A. Individual Tool Descriptions

A.1. Structural vibration control

This tool calculates the performance and stability of several feedback control strategies for vibration attenuation. The three-storey structure is set into motion by a white noise primary force f_p acting on the first floor. Assuming that each floor can be represented by a lumped mass and that the beams between the storeys can be modelled as parallel springs of negligible mass, the equation of motion of the structure can be written as,

$$\mathbf{M}\ddot{\mathbf{x}}+\mathbf{C}\dot{\mathbf{x}}+\mathbf{K}\mathbf{x}=\mathbf{f}_{ext}, \tag{1}$$

where $\mathbf{x}$ is the displacement vector, $\mathbf{M}$, $\mathbf{C}$, $\mathbf{K}$ are the classical mass, stiffness and damping matrices of a lumped parameter model fixed on one end and free at the other, and $\mathbf{f}_{ext}$ is the vector of external forces (both primary and secondary) acting on each storey.

For the control feedback, the control force is provided by an electromagnetic inertial actuator (Rohlfing et al., 2011) that can be placed on any choice of floor. The inertial actuator is modelled as a moving mass m_p mounted on a suspension, whose stiffness and damping parameters are k_p, c_p, respectively. An actuator is rigidly connected to the structure via a base mass m_b. The secondary force f_s generated by the inertial actuator on the structure is given by the sum of the passive mechanical impedance of the actuator and the active force f_c given by the electromagnetic transducer. The force factor of the transducer is $\Phi = Bl$ where B is the magnetic flux of the permanent magnet and l is the length of the coil exposed to the magnetic flux B. Additionally, the voice coil has an electrical impedance Z_e and the current signal has been amplified by the amplifier gain g_a before entering the actuator's electrical winding.

Since the addition of the inertial actuator, the active structure has an additional DoF (the moving mass with displacement x_p), hence the equation of motion becomes,

$$\mathbf{M}_c\ddot{\mathbf{x}}_c+\mathbf{C}_c\dot{\mathbf{x}}_c+\mathbf{K}_c\mathbf{x}_c=\phi_{pc}\mathbf{f}_p, \tag{2}$$

where $\mathbf{x}_c=\left\{\mathbf{x}\;x_p\right\}$ and the mass, stiffness and damping matrices are defined as,

$$\mathbf{M}_c=\begin{bmatrix}\mathbf{M}+\phi_s^T m_b\phi_s & \mathbf{0}\\ \mathbf{0} & m_p\end{bmatrix},\ \mathbf{K}_c=\begin{bmatrix}\mathbf{K}+\phi_s^T k_p\phi_s & -k_p\phi_s^T\\ -k_p\phi_s & k_p\end{bmatrix},\ \mathbf{C}_c=\begin{bmatrix}\mathbf{C}+\phi_s^T c_p\phi_s & -c_p\phi_s^T\\ -c_p\phi_s & c_p\end{bmatrix}, \tag{3}$$

and $\phi_{pc}=\left\{\phi_p\;0\right\}$ with $\phi_p=\{1\;0\;0\}$ is the location of the primary force on the structure and ϕ_s defines the location of the secondary force, for example $\phi_s=\{1\;0\;0\}$.

Four types of control strategies are investigated, which can be selected via a radio button input, hence only one control strategy can be applied at a time. These four control strategies were selected to showcase different approaches: passive versus active, with the active control approach further divided into a model-based control strategy (Linear-Quadratic Gaussian regulator) and a non-model-based control strategy (Direct Velocity Feedback control). The non-model-based approach is subdivided in two options: the Direct Velocity Feedback and the Direct Velocity Feedback with the addition of a tunable electronic compensator. In this way, different control strategies with different settings lead to different behaviours of the structure, which in a future development of DTOP-Cristallo could inform an additional decision module that will pick the best control strategy to be used based on the past and current states of the physical twin. The parameters that characterise each control strategy can be tuned using several numerical and slider inputs. The vibration attenuation performance of these control strategies is evaluated using the time-averaged kinetic energy of the structure, which is defined as,

$$T(j\omega)=\frac{1}{4}\dot{\mathbf{X}}^H(j\omega)\mathbf{M}_*\dot{\mathbf{X}}(j\omega), \tag{4}$$

where H stands for Hermitian transpose, and $\dot{\mathbf{X}}(j\omega)$ is the Fourier transform of the velocity. The velocity of the structure can either be measured directly with a laser vibrometer or calculated from accelerometer or strain gauge measurements. The vibration attenuation performance of the control strategies is compared in terms of same control efforts $P(j\omega) = f_c^*(j\omega)f_c(j\omega)$, where * denotes the complex conjugate, and $f_c(j\omega)$ is the active control force that acts on the structure by reacting against the proof mass of the actuator.

The first control strategy is a passive tuned mass damper (TMD), which is a passive control method that does not require an external power supply and is based on natural frequency matching between the TMD and the structure (fundamental frequency predominately). The other control strategies require an active feedback loop, where the feedback signal is given by a single co-located velocity sensor at the actuator position. This is similar to the schematic scenario shown in the IoT layer of Fig. 2.

