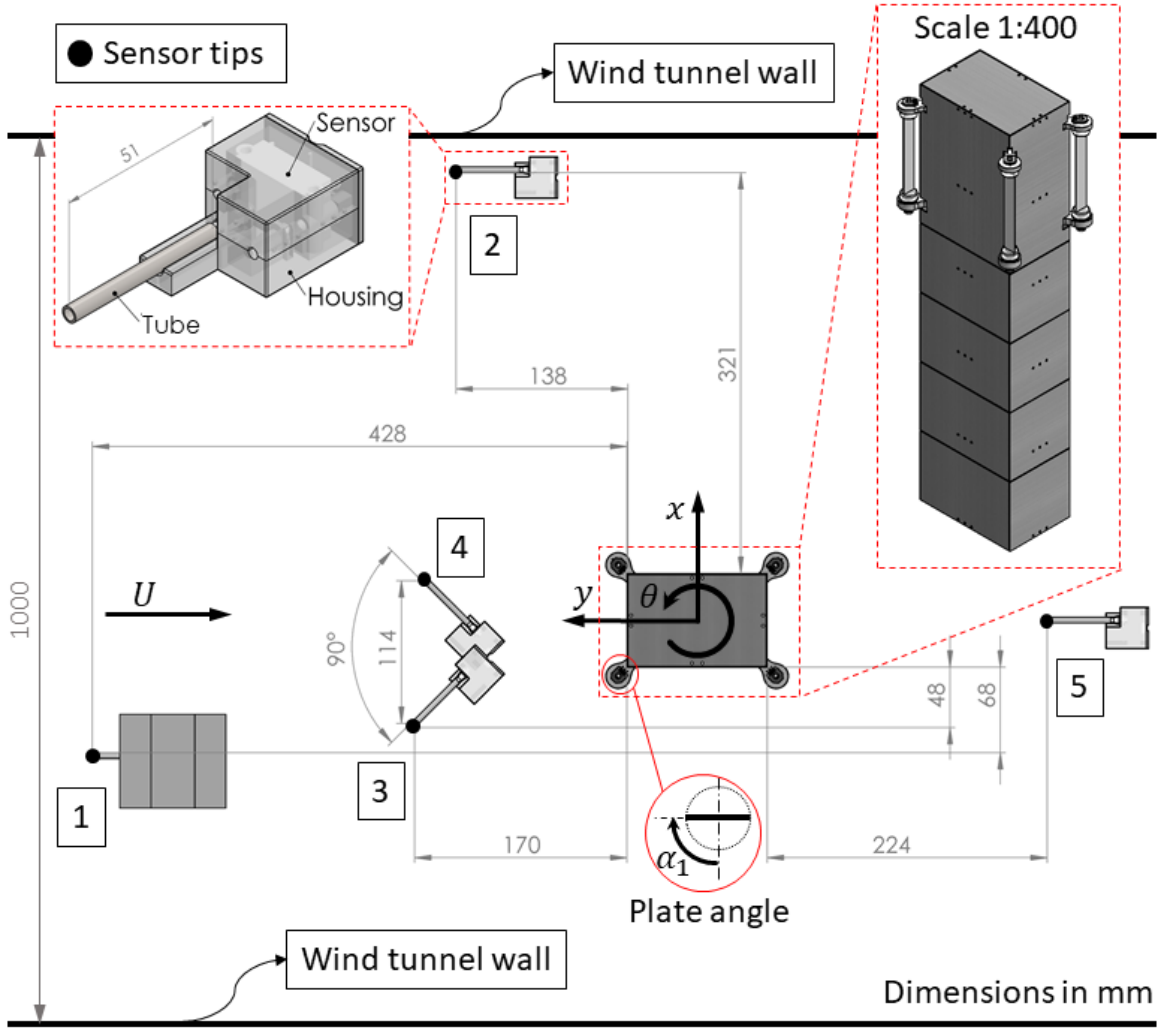


Figure 4.1. A top view of the wind tunnel experiment setup showing: (1) the 1:400 scaled aeroelastic model centered in the width of the tunnel with four rotating plates installed at its corners; (2) the measurement conventions for the DOFs of interest (x , y and θ), plate angles ($\alpha_1 \dots \alpha_4$) and the wind direction U ; and (3) the upstream building and the pressure sensor locations and shape.

configuration is used as an active AM implementation in this chapter. A rigid building model with dimensions $400 \times 85 \times 75$ mm placed upstream of the aeroelastic model is added to the previous experimental setup. This rigid building is included to test the system's ability to handle complex wind flow patterns, such as turbulent wakes and vortex structures.

4.1.1. Sensor network

Figure 4.1 also shows a pressure sensor network consisting of 5 differential pressure sensors that were fitted inside 3d-printed housings with extension tubes on the high-pressure port of the sensors so they can sense the wind flow. The low-pressure port is contained inside the housing to sense the static pressure value. All dimensions are measured relative to the tips of the sensors indicated in Figure 4.1. The wind tunnel floor is used as a reference for all height measurements. Sensor 1 is installed inside the upstream building at a height of 380 mm. Sensor 2 is installed on top of another rigid foam cube at a height of 75 mm. Sensors 3 and 4 are oriented at 45° from the wind direction U and installed on a steel tube extending from the tunnel floor at a height of 200 mm. Sensor 5 is installed on another similar tube at a height of 360 mm. The placement and count of the sensors have a strong effect on the encoder-decoder GRU's ability to fully define the system state. For this initial study, the sensors were placed using human experience. For example, sensor 3 is positioned to endure most of the vortex shedding effect from the upstream rigid building. Though the current placement is shown later to provide acceptable predictions from the encoder-decoder GRU, the optimal placement of the sensors is still being considered for future studies. In addition to the pressure sensors, two accelerometers are installed on the top floor of the scaled building model (see Figure 2.10). Their readings are combined to obtain the accelerations for the 3 DOFs of interest. Finally, an ambient atmospheric conditions sensor is installed outside the wind tunnel to measure temperature, humidity and barometric pressure. As will be discussed in Section 4.2.1, the collection of all sensor readings is used in conjunction with the encoder-decoder GRU to define the system state.

4.1.2. Cyber-physical setup

Figure 4.2 illustrates the cyber-physical setup used to implement the training and control procedure described in Section 4.2. In addition to sensors described in the previous section, four rotational position encoders are installed on the free end of the plates to provide their current position. The plate position readings are not directly used by the encoder-decoder GRU to determine the system state to avoid redundancy with the building accelerations (which partly result from the plate positions). The plate position readings are used by the controller as described in Section 4.2.2 to determine the current plate angles. All sensors and encoders provide analog outputs which are connected to an NI DAQmx® 6463 data acquisition unit. The sampling rate is set at 200 Hz which is determined from the maximum response time of the Sensirion SPD810 differential pressure sensors which were used. The plate motors are controlled by an Arduino® microcontroller. A python script controls the data acquisition through the nidaqmx software library, the wind tunnel speed through analog output and the plate motors through a serial connection to the Arduino microcontroller. The script also implements the online training and control procedures described in the next section. This includes tasks such as timing the experiments, data collection, experiment numbering and online training.

4.2. Controller Design

4.2.1. Encoder-decoder GRU

Figure 4.3 shows the encoder-decoder GRU model. The inputs to the model are the sensor readings $\delta_0 \dots \delta_n$ and the outputs are the future sensor readings $\delta_{n+1} \dots \delta_{n+m}$ for n input and m output time steps, respectively. The model uses the inputs to populate an internal hidden state array S . The sensor readings at time t , δ_t are:

$$\delta_t = [p_0 \ p_1 \ p_2 \ p_3 \ p_4 \ p_b \ T \ H \ \ddot{x} \ \ddot{y} \ \ddot{\theta}]_t \quad (4.1)$$

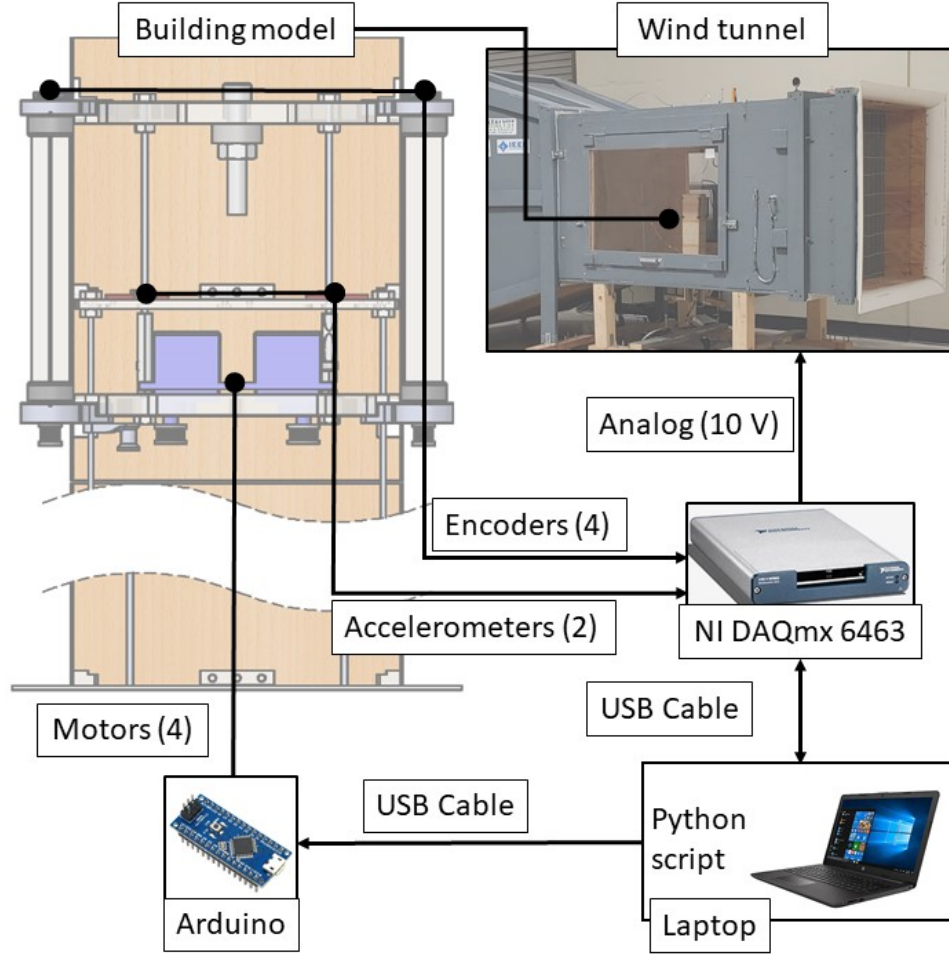


Figure 4.2. A schematic showing the various components of the cyber-physical setup for the wind tunnel experiments. The python script controls all aspects of the online training/control experiment.