Two different active control architectures are implemented within the Python tool. Firstly, a linear-quadratic-Gaussian (LQG) regulator (Preumont, 1997), which is a model based control strategy, and secondly a direct velocity feedback (DVF) controller (Elliott et al., 2001), which is a non-model based approach. The control force generated by the LQG regulator is

$$f_c = -g_a \Phi \mathbf{K}_{lqr} \hat{\mathbf{w}}, \tag{5}$$

where g_a is the amplifier gain, $\hat{\mathbf{w}}$ are the estimates of the system states $\mathbf{w}$ and $\mathbf{K}_{lqr}$ is the optimal full-state feedback gain matrix that minimises the following quadratic cost function,

$$J = \int_0^\infty \left(\mathbf{w}^T \mathbf{Q} \mathbf{w} + f_c^T R f_c \right) dt, \tag{6}$$

where $\mathbf{Q} = q\mathbf{I}$, with $q_{4,4} = 0$ and $q_{8,8} = 0$ is the state weight matrix, where q defines the scalar weight, which is equal for all states, except those associated with the proof mass of the actuator, and R is the cost weight. The control force of the DVF controller is,

$$f_c(j\omega) = -g_a \Phi h \phi_s \dot{\mathbf{x}}, \tag{7}$$

where h is the feedback gain.

In addition to the regular DVF controller, a modified DVF with the addition of an electronic compensator (DVF+EC) (Elliott et al., 2012) is also available, where the control force can be rewritten as,

$$f_c(j\omega) = -g_a \Phi C_p(j\omega) h \phi_s \dot{\mathbf{x}}, \tag{8}$$

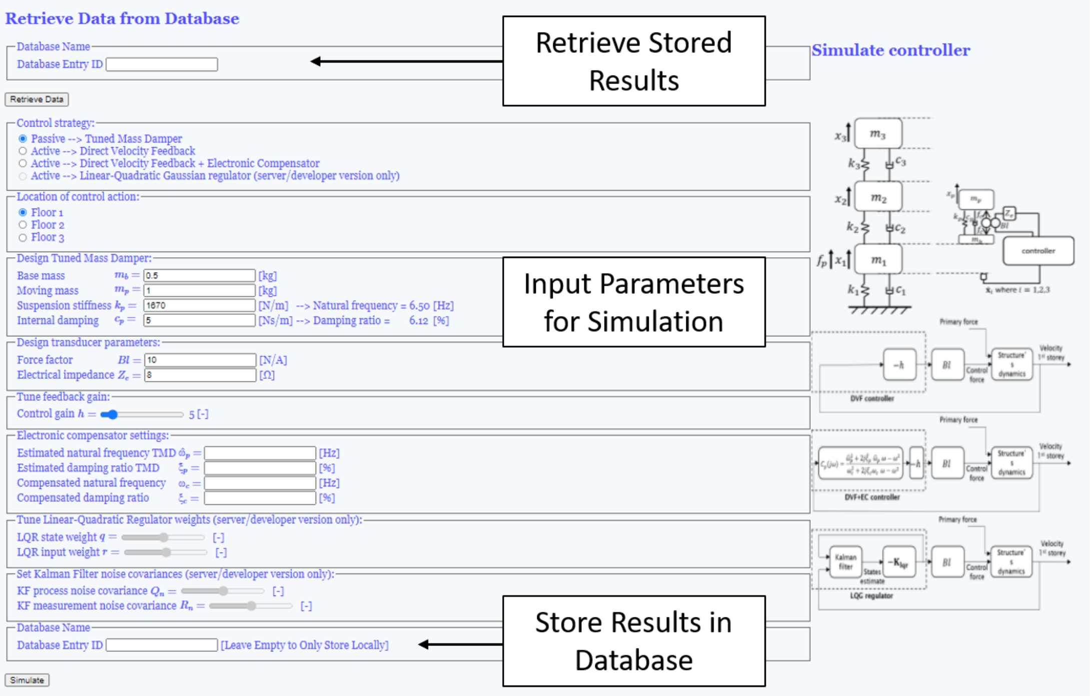
where $C_p(j\omega)$ is an electronic compensator, which was introduced in (Elliott et al., 2012) and results in,

$$C_p(j\omega) = \frac{\hat{\omega}_p^2 + 2j\hat{\zeta}_p \hat{\omega}_p \omega - \omega^2}{\omega_c^2 + 2j\zeta_c \omega_c \omega - \omega^2}, \tag{9}$$

where $\hat{\omega}_p = \sqrt{k_p/m_p}$ is the estimated natural frequency of the actuator, $\hat{\zeta}_p = c_p/(2\sqrt{k_p m_p})$ is the estimated internal damping of the actuator, whereas ω_c, ζ_c are the compensated natural frequency and damping ratio, respectively. The electronic compensator was introduced in order to lower the apparent natural frequency of the actuator and to increase its apparent internal damping in order to improve the stability of the controller. The vibration attenuation performance is measured in terms of time-averaged kinetic energy reduction with respect to the uncontrolled scenario. The mathematical formulation that has been used to calculate the response of the structure under the above control strategies is documented in (Dal Borgo et al., 2020).

Optimisation-based control methods are currently not present in DTOP-Cristallo, however, if an open-source Python optimisation toolbox would become available, then DTOP-Cristallo could be updated to interface with it.

To demonstrate some additional features of the server version of DTOP-Cristallo, this tool is discussed in further detail. The user-interface for this tool is shown in Fig. 6. There are four main aspects to this interface with the first being a schematic for the system of interest and control algorithms. This schematic is used as a descriptive guide to enable users to understand the required input parameters.



Retrieve Data from Database

Database Name
Database Entry ID □

Retrieve Data

Control strategy:
● Passive --> Tuned Mass Damper
○ Active --> Direct Velocity Feedback
○ Active --> Direct Velocity Feedback + Electronic Compensator
○ Active --> Linear-Quadratic Gaussian regulator (server/developer version only)

Location of control action:
● Floor 1
○ Floor 2
○ Floor 3

Design Tuned Mass Damper:
Base mass $m_b =$ 0.5 [kg]
Moving mass $m_p =$ 1 [kg]
Suspension stiffness $k_p =$ 1670 [N/m] --> Natural frequency = 6.50 [Hz]
Internal damping $c_p =$ 5 [Ns/m] --> Damping ratio = 6.12 [%]

Design transducer parameters:
Force factor $Bl =$ 10 [N/A]
Electrical impedance $Z_e =$ 8 [Ω]

Tune feedback gain:
Control gain $h =$ 5 [-]

Electronic compensator settings:
Estimated natural frequency TMD $\hat{\omega}_p =$ □ [Hz]
Estimated damping ratio TMD $\xi_p =$ □ [%]
Compensated natural frequency $\omega_c =$ □ [Hz]
Compensated damping ratio $\xi_c =$ □ [%]

Tune Linear-Quadratic Regulator weights (server/developer version only):
LQR state weight $q =$ [-]
LQR input weight $r =$ [-]

Set Kalman Filter noise covariances (server/developer version only):
KF process noise covariance $Q_n =$ [-]
KF measurement noise covariance $R_n =$ [-]

Database Name
Database Entry ID □ [Leave Empty to Only Store Locally]

Simulate

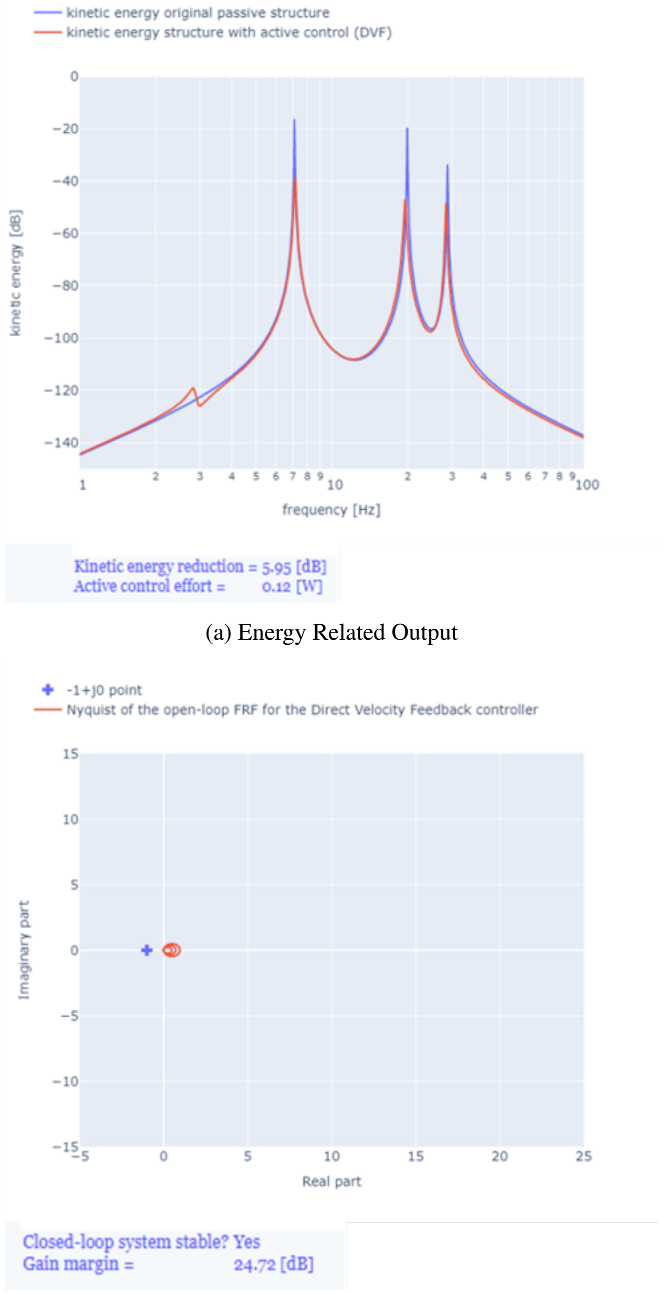
Simulate controller

Figure 6. User-interface for the structural vibration control simulation.

The second main aspect is the selection of input parameter values. These parameters demonstrate the wide variety of possible inputs types for numerical simulations. Firstly, there are Boolean inputs (passive/active control strategy) where only a single value is selected. Secondly are numerical values. These can be a value of any resolution, however it is constrained to match physical meaning (positive mass for example). The last type of input in this aspect is a slider input that has a specified range. Selecting the type of input is a design choice for the tool.

The remaining two aspects of this simulation relate to the database storage and retrieval. At the bottom of the page, there is a text entry box that is used to store information into the database. When there is a string in this box, the simulation results are both returned to the user and saved in the database. Above the input parameter section, a similar text entry box is used to retrieve the information stored in the database. This searches the database for the unique name presented to it and gathers the results from the specified simulation.

Regardless of the method used to solve/store the simulation results, this simulation produces results for two main purposes, the kinetic energy and stability changes caused by the control system. These can be seen in Fig. 7. The kinetic energy outputs are shown in Fig. 7a through a graphical frequency response and two scalar values. The graphical response shows the change in the dynamic response of the system that can demonstrate the effectiveness of the controller for the system. This can be used to show changes in the natural frequency and decreases in the resonance amplification. For the scalar inputs, the kinetic energy reduction integrates the difference between the controlled and non-controlled system to show how much energy is reduced by adding the control system. The other scalar input gives a measure of how much work is done by the control system, which is related to the operating cost of the control system.



Kinetic energy reduction = 5.95 [dB]
Active control effort = 0.12 [W]

(a) Energy Related Output

Closed-loop system stable? Yes
Gain margin = 24.72 [dB]

(b) Stability Related Output

Figure 7. *Main outputs of the structural vibration control tool.*

The second output relates to the stability of system and is seen in Fig. 7b. This graph shows the Nyquist plot of the open-loop controller that visualises the effect that the control system has on the stability of the system. Additionally, the tool also predicts if the system is stable for a closed-looped controller and the gain margin for the control algorithm.