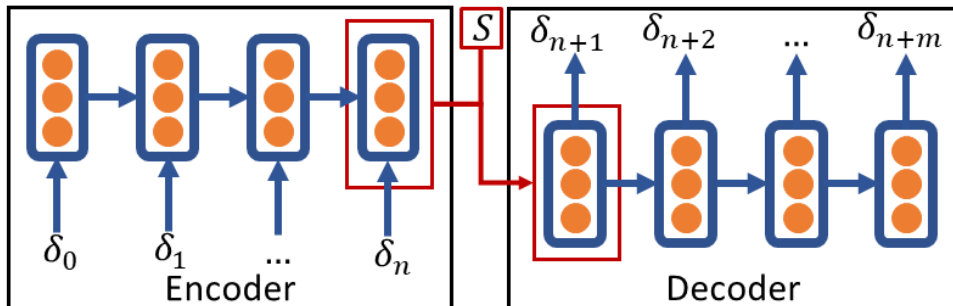


Figure 4.3. A schematic diagram of the encoder-decoder GRU model indicating the sensors inputs $\delta_0 \dots \delta_n$, outputs $\delta_{n+1} \dots \delta_{n+m}$ and the hidden state S

where $p_0 \dots p_4$ are the pressure sensor readings (see Figure 4.1), p_b is the barometric pressure, T is the ambient temperature, H is the humidity, $\ddot{x}$, $\ddot{y}$ and $\ddot{\theta}$ are the acceleration components indicated in Figure 4.1. The encoder portion of the model is stateful, so the hidden state S accumulates its information from all previous inputs as well as the current input. The architecture for the encoder-decoder GRU follows a standard powers-of-2 units for each layer. The encoder has 4 stateful GRU layers with 16, 32, 64 and 256 units respectively. The decoder has the same number of layers, but the unit count is reversed. The number of layers was obtained by trial until an acceptable prediction performance was reached. The encoder-decoder GRU is trained from randomly collected wind tunnel data. The prediction (decoder) portion is not used after initial training. It is only used so the training process can automatically prune the hidden state S to only keep information that is relevant for fully defining the system state.

4.2.2. Genetic reinforcement learning algorithm

The controller developed in this section operates using the same WIV performance function for the steady-state controller as discussed in Section 3.3.1. The only difference is that the GRU hidden state S is used instead of the average wind condition $\bar{w}$ to define the system state:

$$\text{Min. } Q(S, A) = \text{RMS}(\ddot{x}) + \text{RMS}(\ddot{y}) + \frac{B}{2} \text{RMS}(\ddot{\theta}) \quad (4.2)$$

where $A = [\alpha_1 \dots \alpha_4]$ are the plate angles (0-180°). S is the current system state (GRU hidden state). Section 3.3.1 discusses the other terms and lists additional details.

Using this definition of the WIV performance Q , Figure 4.4 illustrates the GRL algorithm that runs after each control interval $[0, t_c]$. At the point of control (indicated by an arrow to the ‘Start’ bubble), the previous plate angles A_p and GRU state S_p are known. Also, the future GRU

state S_f is obtained by feeding the previous sensor information $\delta_0 \dots \delta_{t_c}$ to the GRU. The algorithm is required to do the following: (1) use the previous information to train the MLP policies ($\pi_1, \pi_2, \dots$) and (2) chose which future plate angles A_f to try. The policies have the form $\pi(S) = [A, Q]$, so they not only provide the plate angles dictated by the policy, but also the expected RMS sum when these angles are used. The procedure starts by calculating Equation 4.2 (Q_p) from previous sensor information. Starting from π_1 , the procedure compares the calculated RMS sum Q_p with the policy's prediction $\hat{Q}_1$. If the calculated sum is smaller, the training data is added to the storage for the current policy only and the comparison is not performed for the other policies. The accumulated effect of these comparisons is that the best solutions found (best plate angles) would be stored for π_1 , and solutions that perform worse would be stored for π_2 , and so on. Each of the policies π is trained frequently using the collection of data stored for it. To determine the next plate angles to try A_f , a random number is generated and compared with the probability p_r . If the random number is smaller than p_r , A_f is set to a completely random value to enhance exploration. Otherwise, each of the policies provides optimum angles A_k^* and expected RMS sum $\hat{Q}_k$ for the future GRU state S_f . A single GA iteration is performed on the generated solutions with crossover and mutation probabilities p_c and p_m , respectively. The future plate angles A_f are chosen at random from the GA result. The GA selection (tournament selection) makes it more probable that better solutions would be chosen, so the accumulated effect of the GA iteration is to provide a tendency to try solutions expected to provide better performance for the future state S_f . The probabilities p_r , p_c and p_m provide convenient control of the balance between exploitation and exploration. During operation (after training), $\pi_1(S_f)$ is evaluated directly and A_f is set to A_1^* .

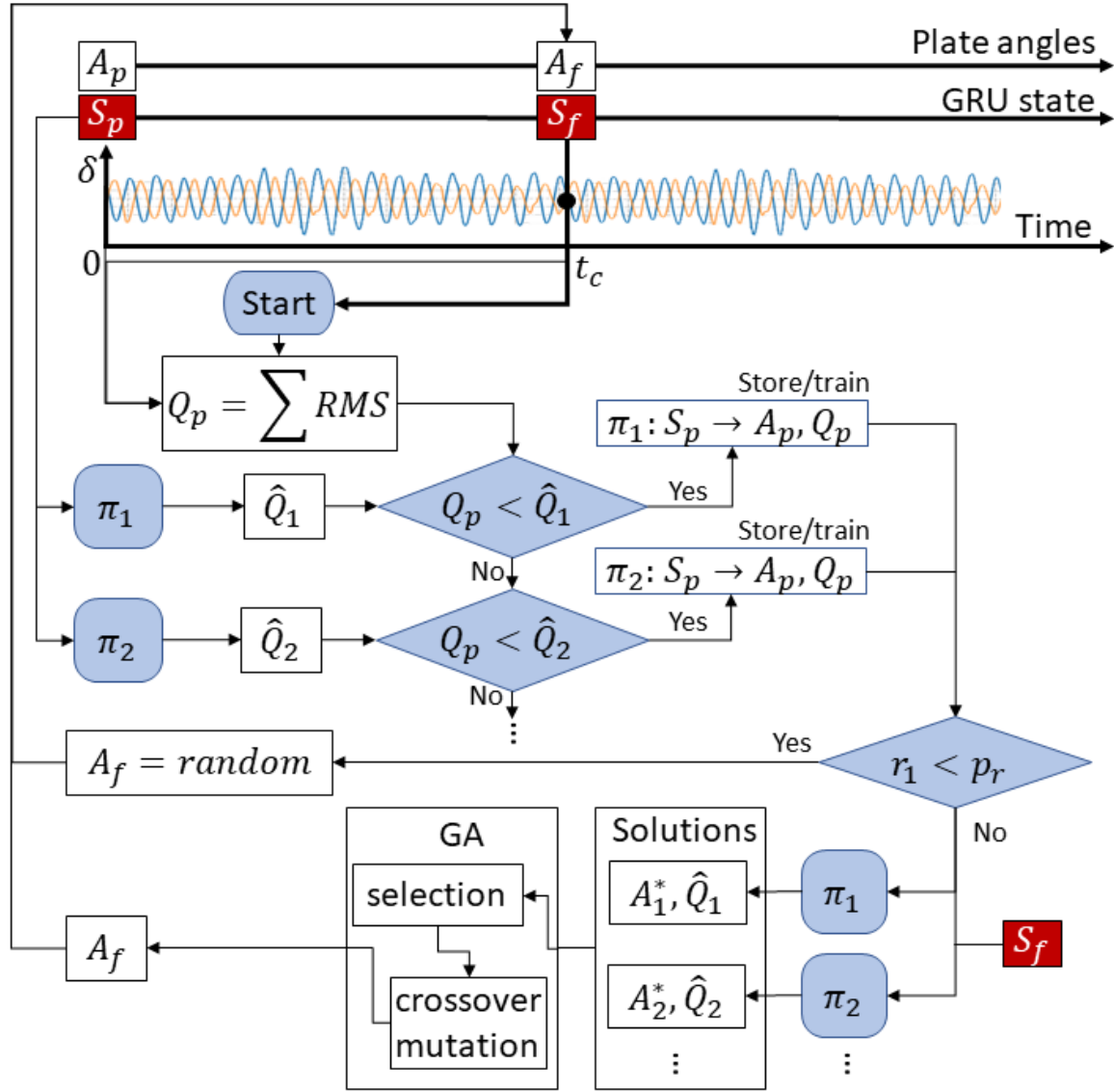


Figure 4.4. The modified RL procedure indicating how the previous plate angles A_p , GRU state S_p and acceleration sum Q_p (Equation 4.2) are used to train the policy MLPs (π_k). The policy MLPs are then used to determine the future plate angles A_f using the future GRU state S_f