The example simulation shown in Fig. 7 is one typical type of simulation for these mechanical systems. These take in Boolean and numerical values and produce visual and tabular outputs that an analyst can use for a range of different situations. For example, a situation that analysts may be interested in are design modifications. In order to predict if a control strategy will work, analysts use these types of numerical simulations before creating a prototype to test on the physical system. While the shown simulation for the control strategy is currently based on numerical data (for the reasons mentioned earlier), other future tools will be able to incorporate a combination of numerical modelling and experimental data. These tools require both input from the user for the numerical modelling and the selection of what experimental data to incorporate. In the implementation, it is easy to generate a list of stored data within the database. So these tools would take in Boolean, numerical, and list based input parameters.

A.2. Design under uncertainty

This tool is designed to analyse the propagation of uncertainties through the digital twin in the context of modifying the design. Additionally, this tool assesses the degree to which the uncertainties associated with each of the uncertain input parameters affects the uncertainty of the design requirements. This is achieved by performing a sensitivity analysis (Liu et al., 2005; Saltelli, 2002) based on the Fisher information matrix (Cover, 1999), which can be computed from Monte Carlo realisations using Python. The standard form of the Fisher information matrix derived from the design entropy can be expressed as

$$F_{jk} = \int \frac{1}{p\left(\mathbf{y} \mid \mathbf{b}\right)} \frac{\partial p\left(\mathbf{y} \mid \mathbf{b}\right)}{\partial \overline{b}_j} \frac{\partial p\left(\mathbf{y} \mid \mathbf{b}\right)}{\partial \overline{b}_k} \mathrm{d}\mathbf{y}, \tag{10}$$

where $\mathbf{b}$ is the vector of the parameters of the distribution of the input variables and $p\left(\mathbf{y} \mid \mathbf{b}\right)$ is the conditional joint Probability Density Function (PDF) of the output vector $\mathbf{y}$. The partial derivatives of the joint-PDF with respect to the j^{th} and k^{th} parameter of the distribution b are known as the sensitivities of the output distribution. The Fisher eigenvectors associated with the highest eigenvalues then give information regarding the degree of impact of the uncertainties on each of the parameters of the input distributions in terms of the uncertainties of the output requirements.

The tool is comprised of four modules to help organise, sequentially, the Fisher information-based analysis. The input parameters (for a vibration analysis) of the three-storey structure are the mass, stiffness, damping, and input force for each of the three floors. These parameters can be either deterministic or stochastic. The first module presents a set of tables that allows the user to select the type of distribution for each parameter and hyper-parameters for the chosen distribution (Gaussian and Gamma are currently implemented). If any of the inputs is deterministic, the user can select the Gaussian distribution and set the value of zero for the standard deviation. Once completed, the user can press “enter” on the keyboard to plot the displacement frequency response function for each floor. At the bottom of the page the other analysis parameters are chosen, such as the frequency at which the user is carrying out the analysis, the number of Monte Carlo realisations requested, and the outputs of interest (floor displacement and shear force between floors).

After pressing “enter input data” on the first module, the user can move to the second module by clicking on “go to output distribution page”, where the distribution of the requested outputs are plotted. On this page, the user can select the number of bins for the histograms of each output, and the number of bins for the joint distribution used in the Fisher information matrix. This is shown in the third module by clicking on the “go to Fisher information page” link. The third module presents the eigenvalues and eigenvectors of the Fisher information matrix at the bottom of the page, and the user can navigate

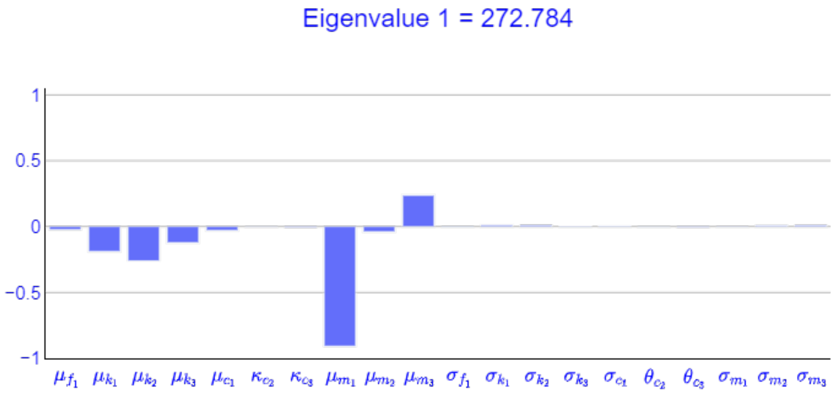


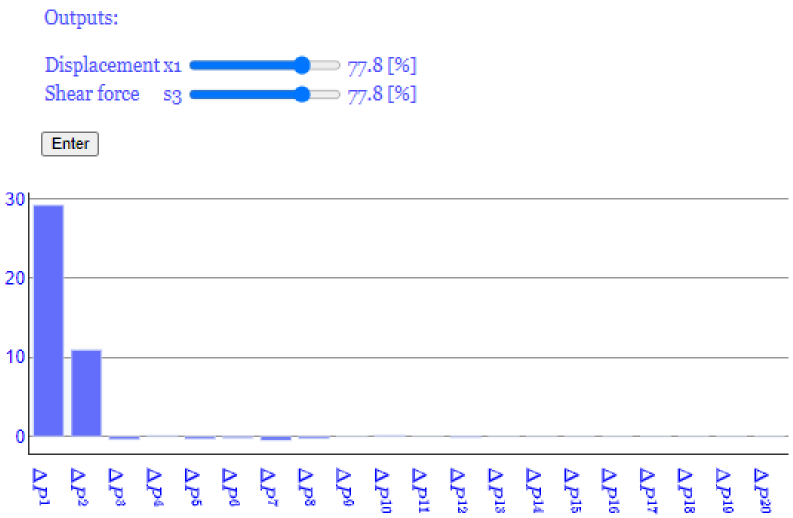
Figure 8. *Example output from the "Design Under Uncertainty" tool: Fisher eigenvector associated to the largest eigenvalue.*