4.3. Results and Discussion

4.3.1. Encoder-decoder GRU prediction

The time step for the encoder-decoder GRU model was set to 0.01 s and the sensor readings were down-sampled to match it. The training data is collected from constantly varying random wind conditions and plate orientations. Therefore, increasing the input/output time steps

(prediction time) negatively impacts the prediction performance when the condition variations no longer have precursors in the input data. For this study, the prediction time was increased gradually starting from 1 second until the prediction performance started to diminish. The prediction time reached 7.5 s (750 time steps) without considerable impact to the prediction performance. The time scale with the full-scale building is 1/33, so 7.5 seconds equals approximately 4 minutes for the full-scale building. This gives the control system enough time to respond to most wind events (Abdelaziz et al. 2021). Figure 4.5 compares the predicted and the actual rotational acceleration $\ddot{\theta}$ for a 16-minute data set collected from the wind tunnel. The figure also zooms on a full 7.5 seconds of encoder-decoder output. For the full 16 minutes of prediction, the mean absolute error was only 3.5% of the maximum rotational acceleration. The rotational acceleration $\ddot{\theta}$ is used in this comparison because it was found to be the most significant acceleration component. The comparison confirms the encoder-decoder model's ability to predict and thus fully define the system in its hidden state S .

4.3.2. Controller performance

In reference to Figure 4.4, Table 4.1 lists the parameters used to implement the GRL algorithm for WIV control. The control interval t_c was set by observing the stability of the RMS accelerations at constant wind and plate conditions. When the RMS acceleration becomes constant, it means that the important oscillatory features have already been captured in the considered control length. The control interval is double the GRU input/output time, so the GRU is dispatched two times per control interval to construct the hidden state S . The MLP policy count k was initially set to 12, but the GRL algorithm only managed to store data for 7 out of 12 policies. The process was then repeated using only 7 policy MLPs.

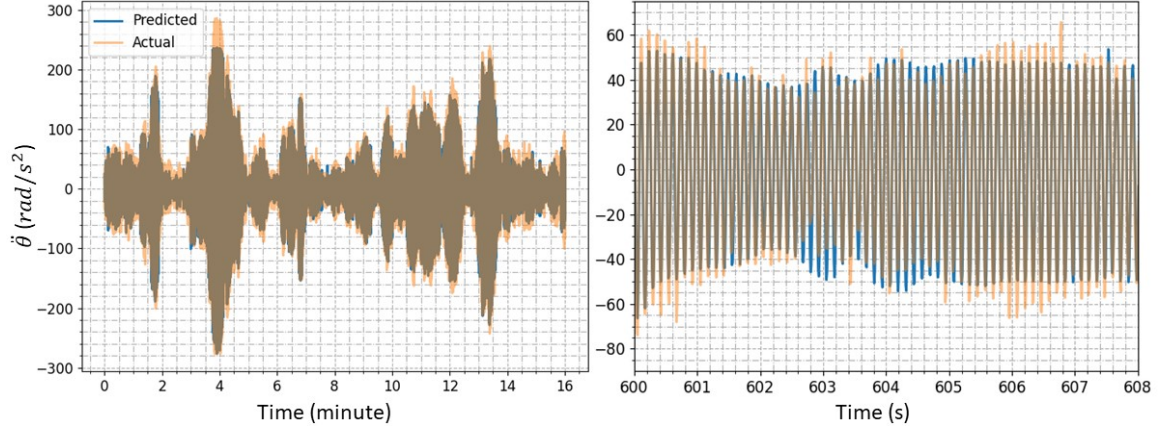


Figure 4.5. A comparison between the predicted and actual building rotational acceleration $\ddot{\theta}$ for 16 minutes and 7.5 seconds of data.

Table 4.1: GRL algorithm parameters for WIV control of the 1:400 scaled building model

Parameter	Value
Control interval t_c	15 s
Policy MLP count k	7
Random probability p_r	0.3
Crossover probability p_c	0.8
Mutation probability p_m	0.2

The random probability p_r was set at 0.3 for this initial study. In future studies, p_r may be set to 1.0 until the policies have enough data points to construct reasonable predictions. The crossover and mutation probabilities p_c and p_m are typical values for GAs.

The procedure illustrated in Figure 4.4 was run on random variations of the wind tunnel speed. The intervals of the speed changes as well as the speed values were randomized. The length of every generated wind speed profile was 10 minutes. Figure 4.6 illustrates the change in the WIV performance Q across subsequent GRL training iterations which are run against different wind speed profiles. To establish a baseline for the controller performance, random plate angle combinations were tested against the same wind speed profiles. For each 10-minute randomly varying wind profile, the plate angles were positioned at a constant (not controlled)

random position. This was done for 90 different combinations of wind profiles and plate angles. The WIV performance Q values for each random case is plotted in Figure 4.6. It is important to note that the minimum Q value attained from these static tests corresponds to a specific combination of plate angles and random wind profile. If a different wind profile was used for this same set of plate angles, the Q value will certainly increase, thus motivating the need for an active (rather than static/passive) façade system. Figure 4.6 also shows that the controller performance indicated by the green training points improves (decreases) across iterations despite being subjected to all random wind speed profiles. The WIV performance Q points for the controller during training does not subject the building to worst-case plate angles, which is an important practical consideration and large improvement from the previous controller design. The controller manages to surpass the performance of the best random plate angles after 5 training iterations. These findings confirm the effectiveness of the GRL algorithm. After the controller was trained for 10 iterations, several aspects of its performance are discussed herein.

Figure 4.7 focuses on two different timeseries locations of interest. In both cases, the wind speed U is changing in a way that causes the acceleration response to increase. This is also evident from the RMS acceleration plot. As soon as the acceleration's control interval t_c passes, the controller detects this increase and repositions the plates according to the trained policy MLP π_1 . As a result, the acceleration response decreases even though the wind speed keeps increasing. This experiment confirms that the controller can anticipate incoming wind events and adjust for them before accelerations become severe. This prevents phenomena such as aeroelastic lock-in from occurring.

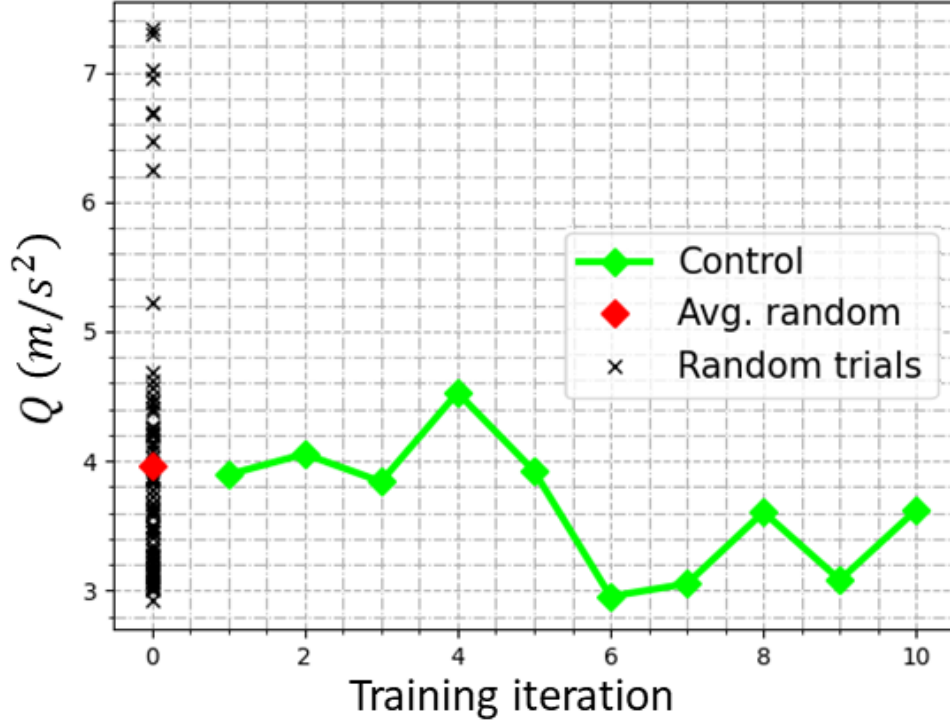


Figure 4.6. Evolution of the controller WIV mitigation performance during 10 iterations of the GRL algorithm subjected to randomly generated wind speed profiles compared to baseline data created with 90 combinations of static (not controlled) plate angles and random wind speed profiles

To further illustrate the WIV mitigation mechanism, Figure 4.8 compares the RMS rotational acceleration $\ddot{\theta}$ with and without using the trained controller. The RMS is calculated using a 20-second window. The wind speed profile is randomly generated but constant for both cases to make a fair comparison. The comparison shows that the controller reduces RMS rotational acceleration by up to 75% and maintains accelerations significantly lower than the uncontrolled case for a vast majority of the time.