through them by pressing the "previous" or "next" buttons. An example of this eigenvalue output is shown in Fig. 8 for the default parameter values. This information portrays the degree of impact that the uncertainties have on the design output which can be visualised via the magnitude on the dominant eigenvectors. In Fig. 8, it can be seen that the uncertainty of the mean of the first floor mass is dominant in terms of the first eigenvalue.

Finally, the user can move to the fourth module by clicking on "go to sensitivity analysis" where a further analysis of the Fisher eigenvectors is performed to assess the effect of the uncertainties in the probability of failure. It is found that the projection of the Fisher eigenvectors onto the sensitivity vectors are proportional to the contribution of each eigenvector to the variation of the probability of failure. In this module, the user must select the reliability level of interest (inversely related to the probability of failure). The reliability of the system is given as the probability that the output is less or equal to a target value. In this implementation, this target is specified as a percentage of the maximum value found in the Monte Carlo sampling. After defining this criteria, the user can press "enter" to plot the magnitude of the projection of each Fisher eigenvector onto the sensitivity vectors. This diagram is helpful to assess the dominance of each parameter on the variation of the probability of failure, as can be seen in Fig. 9, where, for the selected reliability, uncertainties in the probability of failure are dominated by the information given by the first two Fisher eigenvectors.

A.3. Uncertainty propagation

In this simulation tool, Python is used to calculate the uncertainty bounds on the frequency response function based on the tolerance interval of the input parameters. To calculate the bounds, initially, the deterministic solution is calculated using linear algebra. The deterministic solver outputs the response of the structure in terms of displacement in the frequency domain. This tool analyses the effect of the uncertainty characterised by intervals with varying bounds, for the mass, stiffness, and damping parameters at each level of the structure. Mass, stiffness, and damping are assigned interval values based on the user's input. The user can define the uncertainty of each of these nine parameters by means of sliders, where the value of the slider quantifies the amount of relative uncertainty each parameter carries. Because the value of M, K, and C are constrained to be positive, the maximum relative uncertainty



Outputs:

Displacement x1 77.8 [%]

Shear force s3 77.8 [%]

Figure 9. *Projection of the eigenvectors onto the sensitivity vectors for the probability that the displacement of the first floor and shear force at the third floor are less than the 77.8% of the maximum corresponding outputs.*

allowed in this example is 100%. Within the tool, the user can also change the nominal parameter values, however it defaults to the nominal values. These intervals can be pushed through the deterministic simulation to identify the bounds on the displacement frequency response function. For more details on how to propagate interval uncertainty the reader is referred to (Gray et al., 2020; DeAngelis et al., 2020). The plots of the frequency response function bounds for each level are shown on the right-end side of the page. These bounds are computed using two different methods, random uniform samples compared with samples arranged in a regular grid. The second method, albeit more expensive, generally yields more accurate bounds than the first. The user interface has controls for the number of samples used to obtain the bounds. To demonstrate this, Fig. 10 gives an example output from this tool. For each floor, there are three plots, the nominal/mean frequency response, the bounds given by the purely random selection labelled as Monte Carlo, and the bounds given from the uniform sampling labelled as Caresian.

The system's matrices are constructed by the classical mass, stiffness, and damping matrices procedure. Under stationary harmonic excitation, the displacement at floor i can be obtained in the frequency domain by inverting the following matrix:

$$D = -\omega^2 M + j\omega C + K \tag{11}$$

where j is the imaginary unit, and ω is the excitation frequency of interest. The displacement x_i at floor i, given a harmonic excitation of intensity f_k at floor k, is obtained with matrix multiplication:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = D^{-1} \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix}. \tag{12}$$

Under the assumption that the uncertainty in the mass, stiffness and damping arises from imperfections in the micro-structure of the material, and that the size and volume of each component can be