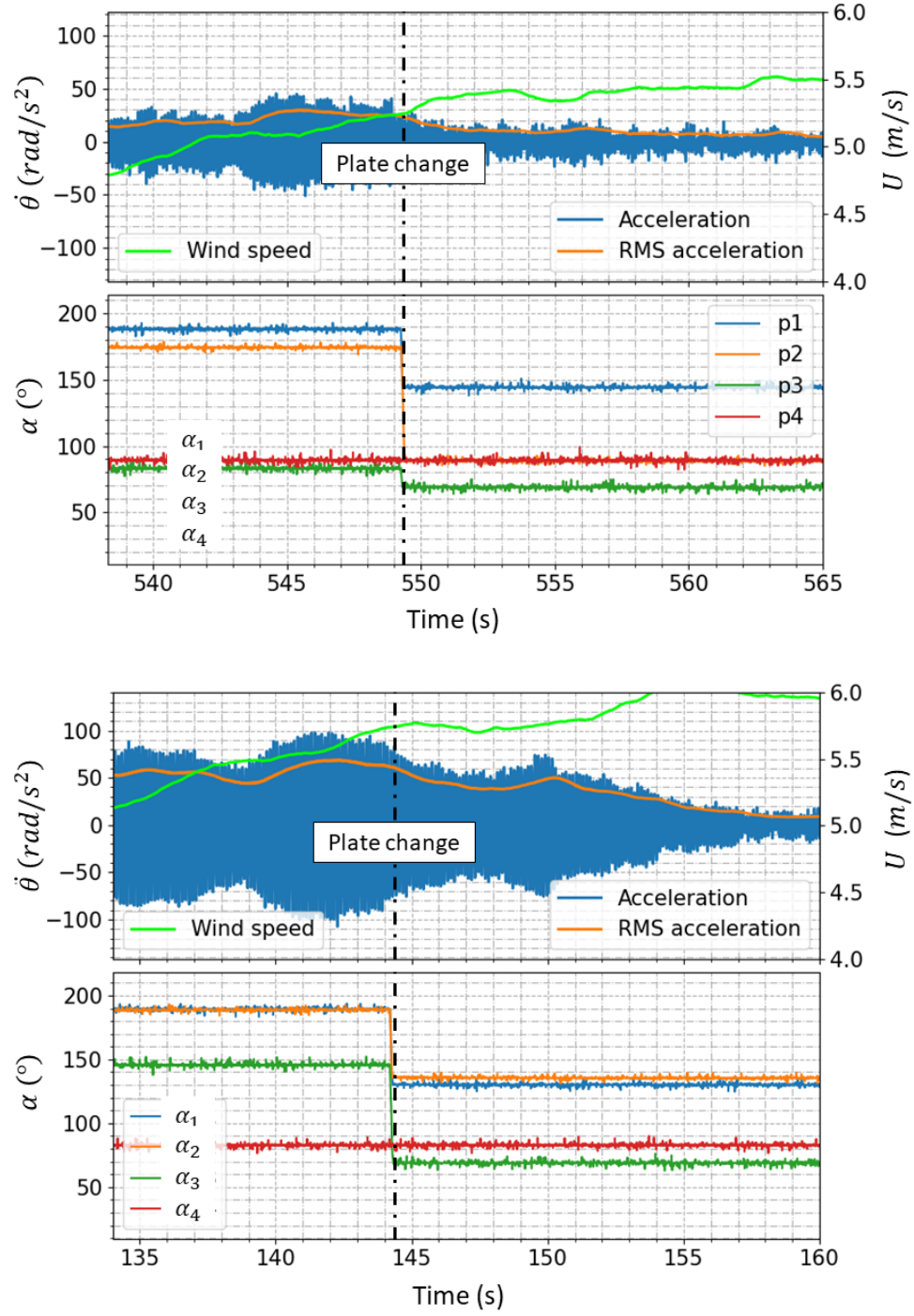


Figure 4.7. Two different cases where the trained controller automatically responded to the increase in the rotational acceleration $\ddot{\theta}$ (resulting from the change in wind speed U) by changing the plate angles which immediately reduced the rotational acceleration

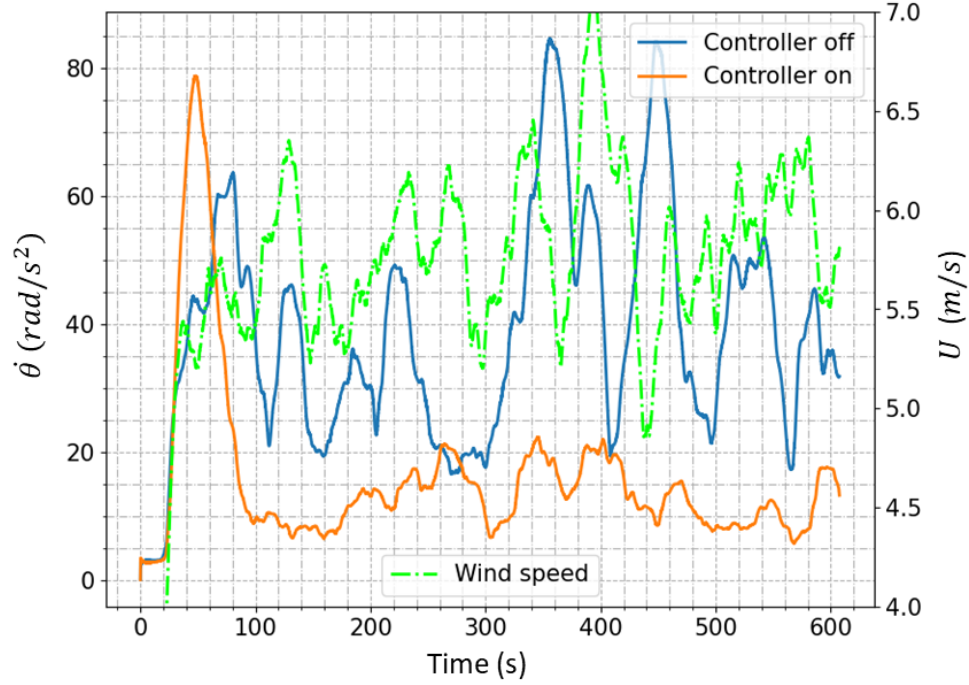


Figure 4.8. A comparison of the RMS rotational acceleration (20-second window) for the same randomly varying wind speed profile when the controller is turned off and on

To get an overall representation of the trained controller performance across many conditions, Figure 4.9 shows the results of testing the trained controller by subjecting it to multiple wind speed profiles and compares the WIV performance with the random baseline. This differs from Figure 4.6, which focuses on the WIV acceleration response during training. It is shown that the controller performance is not strongly affected by changing the wind conditions. This proves that the controller trained using the GRL algorithm can generalize to conditions it did not specifically encounter during training. This is an important practical consideration because the real-life conditions are infinite and training for all of these conditions is not feasible.

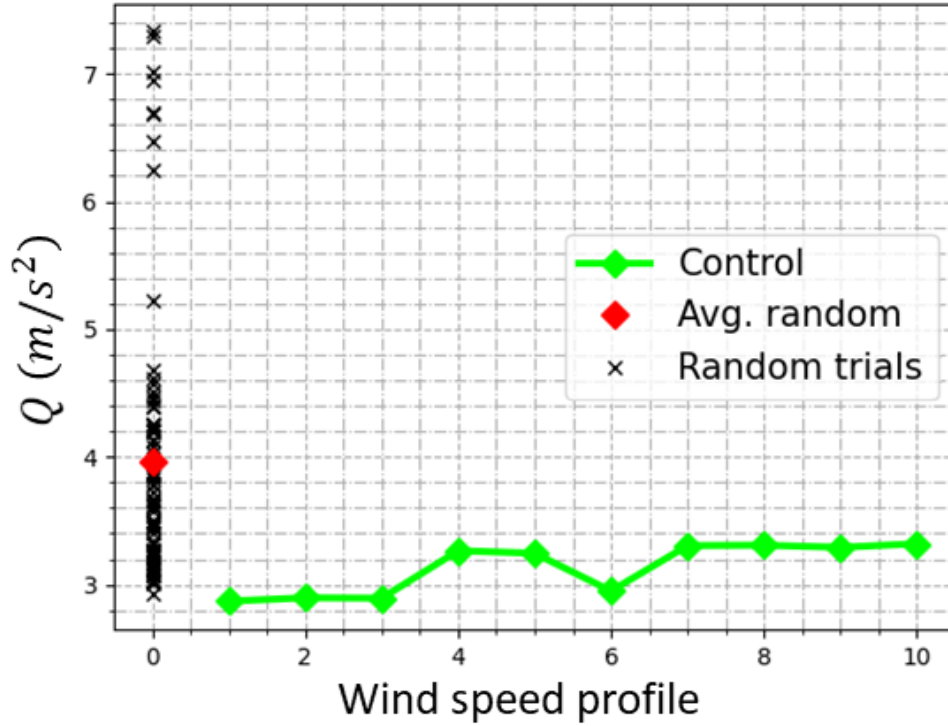


Figure 4.9. The effect of changing the wind speed profile on the controller performance Q compared with the performance of the evaluated random plate angles

4.4. Conclusions

This chapter presents a WIV controller that addresses most of the practical concerns for the steady-state controller developed in the previous chapter. The new controller relies on genetic reinforcement learning (GRL) integrated with a distributed sensor network and a gated recurrent unit neural network (GRU). The system uses four plates that are able to rotate about their axes of symmetry to function as aerodynamic control surfaces. A technique that enables defining fast varying and complex wind conditions is presented. A sensor network consisting of a set of pressure probes, accelerometers and ambient conditions sensor are used as time series inputs to an encoder-decoder GRU model. The model uses the inputs to define an instantaneous hidden state that fully defines the wind-building system state and ‘remembers’ long-term effects relevant for predicting future sensor readings. A genetic reinforcement learning procedure is developed to

use the encoder-decoder hidden state to perform pseudo-random optimization in a highly dynamic environment. The process used a series of trainable policy MLP functions as ‘agents’ that can provide a genetic algorithm with a set of diverse solutions. The control system was tested on a 1:400 scale robotic aeroelastic building model in wind tunnel experiments. The important findings of this work are summarized below:

1. The encoder-decoder GRU model was trained from wind tunnel experiment data and was able to predict the future rotational acceleration from the current rotational acceleration with an average absolute error that is only 3.5% of the maximum encountered acceleration. This confirms that the encoder-decoder GRU hidden state fully defines the wind-building system.
2. The GRL algorithm managed to considerably improve the controller performance within only five 10-minute training iterations. During training, the algorithm did not subject the building to severe catastrophic conditions such as encountered by some of the baseline random plate angles.
3. The trained controller reduced the RMS rotational acceleration by up to 75% for the same test wind speed profile.
4. The trained controller was able to detect adverse wind conditions before the peak wind speed occurred. So, it can anticipate severe conditions and adjust for them before they happen.
5. The controller performance was not affected considerably when tested against a group of random wind speed profiles, which confirms its ability to generalize to conditions it was not specifically trained for.

In future work, a sensitivity analysis of the different controller parameters should be carried out to establish guidelines for setting these parameters. This includes the number of iterations to train the controller for, among others. Comparison with other reinforcement learning techniques and variations in the literature is also an area of potential future work.

Chapter 5 - Conclusions and Recommendations

This work presented the development and implementation of intelligent control systems for WIV mitigation in tall buildings. Two main modeling approaches were used to develop WIV minimization controllers that use active aerodynamic modification (AM) surfaces: (1) Fluid-structure-interaction (FSI) simulations and (2) wind tunnel experiments of scaled robotic aeroelastic building models. The Commonwealth Aeronautical Advisor Research Committee (CAARC) standard building was used as a base case for both approaches. The FSI simulation model was designed to enable the rotation of the active AM plates without requiring reconstruction of the mesh. This was achieved using interpolation interfaces around the plate rotation zones. The initial results from the FSI simulation showed the great effect of the active AM plate orientations on the building's WIV response. This finding constitutes a cornerstone to the controller development.

The second approach was to employ small- and large-scale wind tunnel experiments of robotic aeroelastic building models. The small-scale (1:400) building model was designed using a genetic algorithm (GA) procedure that matches its dynamic properties to the full-scale building. The characterization of the resulting manufactured model showed minor deviations from the design values which proves the model's validity for conducting wind tunnel experiments.

The large-scale model (1:73) allowed for the inclusion of 9 lumped masses, instead of 4 for the small-scale model. This is expected to enable the model to simulate higher modes of oscillation. The increase in available construction space also allowed the model to accommodate

different active AM designs and additional AM control surfaces, both of which are expected to better demonstrate the effectiveness of the developed controllers. As of the time of this writing, the large-scale model is being prepared for shipping to the Natural Hazards Engineering Research Infrastructure (NHERI) Wall-of-Wind (WoW) facility for testing. The results of these tests will be submitted for publication as soon as they are completed.

The aforementioned modeling techniques were used to develop a steady-state controller design that relies on average readings of the wind condition to determine corner plate angles to minimize the building's WIV response. The main component in the controller is a predictor DNN that estimates the building's WIV response for different average wind conditions and plate angle combinations. An optimizer DNN acts as a look up that the controller can use directly during operation. Background genetic algorithm (GA) optimizations are used to train the optimizer DNN. Two different approaches were used to train and evaluate the steady-state controller design: (1) offline GA optimization using the results of fluid-structure-interaction simulations and (2) online iterative optimum training (IOT) using the results of wind tunnel experiments of a scaled robotic aeroelastic model. The following findings were obtained from the offline GA experiments:

- 1- An amplitude reduction of 60-80% was obtained as a result of the optimization.
- 2- The trained predictor DNN predicted the building deflections with a root-mean-square-error (RMSE) of only ~1 mm (7% of maximum deflections).
- 3- . The actuation speed of the controller, which includes the speed at which the wind-building system changes, was evaluated using a separate simulation procedure. The

controller was found to be able to reduce the oscillation amplitudes by 60% in 3 seconds and by 94% in 7 seconds.

These findings proved the feasibility of the WIV control technique presented and justified the use of wind experimental models. The following findings were obtained from the online IOT procedure using the wind tunnel experimental model:

- 1- A reduction of 41-95% in RMS acceleration was obtained by changing the plate orientations across various wind conditions. The maximum reduction was obtained at $AOA=90^\circ$.
- 2- Depending on the wind condition, some plate orientations maximized the RMS response which justifies the need for actively controlled rather than static aerodynamic modifications.
- 3- In the frequency domain, the RMS reduction was often associated with a diversion of the wind force dominant frequencies away from the structural natural frequencies of the model.
- 4- At 90° and a reduced velocity $V_r = 9.7$, changing from the worst to the best plate angles reduced the RMS rotational acceleration by 33% in as little as 1.1 seconds, which establishes the effectiveness of the system to mitigate wind events at the time scale of the full-scale building.
- 5- The MLP generated from the IOT procedure predicted the RMS acceleration response of the building with an average absolute error of only 0.3% of the maximum RMS acceleration sum.

Several practical considerations motivated the development of an alternate controller design that addresses most of the practical concerns for the steady-state controller. The new controller relies on genetic reinforcement learning (GRL) integrated with a distributed sensor network and a gated recurrent unit neural network (GRU). The system uses four plates that are able to rotate about their axes of symmetry to function as aerodynamic control surfaces. A technique that enables defining fast varying and complex wind conditions is presented. A sensor network consisting of a set of pressure probes, accelerometers and ambient conditions sensor are used as time series inputs to an encoder-decoder GRU model. The model uses the inputs to define an instantaneous hidden state that fully defines the wind-building system state and ‘remembers’ long-term effects relevant for predicting future sensor readings. A genetic reinforcement learning procedure is developed to use the encoder-decoder hidden state to perform pseudo-random optimization in a highly dynamic environment. The process used a series of trainable policy MLP functions as ‘agents’ that can provide a genetic algorithm with a set of diverse solutions. The control system was tested on a 1:400 scale robotic aeroelastic building model in wind tunnel experiments. The important findings of this work are summarized below:

1. The encoder-decoder GRU model was trained from wind tunnel experiment data and was able to predict the future rotational acceleration from the current rotational acceleration with an average absolute error that is only 3.5% of the maximum encountered acceleration. This confirms that the encoder-decoder GRU hidden state fully defines the wind-building system.
2. The GRL algorithm managed to considerably improve the controller performance within only five 10-minute training iterations. During training, the algorithm did not subject the

building to severe catastrophic conditions such as encountered by some of the baseline random plate angles.

3. The trained controller reduced the RMS rotational acceleration by up to 75% for the same test wind speed profile.
4. The trained controller was able to detect adverse wind conditions before the peak wind speed occurred. So, it can anticipate severe conditions and adjust for them before they happen.
5. The controller performance was not affected considerably when tested against a group of random wind speed profiles, which confirms its ability to generalize to conditions it was not specifically trained for.

5.1. Recommendations for Future Work

Multiple research approaches were involved in this multidisciplinary study. This section discusses multiple ideas for future research that may either address weaknesses in or enhance the developed system.

5.1.1. Economic analysis

The total investment cost of tall buildings is the main motivator behind this research. Currently, tall buildings are built to withstand severe wind conditions by including additional materials to increase the stiffness. The design process for WIV resistance is also currently complex, because it involves wind tunnel experiments of scaled models. Both factors considerably increase the total investment cost. The controller developed in Chapter 4 - Chapter 4 requires the least amount of training and designer involvement because it is generalized to handle arbitrary building designs and active aerodynamic modification (AM) design.

An economic analysis for the final system, including all necessary components of the active AM implementation, is necessary to prove that the system will result in a reduced total investment cost relative to adding more materials and conducting wind tunnel experiments, or relative to including a tuned mass/liquid damper only. If favorable, the results of this economic analysis will convince designers to adopt the developed system.

5.1.2. Alternate AM designs

While the effectiveness and practicality of the four-corner-plate active AM design was proven in this work, thorough investigation of other designs was not accomplished. Other designs may prove to be more effective or practical. The economic analysis discussed in the previous section may also help to determine which final design to employ.

Even for the same vertical plates design, adding additional vertical plates around the perimeter may improve the effectiveness of the system. This point was highlighted in Section 3.3.4.2 where the wind direction caused one of the corner plates to be shadowed by the building geometry. In that case, adding one plate in the center of each edge may be especially beneficial.

5.1.3. Genetic reinforcement learning (GRL)

Reinforcement learning (RL) is widely discussed in the literature. GRL was used herein as a modification to RL that enhances the ability to find global optimum solutions, which reduces the total training time. The independent effectiveness of GRL was established in Chapter 4. For future research, comparison with other RL algorithms, possibly using benchmark problems, can be very beneficial in proving the benefits of GRL, which can lead to adoption in other applications.

Additionally, a thorough analysis of the step-by-step progress of GRL can benefit further improvement and establish superiority to other approaches. This analysis can only be performed using a benchmark problem that does not have the timing constraints of the wind tunnel experiments.

5.1.4. Full-scale training

Both small scale and large-scale wind tunnel models were performed in this study. Full-scale training, however, would eliminate the need to perform wind-tunnel testing altogether. This was one of the design goals for the GRL-based controller, which was trained from randomly varying wind speed profiles to mimic fast-varying wind in real-life. Alternatively, models pre-trained using wind tunnel experiments for different building designs can be used to initialize the full-scale training process for the current design. Full-scale training can greatly increase adoption of the developed system.

5.1.5. Integration with other WIV mitigation methods

Integration with TMD/TLD systems and other means of WIV mitigation can be beneficial from the perspective of total investment cost. While the effectiveness of active AMs in reducing WIV effects is established in this study, tall buildings are also prone to earthquake vibration damage and failure. Therefore, integration with TMD/TLD systems may be necessary for locations where earthquakes are expected to happen.