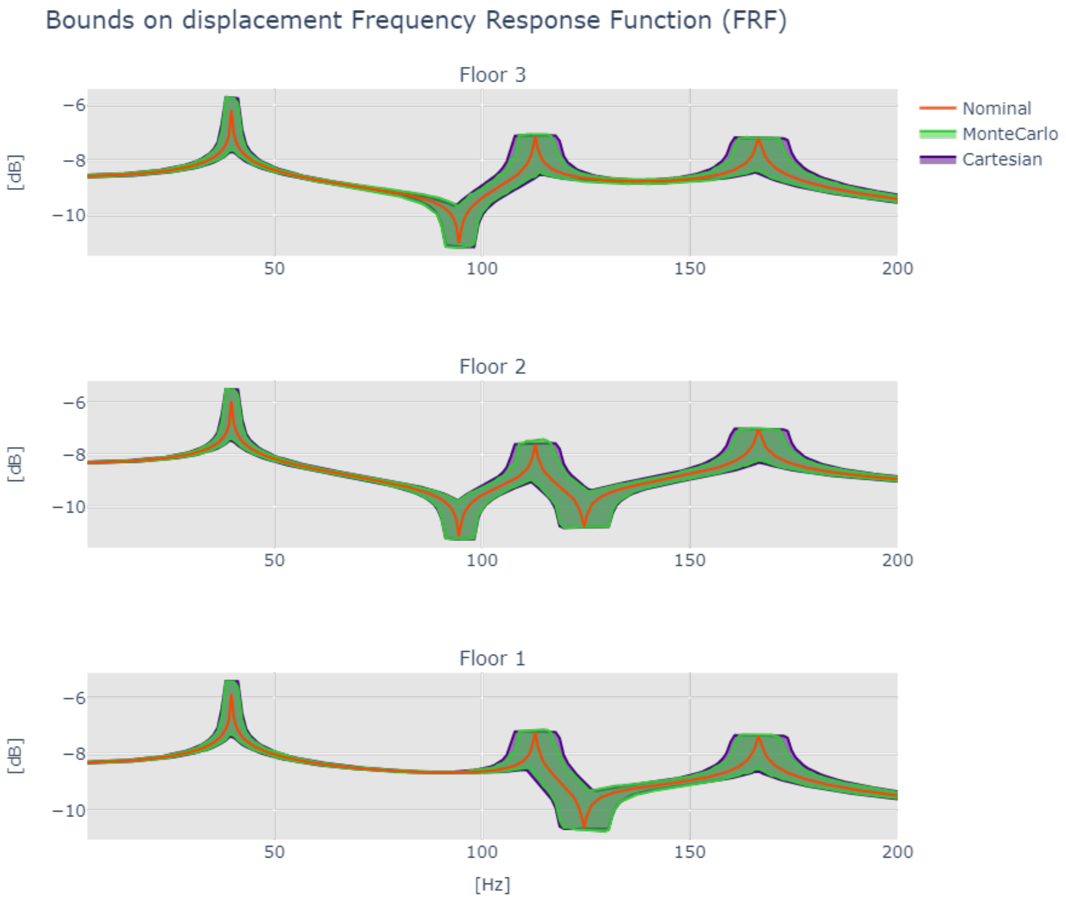


__Figure 10.__ Example output from the "Uncertainty Propagation" tool.

measured to arbitrary precision, we can assume one uncertain dimension for each physical quantity. This can be expressed in mathematical notation as follows:

$$[M] = M + M_e\ \Delta_m;\quad [K] = K + K_e\ \Delta_k;\quad [C] = C + C_e\ \Delta_c; \tag{13}$$

$M_e,\ K_e,\ C_e$ are matrices with upper bound error values provided by the user, $[M],\ [K],\ [C]$ are the intervals for the matrices, and $\Delta_m, \Delta_k, \Delta_c = [-1, 1]$ are unitary intervals.

A.3.1. Strategy for propagation

The goal of the uncertainty quantification tool is the rigorous propagation of the bounds expressed in Eq. 13 through the deterministic solver of Eq. 12. Because of the relatively low dimensional problem, a good inner approximation of the bounds can be obtained via either random or grid sampling. This kind of propagation is non-intrusive, because the solver need not change its internal arithmetic rules. For this problem, an intrusive solver would have needed to track the functional dependencies encoded by Eq. 12, which are rather complex due to the matrix inversion. Although this may sound daunting, the advantages of using intrusive solvers can outweigh the inconvenience of tracking dependencies, especially when the uncertainty dimension is high.

With random sampling the continuum identified by the interval matrices of Eq. 13 is discretised by randomly drawn values within the intervals. The uncertainty is then propagated by evaluating Eq. 12 for all samples, and taking the minimum and maximum of the output displacement for each given frequency. The process articulates in the following steps:

1. Sample from the interval matrices $[M]$, $[K]$, $[C]$ based on the number of samples selected by the user.
2. Evaluate the deterministic Eq. 12 for each random draw $i = 1, \ldots, N$. Equation 12 can be expressed in implicit notation as $g : \mathbb{R}^{10} \to \mathbb{R}^3$, with $x_{[1:3]} = g(\omega, M, K, C)$. So at each random draw the displacement is:
$$x_{[1:3,i]}(\omega) = g(\omega, M_i, K_i, C_i)$$
3. Find the indices i_1, i_2, i_3 for which the displacement vector is maximum (and minimum):
$$i^{\max}_{[1:3]} = \arg\max\{x_{[1:3,i]}(\omega) : i = 1, \ldots, N\}, \quad i^{\min}_{[1:3]} = \arg\min\{x_{[1:3,i]}(\omega) : i = 1, \ldots, N\}.$$
4. Collect the inner approximation of the bounds on the displacement vector:
$$[x](\omega) = [x_{[1:3], i^{\min}_{[1:3]}},\ x_{[1:3], i^{\max}_{[1:3]}}]$$

The process repeats with the same samples from $[M]$, $[K]$, $[C]$ for all values of ω in the frequency range of interest.

With grid sampling, the procedure to obtain the bounds is very similar. The only difference is in the way the samples are generated. In the grid sampling, instead of pulling random samples from a bounded uniform distribution in Step 1, linear spaced samples are selected within the 3 dimensional space spanned by the mass, stiffness, and damping intervals. This results in N^3 evaluations of Eq. 12. Because of the high cardinality, the number of samples chosen for the grid sampling in each dimension is typically lower than the randomly chosen values from the random sampling.

A.4. Control-based continuation

This tool demonstrates a control-based continuation (Sieber et al., 2008) on the three-storey structure. Control-base continuation is a technique for tracing steady-state periodic solutions in nonlinear dynamic systems. To introduce nonlinearity in the three-storey structure (which is predominately linear at the macro scale), a cubic stiffness is added between the first floor and the base.