References

- Abdelaziz K.M., Alipour A. and Hobeck J.D., 2021. A smart façade system controller for optimized wind-induced vibration mitigation in tall buildings. *Journal of Wind Engineering and Industrial Aerodynamics* 212 , 104601. DOI: <https://doi.org/10.1016/j.jweia.2021.104601>
- Abdelaziz K.M., Hamza K., El-Morsi M., Nassef A.O., Metwalli S.M. and Saitou K., 2016. Optimum Solar Humidification–Dehumidification Desalination for Microgrids and Remote Area Communities. *Journal of Solar Energy Engineering* 138 , DOI: 10.1115/1.4032477
- Abdelaziz K.M. and Hobeck J.D., 2019. Developing a Smart Façade System Controller for Wind-Induced Vibration Mitigation in Tall Buildings ASME 2019 Conference on Smart Materials, Adaptive Structures and Intelligent Systems. DOI: 10.1115/smasis2019-5674
- Abdelaziz K.M. and Hobeck J.D., 2022. Optimum aeroelastic control via iterative neural network training for wind-resistant cyber–physical buildings. *Applied Soft Computing* 114 , 108100. DOI: <https://doi.org/10.1016/j.asoc.2021.108100>
- Amari S.-i., 1993. Backpropagation and stochastic gradient descent method. *Neurocomputing* 5 (4–5), 185–196. DOI: 10.1016/0925-2312(93)90006-o
- American Meteorological Society (2020), [Online], Available: www.ametsoc.org.
- Angelova M. and Pencheva T., 2011. Tuning Genetic Algorithm Parameters to Improve Convergence Time. *International Journal of Chemical Engineering* 2011 , 1–7. DOI: 10.1155/2011/646917
- Antonini E.G.A., Romero D.A. and Amon C.H., 2019. Improving CFD wind farm simulations incorporating wind direction uncertainty. *Renewable Energy* 133 , 1011-1023. DOI: <https://doi.org/10.1016/j.renene.2018.10.084>
- Ayachit U. (2015) *The ParaView Guide (Full Color Version): A Parallel Visualization Application*, Kitware, Incorporated.
- Banks J.W., Henshaw W.D., Schwendeman D.W. and Tang Q., 2018. A stable partitioned FSI algorithm for rigid bodies and incompressible flow in three dimensions. *Journal of Computational Physics* 373 , 455–492. DOI: 10.1016/j.jcp.2018.06.072
- Besem F.M., Thomas J.P., Kielb R.E. and Dowell E.H., 2016. An aeroelastic model for vortex-induced vibrating cylinders subject to frequency lock-in. *Journal of Fluids and Structures* 61 , 42-59. DOI: <https://doi.org/10.1016/j.jfluidstructs.2015.10.009>
- Cho K., van Merriënboer B., Bahdanau D. and Bengio Y., 2014. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. *CoRR* abs/1409.1259 ,

- Clerc M., 1999. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406). DOI: 10.1109/cec.1999.785513
- Clevert D.-A., Unterthiner T. and Hochreiter S. (2016) *Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)*.
- Conde-Gutiérrez R.A., Cruz-Jacobo U., Huicochea A., Casolco S.R. and Hernández J.A., 2018. Optimal multivariable conditions in the operation of an absorption heat transformer with energy recycling solved by the genetic algorithm in artificial neural network inverse. *Applied Soft Computing* 72 , 218–234. DOI: 10.1016/j.asoc.2018.08.008
- Conti E., Madhavan V., Such F.P., Lehman J., Stanley K.O. and Clune J. (2018) *Improving Exploration in Evolution Strategies for Deep Reinforcement Learning via a Population of Novelty-Seeking Agents*.
- Courant R., Friedrichs K. and Lewy H., 1928. Über die partiellen Differenzengleichungen der mathematischen Physik. *Mathematische Annalen* 100 (1), 32–74. DOI: 10.1007/bf01448839
- Dasgupta J., Sikder J. and Mandal D., 2017. Modeling and optimization of polymer enhanced ultrafiltration using hybrid neural-genetic algorithm based evolutionary approach. *Applied Soft Computing* 55 , 108–126. DOI: 10.1016/j.asoc.2017.02.002
- Fagley C.P., Broadbent D., Seidel J. and McLaughlin T.E., 2017. Stall Flutter Prediction and Experimental Verification using a Cyber-Physical Wing 55th AIAA Aerospace Sciences Meeting. DOI: 10.2514/6.2017-1416
- Fangwei H., 2020. Novel Methods for Predicting and Controlling the Wind-Induced Buffeting Response of Tall Buildings in Time Domain. Iowa State University,
- Fernández-Cabán P.L., Whiteman M.L., Phillips B.M., Masters F.J., Davis J.R. and Bridge J.A., 2020. Cyber-physical design and optimization of tall building dynamics using aeroelastic wind tunnel modeling. *Journal of Wind Engineering and Industrial Aerodynamics* 198 , 104092. DOI: 10.1016/j.jweia.2020.104092
- Frison G., Marra A.M., Bartoli G. and Scotta R. (2019) Full-Aeroelastic Model of CAARC Building: Iterative Design Procedure and Wind Tunnel Tests, in *Lecture Notes in Civil Engineering*, Springer International Publishing, Available: https://doi.org/10.1007/978-3-030-12815-9_25.
- Gardarsson S., Yeh H. and Reed D., 2001. Behavior of Sloped-Bottom Tuned Liquid Dampers. *Journal of Engineering Mechanics* 127 (3), 266-271. DOI: 10.1061/(ASCE)0733-9399(2001)127:3(266)
- Haarnoja T., Ha S., Zhou A., Tan J., Tucker G. and Levine S. (2019) *Learning to Walk via Deep Reinforcement Learning*.

- Haftka R.T., Villanueva D. and Chaudhuri A., 2016. Parallel surrogate-assisted global optimization with expensive functions – a survey. *Structural and Multidisciplinary Optimization* 54 , 3–13. DOI: 10.1007/s00158-016-1432-3
- Hobeck J.D. and Inman D.J., 2012. Artificial piezoelectric grass for energy harvesting from turbulence-induced vibration. *Smart Materials and Structures* 21 (10), 105024. DOI: 10.1088/0964-1726/21/10/105024
- Hobeck J.D. and Inman D.J., 2015. The J_d number: An empirical constant for predicting dual cantilever flutter velocity. *Applied Physics Letters* 106 , 244103. DOI: 10.1063/1.4922876
- Hobeck J.D. and Inman D.J., 2016. Dual cantilever flutter: Experimentally validated lumped parameter modeling and numerical characterization. *Journal of Fluids and Structures* 61 , 324–338. DOI: 10.1016/j.jfluidstructs.2015.11.018
- Holzmann T. (2016) *Mathematics, Numerics, Derivations and OpenFOAM®*, Holzmann CFD, Available: <http://rgdoi.net/10.13140/RG.2.2.27193.36960>.
- Hong Z.-W., Shann T.-Y., Su S.-Y., Chang Y.-H. and Lee C.-Y. (2018) *Diversity-Driven Exploration Strategy for Deep Reinforcement Learning*.
- Hou F. and Sarkar P.P., 2020. Aeroelastic model tests to study tall building vibration in boundary-layer and tornado winds. *Engineering Structures* 207 , 110259. DOI: 10.1016/j.engstruct.2020.110259
- Householder A.S., 1958. Unitary Triangularization of a Nonsymmetric Matrix. *Journal of the ACM* 5 (4), 339–342. DOI: 10.1145/320941.320947
- Huang H.-X., Li J.-C. and Xiao C.-L., 2015. A proposed iteration optimization approach integrating backpropagation neural network with genetic algorithm. *Expert Systems with Applications* 42 , 146–155. DOI: 10.1016/j.eswa.2014.07.039
- Hu G., Hassanli S., Kwok K.C.S. and Tse K.T., 2017. Wind-induced responses of a tall building with a double-skin façade system. *Journal of Wind Engineering and Industrial Aerodynamics* 168 , 91–100. DOI: 10.1016/j.jweia.2017.05.008
- Hurricane Science* (2020), [Online], Available: www.hurricanescience.org.
- Inman D.J. (2013) *Engineering Vibration (4th Edition)*, Pearson, Available: <https://www.amazon.com/Engineering-Vibration-4th-Daniel-Inman/dp/0132871696?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0132871696>.
- Jaderberg M., Czarnecki W.M., Dunning I., Marris L., Lever G., Castañeda A.G., Beattie C., Rabinowitz N.C., Morcos A.S., Ruderman A., Sonnerat N., Green T., Deason L., Leibo J.Z., Silver D., Hassabis D., Kavukcuoglu K. and Graepel T., 2019. Human-level

- performance in 3D multiplayer games with population-based reinforcement learning. *Science* 364 , 859-865. DOI: 10.1126/science.aau6249
- Jafari M. and Alipour A., 2021. Methodologies to Mitigate Wind-induced Vibration of Tall Buildings: A State-of-the-art Review. *Journal of Building Engineering* 33 , DOI: 10.1016/j.jobbe.2020.101582
- Jamil M. and Yang X.S., 2013. A literature survey of benchmark functions for global optimisation problems. *International Journal of Mathematical Modelling and Numerical Optimisation* 4 (2), 150. DOI: 10.1504/ijmmno.2013.055204
- Jasak H., 2009. Dynamic Mesh Handling in OpenFOAM 47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition. DOI: 10.2514/6.2009-341
- Kingma D.P. and Ba J. (2017) *Adam: A Method for Stochastic Optimization*.
- Kiran B.R., Sobh I., Talpaert V., Mannion P., Sallab A.A.A., Yogamani S. and Pérez P., 2021. Deep Reinforcement Learning for Autonomous Driving: A Survey. *IEEE Transactions on Intelligent Transportation Systems*, 1-18. DOI: 10.1109/TITS.2021.3054625
- Kong Z., Tang B., Deng L., Liu W. and Han Y., 2020. Condition monitoring of wind turbines based on spatio-temporal fusion of SCADA data by convolutional neural networks and gated recurrent units. *Renewable Energy* 146 , 760-768. DOI: <https://doi.org/10.1016/j.renene.2019.07.033>
- Lin C., Wang H., Fu M., Yuan J. and Gu J., 2021. A Gated Recurrent Unit-Based Particle Filter for Unmanned Underwater Vehicle State Estimation. *IEEE Transactions on Instrumentation and Measurement* 70 , 1-12. DOI: 10.1109/TIM.2020.3011789
- Li D., Ronch A.D., Chen G. and Li Y., 2019. Aeroelastic global structural optimization using an efficient CFD-based reduced order model. *Aerospace Science and Technology* 94 , 105354. DOI: 10.1016/j.ast.2019.105354
- Love J.S., Haskett T.C. and Morava B., 2018. Effectiveness of dynamic vibration absorbers implemented in tall buildings. *Engineering Structures* 176 , 776-784. DOI: <https://doi.org/10.1016/j.engstruct.2018.09.050>
- Márquez-Nolasco A., Conde-Gutiérrez R.A., Hernández J.A., Huicochea A., Siqueiros J. and Pérez O.R., 2017. Optimization and Estimation of the Thermal Energy of an Absorber With Graphite Disks by Using Direct and Inverse Neural Network. *Journal of Energy Resources Technology* 140 , DOI: 10.1115/1.4036544
- Micheli L., Alipour A. and Laflamme S., 2018. Performance-Based Design for Wind-Excited Tall Buildings Equipped with High Performance Control Systems. DOI: 10.1061/9780784481325.019

- Miranda R., Babilio E., Singh N., Santos F. and Fraternali F., 2020. Mechanics of smart origami sunscreens with energy harvesting ability. *Mechanics Research Communications* 105 , 103503. DOI: 10.1016/j.mechrescom.2020.103503
- Momtaz A.A., Abdollahian M.A. and Farshidianfar A., 2017. Study of wind-induced vibrations in tall buildings with tuned mass dampers taking into account vortices effects. *International Journal of Advanced Structural Engineering* 9 (4), 385–395. DOI: 10.1007/s40091-017-0174-9
- Nagy Z., Svetozarevic B., Jayathissa P., Begle M., Hofer J., Lydon G., Willmann A. and Schlueter A., 2016. The Adaptive Solar Facade: From concept to prototypes. *Frontiers of Architectural Research* 5 (2), 143–156. DOI: 10.1016/j.foar.2016.03.002
- Noriega J.R. and Wang H., 1998. A direct adaptive neural-network control for unknown nonlinear systems and its application. *IEEE Transactions on Neural Networks* 9 , 27–34. DOI: 10.1109/72.655026
- Okajima A., 1982. Strouhal numbers of rectangular cylinders. *Journal of Fluid Mechanics* 123 , 379–398. DOI: 10.1017/s0022112082003115
- Parkinson G., 1989. Phenomena and modelling of flow-induced vibrations of bluff bodies. *Progress in Aerospace Sciences* 26 , 169–224. DOI: 10.1016/0376-0421(89)90008-0
- Paulavičius R., Sergeyev Y.D., Kvasov D.E. and Žilinskas J., 2020. Globally-biased BIRECT algorithm with local accelerators for expensive global optimization. *Expert Systems with Applications* 144 , 113052. DOI: 10.1016/j.eswa.2019.113052
- Pryor S.C., Barthelmie R.J., Young D.T., Takle E.S., Arritt R.W., Flory D., Gutowski W.J., Nunes A. and Roads J., 2009. Wind speed trends over the contiguous United States. *Journal of Geophysical Research* 114 (D14), DOI: 10.1029/2008jd011416
- Regis R.G. and Shoemaker C.A., 2007. A Stochastic Radial Basis Function Method for the Global Optimization of Expensive Functions. *INFORMS Journal on Computing* 19 , 497–509. DOI: 10.1287/ijoc.1060.0182
- Rehmer A. and Kroll A., 2019. On Using Gated Recurrent Units for Nonlinear System Identification 2019 18th European Control Conference (ECC), 2504-2509. DOI: 10.23919/ECC.2019.8795631
- Ribes A. and Caremoli C., 2007. Salome Platform Component Model for Numerical Simulation Proceedings of the 31st Annual International Computer Software and Applications Conference - Volume 02, Washington, 553–564. DOI: 10.1109/COMPSAC.2007.185
- Samanta A. and Banerji P., 2010. Structural vibration control using modified tuned liquid dampers. *The IES Journal Part A: Civil & Structural Engineering* 3 (1), 14-27. DOI: 10.1080/19373260903425410

- Sarpkaya T., 2004. A critical review of the intrinsic nature of vortex-induced vibrations. *Journal of Fluids and Structures* 19 (4), 389-447. DOI: <https://doi.org/10.1016/j.jfluidstructs.2004.02.005>
- Sergeyev Y.D., Kvasov D.E. and Mukhametzhanov M.S., 2018. On the efficiency of nature-inspired metaheuristics in expensive global optimization with limited budget. *Scientific Reports* 8 , DOI: 10.1038/s41598-017-18940-4
- Sharma A., Mittal H. and Gairola A., 2018. Mitigation of wind load on tall buildings through aerodynamic modifications: Review. *Journal of Building Engineering* 18 , 180–194. DOI: 10.1016/j.jobbe.2018.03.005
- Tait M.J., El Damatty A.A., Isyumov N. and Siddique M.R., 2005. Numerical flow models to simulate tuned liquid dampers (TLD) with slat screens. *Journal of Fluids and Structures* 20 , 1007-1023. DOI: 10.1016/j.jfluidstructs.2005.04.004
- Tang U.F. and Kwok K.C.S., 2004. Interference excitation mechanisms on a 3DOF aeroelastic CAARC building model. *Journal of Wind Engineering and Industrial Aerodynamics* 92 (14-15), 1299–1314. DOI: 10.1016/j.jweia.2004.08.004
- Tuan A.Y. and Shang G.Q., 2014. Vibration control in a 101-storey building using a tuned mass damper. *Journal of Applied Science and Engineering* 17 , 141-156. DOI: 10.6180/jase.2014.17.2.05
- U. S. Department of Commerce (2020) *National Weather Service*, [Online], Available: <http://www.noaa.gov>.
- Wei D., Wang J., Niu X. and Li Z., 2021. Wind speed forecasting system based on gated recurrent units and convolutional spiking neural networks. *Applied Energy* 292 , 116842. DOI: <https://doi.org/10.1016/j.apenergy.2021.116842>
- Weller H.G., Tabor G., Jasak H. and Fureby C., 1998. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics* 12 (6), 620. DOI: 10.1063/1.168744
- Whiteman M.L., Fernández-Cabán P.L., Phillips B.M., Masters F.J., Bridge J.A. and Davis J.R., 2018. Multi-Objective Optimal Design of a Building Envelope and Structural System Using Cyber-Physical Modeling in a Wind Tunnel. *Frontiers in Built Environment* 4 , DOI: 10.3389/fbuil.2018.00013
- Wind Tunnel Studies of Buildings and Structures* (1999), American Society of Civil Engineers, Available: <https://doi.org/10.1061/9780784403198>.
- Xie J. and Yang X.-y., 2019. Exploratory study on wind-adaptable design for super-tall buildings. *Wind and Structures* 29 (6), 489-497. DOI: <http://dx.doi.org/10.12989/was.2019.29.6.489>

Yang Q., Liu Z., Hui Y. and Li Z., 2020. Modification of aerodynamic force characteristics on high-rise buildings with arrangement of vertical plates. *Journal of Wind Engineering and Industrial Aerodynamics* 200 , 104155. DOI: 10.1016/j.jweia.2020.104155

Appendix A - Code Highlights

This appendix lists key approaches to program the various programs and modules used throughout this work. The discussion refers to a ‘Supplementary Material’ data bank that is available upon request to the major professor.

Appendix A.1. Fluid-Solid-Interaction Simulation

The fluid-solid-interaction simulation code is implemented as a ‘function object’ module for the OpenFOAM® 6 opensource software. Running the ‘pimpleFoam’ executable without any parameters gives the following output:

```
/*-----*\
===== |
\\ / F ield | OpenFOAM: The Open Source CFD Toolbox
\\ / O peration | Website: https://openfoam.org
\\ / A nd | Version: 6
\\ M anipulation |
\*-----*/

Build : 6-76a8284120c3
```

It is important to note the version and build numbers because there are two separate and parallel OpenFOAM implementations and the module implementation may not work for different versions. The full folder tree, source code and binaries for the used version are provided in the ‘openfoam6’ folder of the Supplementary Material.

The module has these main parts: (1) fluid force calculation, (2) rigid body model, (3) mesh motion and (4) logging and simulation control. The module code is fully contained in the ‘openfoam_case/mslsolid’ folder. OpenFOAM dispatches the function object’s ‘mslsolid.C/execute’ method every timestep. This method is the entry point for all the code contained in the module. The fluid force calculation part is implemented in the mslsolid/MSLPatch.C: functionObjects::MSLPatch::getLoads method. The function takes the density of the fluid and the depth of the profile as parameters and returns the forces and moments summed on the patch perimeter and transferred to the center of the patch. MSLPatch is a class that implements a generic boundary. It is subclassed by Building.C and Plate.C which have their own specific methods.