This control-based continuation tool offers three simulation results, open-loop frequency sweep, open-loop (forcing) amplitude sweep and control-based continuation of the solution branch with varying forcing amplitude. An example of these three results is shown in Fig. 11. In the open-loop tests, the bifurcation parameter (either the forcing frequency or the forcing amplitude) is first increased then decreased in a given interval then, after the transient decays, the vibration amplitude is registered. The open-loop test captures the stable steady-state solutions of the system revealing ‘bistable’ parameter domains where two stable periodic orbits coexist. This coexistence produces a region where the natural frequency and frequency response is dependent on the forcing amplitude. In these parameter domains, the system also has an unstable periodic solution which is undetectable in open-loop simulations and experimentation. In contrast, both the stable and the unstable solutions can be traced with control-based continuation. Thus, this technique can reveal more information about the structure than is possible with the conventional open-loop approach.

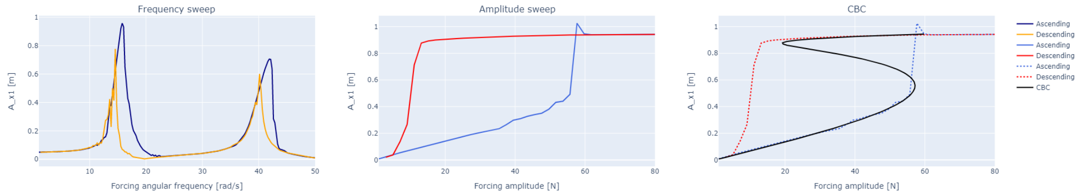


Figure 11. *Example output from the "Control-Based Continuation" tool.*

In order to trace the unstable solutions of the system, a proportional-derivative control is applied. The control-force, applied to the first floor of the structure, is given by

$$F_{\mathrm{ctrl}}(t) = k_{\mathrm{p}}(x_{\mathrm{t}}(t) - x_1(t)) + k_{\mathrm{d}}(\dot{x}_{\mathrm{t}}(t) - \dot{x}_1(t)), \tag{14}$$

where $x_1(t)$ is the displacement of the first storey. Provided that the control gains $k_{\mathrm{p}}, k_{\mathrm{d}}$ are chosen appropriately, the control in Eq. 14 stabilises the system at a steady-state periodic orbit. Then we perform an iteration on the control target to find the steady-state solution of the open-loop system. For this purpose, the displacement $x_1(t)$ and control target $x_{\mathrm{t}}(t)$ is considered by their truncated Fourier series

$$x_1(t) \approx A_0 + \sum_{k=1}^{N} A_k \cos(k\omega t) + B_k \sin(k\omega t), \tag{15}$$

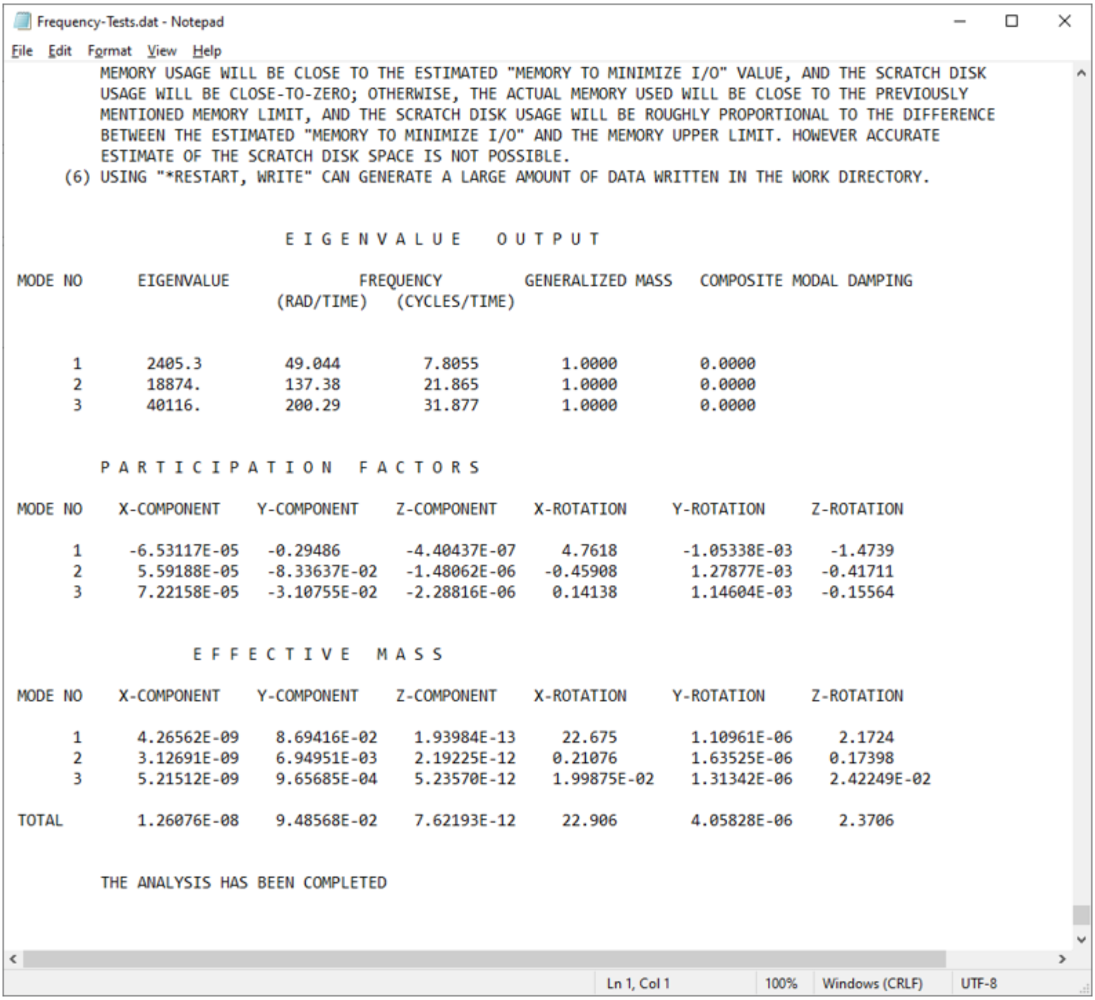
$$x_{\mathrm{t}}(t) \approx A_{0\mathrm{t}} + \sum_{k=1}^{N} A_{k\mathrm{t}} \cos(k\omega t) + B_{k\mathrm{t}} \sin(k\omega t). \tag{16}$$