The rigid body model is implemented in ‘mslsolid/Building.C’. The method ‘setLoads’ dispatches getLoads from MSLPatch.C on all the attached plates and the building itself then calculates the forces and moments at the center of the building. The method then proceeds to solve the dynamic system of equations to calculate the timestep deflections.

Mesh motion is implemented in Building.C/move and mesh rotation is implemented in the base class MSLPatch.C/rotate. Logging is implemented in mslsolid.C/write, which writes a single line in an independent timeseries log for each timestep. The written data includes the forces, moments, deflections and velocities. The internal solution state is saved to ‘saved_state.txt’ by mslsolid/saveState to support simulation resuming if it stops abruptly. The mslsolid/loadState’ method is used later to reload the saved state and continue the simulation.

Other files included in the mslsolid folder implement a finite element frame model that was used in earlier iterations of the implementation. Other methods in mslsolid.C are typical for openfoam function objects such as the read method which loads the module configuration from the case file.

Appendix A.2. Mesh Generator

A mesh generator script was written in Python for the SALOME opensource software package. The script permits parametric generation of an arbitrary number of plate interpolation bubbles, surrounding arbitrary rectangular building dimensions. While the script code exists in a single file, it was typically executed by copying and pasting parts of it into the SALOME editor. This allows the user to check the execution of each step. The script lies in the mesh_generator folder of the supplementary materials.

Appendix A.3. Genetic Algorithms

The GAs implementation used throughout this work is provided in the GeneticAlgorithm.py file contained in multiple folders of the Supplementary Materials (e.g., FIU_control/GeneticAlgorithm.py). A brief description for each class and method is provided below:

- GeneSpec: implements a flexible definition of the gene (decision variable) that allows mixing integer and continuous variables and automatic scaling/descaling
 - Generate_value: generates a random value taking into account the type of the gene
 - Scale/descale: perform scaling/descaling into the 0-1 range based on the preconfigured limits

- FitnessSpec: implements a flexible definition of the fitness (objective function) that permits using multiple objectives and constraints with configurable ranges for each
 - Getworst: provides a worst value based on whether the objective is maximization or minimization (used to initialize unknown solutions)
- Genome: implements a solution:
 - Weighted_fitness: return a the weighted fitness for multi-objective cases
 - Mutate: implement mutation for a random number of genes
 - Crossover: implement crossover between two provided genomes and update the current genome. Both single and double-point crossovers are implemented
- Reset_fitness: indicate to the algorithm and the user that this genome needs re-evaluation by setting its fitness to worst fitness
- GeneticAlgorithm: implements the main algorithm logic using tournament selection with a 'tour_size' number of genomes per tournament
 - Run: executes a single iteration of the algorithm. The external user has to loop on 'genomes' and set the fitness for each genome before calling 'run'
 - Tournament_select: selects a genome using tournament select. Used internally, but can also be used externally to select one genome from the current population

Appendix A.4. Iterative optimum training

The Alpine 2 function benchmark implementation code is in `alpine_n4_iterative_code_3` folder of the Supplementary Materials. The IOT implementation exists in the `iterative.py` file. The code uses TensorFlow to implement the MLP neural network and the GeneticAlgorithm library mentioned in the previous appendix. The ‘`schwefel_XX`’ are placeholders for the current function being tested (Alpine 2). This code was tried using many other benchmark functions, including Schwefel. The code makes use of the GPU where possible.

The IOT implementation for the wind tunnel experiments exists in the ‘`small_buidling_iot`’ folder. The procedure is split among two files: ‘`get_best.py`’ and ‘`train_ann.py`’ to allow the user to set the fitness for each solution by running the wind tunnel experiments. The data files in the folder can be used as a reference for the input/output formats of the different steps.

Appendix A.5. Cyber-physical system

The cyber-physical system for the small-scale and large-scale experiments exists in the ‘`small_scale_cyber`’ and ‘`FIU_control`’ folders, respectively. A brief description of each code file is provided below. Any file that is not mentioned below is obsolete and can be ignored:

- `Definitions.py`: Common imports, function definitions, GRU model definition, variable definitions (e.g., `input_steps`, `output_steps`, hardware timestep, GRU timestep ... etc.)
- `CyberPhysical.py`: Core cyber-physical library implementation. Classes and methods are:

- DataSource: a common implementation of used during experiment execution and to retrieve data from previous experiments. The constructor automatically postprocesses the data which ensures the same filtering, and down sampling steps are utilized in all cases.
 - If 'filename' is provided, the code retrieves and prepares stored data.
 - If 'DAQArray' is provided, the code prepares the live DAQ input for further processing by the user. If 'write_file' is provided, the code also writes the DAQArray data simultaneously, so data collection can be implemented during control.
 - Get_sample: returns a sample from the prepared data. The start_time and end_time parameters can be used to extract only a portion from the whole data set. The timestep parameter allows the user to down sample the hardware sampling rate into any other desired sample rate. This function also performs scaling of the variables so they are ready for GRU training
- Sample: implements a scaled, processed sample of data ready for GRU training. Many methods in this class are not used in the final implementation.
 - Get_encdec_io_2: returns the GRU inputs and outputs from the sample. The method first slices the sample into proper sizes, then choses which variables from each timestep are provided for the input/output

- `Get_encdec_input_2`: return the GRU input only. This returns all the available data in the sample as inputs rather than being constrained to return the last portion of the data as an output.
 - `Nopow2` parameter: by default the returned numpy arrays from both functions are trimmed to the length of the nearest power of 2. This is a requirement for training GRUs. If the data will be directly processed by the user (e.g., to calculate Q), and not used for training, set `nopow2` to `True` when calling either function.
- `Output`: A common implementation for evaluating the objective function from the sample data:
 - `Get_rms`: returns the three RMS acceleration components as a tuple
 - `Get_Q`: returns the Q function (RMS sum)
- `Control_2.py`: the final implementation of the GRL training and control algorithm. The same script is used to evaluate random plate positions, refer to the comments for info on how to switch the operating mode
- `Train_encdec.py`: The GRU training and evaluation code
- `Read_continuous_3.py`: GRU training data collection code
- `Random_vs_control.py`: aggregate result plotting code
- `Plot_sample.py`: plot a single wind tunnel experimental sample
- `Plate_change_2.py`: plots the second case of plate change effect
- `Plate_change_1.py`: plots the first case of plate change effect
- `Ltrim.py`: left-trim a data set to possibly to pad it to another one

- `Index_and_compare.py`: compare control and random samples and generate a convergence chart
- `Generate_speed_profile.py`: generates a list of random wind tunnel speed profiles
- `Change_tunnel_speeds.py`: while running, this script randomly changes the wind tunnel speed. To be run simultaneously with `read_continuous_3.py`
- `Change_plate_angles.py`: while running, this script randomly changes the plate positions. To be run simultaneously with `read_continuous_3.py`
- `Build_angle_table_live.py`: constructs an array that maps the encoder positions to the motor command positions. This allows the controller to be resilient against belt slips or model reassembly. This script is run only once before performing a large group of experiments.

Appendix A.5.1. Large-scale experiments

The code for large-scale wind tunnel experiments performed at Florida International University (FIU) is contained in the `FIU_control` folder. The code is essentially a copy of the `small_scale_cyber` folder used for the small-scale experiments. The code was modified to include additional pressure sensors, change GRL parameter values, among other improvements and modifications. A description of additional/modified files is provided below:

- `Sensitivities.txt`: the sensitivity for each accelerometer installed on the top floor of the model
- `Plate_configurations_to_test.txt`: the plate configurations from the Iowa State University horizontal plates façade system that were evaluated during the FIU experiments

- `Arduino_pinout_2.txt`: the mapping of the servo motors in each floor to the pin on the Arduino board
- `Ac_experiment_sets.txt`: the description for each active control experiment set. Active control experiment file names end with experiment set number (e.g., `_T00` refers to the 00 experiment set and `_C01` refers to the 01 experiment set). The T and C letter refers to training and control, respectively.
- `Filename_format.txt`: explains the filename formats for the result files.
- `PPM.py`: pre-programmed façade angle data collection code (for Iowa state façade design)
- `PPM_sm2.py`: pre-programmed façade angle data collection code (for Kansas state façade design)
- `gen_random_plate_combs.py`: generate random plate combinations for the Kansas state façade design
- `COL.py`: GRU training data collection
- `Ac.py`: active control (GRL implementation) for the large-scale model
- `FIU_control.xlsx`: DAQ input array and `CyberPhysical.py` sample ordering of that array data. The second sheet has a random generator for wind tunnel throttle % used by the FIU WoW operators to generate the tunnel speeds for the experiments
- `Accelerometer_data`: data collected from the additional accelerometers that were not used by the controller, but installed by the FIU team on the model

Appendix B - List of Acronyms

AM	Aerodynamic modification
ANN	Artificial neural network
AOA	Angle of attack
CAARC	Commonwealth advisory aeronautical research council
CFD	Computational fluid dynamics
DNN	Deep neural network
DOF	Degree of freedom
FE	Finite elements
FSI	Fluid-solid-interaction
GAS	Genetic algorithms
GRL	Genetic reinforcement learning
GRU	Gated recurrent unit
IOT	Iterative optimum training
LDV	Laser doppler vibrometer
LES	Large eddy simulation
MLP	Multi-layer perceptron
PSD	Power spectral density
RANS	Reynolds averaged Navier-Stokes
RL	Reinforcement learning
RMSE	Root mean square error
SST	Shear stress transport
TLD	Tuned liquid damper
TMD	Tuned mass damper
VIV	Vortex-induced vibration
WIV	Wind-induced vibration