With the correctly chosen control target coefficients, the displacement should converge to the control target: $A_k \longrightarrow A_{k\mathrm{t}}$, $B_k \longrightarrow B_{k\mathrm{t}}$ for $k \neq 1$ resulting in $F_{\mathrm{ctrl}}(t) \longrightarrow A_1^* cos(\omega t) + B_1^* sin(\omega t)$. In this case, the control is both stabilising and non-invasive and the controlled system converges to the steady-state solution of the open-loop system regardless of its stability. The forcing amplitude corresponding to the limit cycle can be calculated from the remaining forcing coefficients as $\sqrt{A_1^{*2} + B_1^{*2}}$. The continuation for the branch of solutions then moves forward with increasing the fundamental harmonic coefficients $A_{1\mathrm{t}}$ and $B_{1\mathrm{t}}$ of the control target, starting the iteration to find the next limit cycle. Further details of the control-based continuation technique are given in (Beregi et al., 2021).

The numerical simulation is realised in Python using Flask and Javascript to interact with the DTOP such that the progress of the simulations can be followed in a series of bifurcation diagrams that automatically update themselves. The DTOP enables the modification of the system and algorithm parameters; thus, a user can either ask for a faster solution with the diagrams updated more frequently with less accuracy or opt to wait longer in order to identify the steady-state solutions of the system more accurately. In addition to displaying these results to the user, the results can also be downloaded locally via a text file.

A.5. ABAQUS FEA model generation

This tool is designed to be a prototype for the connectivity of the DTOP to non-Python simulations and third party software, e.g. as part of the cloud layer shown in Fig. 2. Specifically, this tool interacts with ABAQUS to perform a Finite Element Analysis (FEA). In order to do this the ABAQUS Python scripting language is used. While there are multiple ways to create this script, the "record macros"



```
         MEMORY USAGE WILL BE CLOSE TO THE ESTIMATED "MEMORY TO MINIMIZE I/O" VALUE, AND THE SCRATCH DISK
         USAGE WILL BE CLOSE-TO-ZERO; OTHERWISE, THE ACTUAL MEMORY USED WILL BE CLOSE TO THE PREVIOUSLY
         MENTIONED MEMORY LIMIT, AND THE SCRATCH DISK USAGE WILL BE ROUGHLY PROPORTIONAL TO THE DIFFERENCE
         BETWEEN THE ESTIMATED "MEMORY TO MINIMIZE I/O" AND THE MEMORY UPPER LIMIT. HOWEVER ACCURATE
         ESTIMATE OF THE SCRATCH DISK SPACE IS NOT POSSIBLE.
     (6) USING "*RESTART, WRITE" CAN GENERATE A LARGE AMOUNT OF DATA WRITTEN IN THE WORK DIRECTORY.


                              E I G E N V A L U E    O U T P U T

 MODE NO      EIGENVALUE              FREQUENCY         GENERALIZED MASS   COMPOSITE MODAL DAMPING
                             (RAD/TIME)   (CYCLES/TIME)


      1        2405.3         49.044         7.8055         1.0000         0.0000
      2        18874.         137.38         21.865         1.0000         0.0000
      3        40116.         200.29         31.877         1.0000         0.0000


         P A R T I C I P A T I O N   F A C T O R S

 MODE NO     X-COMPONENT    Y-COMPONENT    Z-COMPONENT    X-ROTATION     Y-ROTATION     Z-ROTATION

      1      -6.53117E-05   -0.29486       -4.40437E-07    4.7618        -1.05338E-03   -1.4739
      2       5.59188E-05   -8.33637E-02   -1.48062E-06   -0.45908        1.27877E-03   -0.41711
      3       7.22158E-05   -3.10755E-02   -2.28816E-06    0.14138        1.14604E-03   -0.15564


                    E F F E C T I V E   M A S S

 MODE NO     X-COMPONENT    Y-COMPONENT    Z-COMPONENT    X-ROTATION     Y-ROTATION     Z-ROTATION

      1       4.26562E-09    8.69416E-02    1.93984E-13    22.675         1.10961E-06    2.1724
      2       3.12691E-09    6.94951E-03    2.19225E-12   0.21076         1.63525E-06   0.17398
      3       5.21512E-09    9.65685E-04    5.23570E-12    1.99875E-02    1.31342E-06    2.42249E-02

 TOTAL        1.26076E-08    9.48568E-02    7.62193E-12    22.906         4.05828E-06    2.3706


         THE ANALYSIS HAS BEEN COMPLETED
```

Figure 12. *Example .dat file output from the "ABAQUS FEA Model Generation" tool.*

functionality is used. This records the user's actions in the CAE environment into the Python scripting language. After the actions are recorded, the script is then generalized to accept the parametric input given by the user via the DTOP. For this implementation, the analysis performs a simple Lanczos algorithm to calculate all the frequencies below the user-defined maximum frequency.

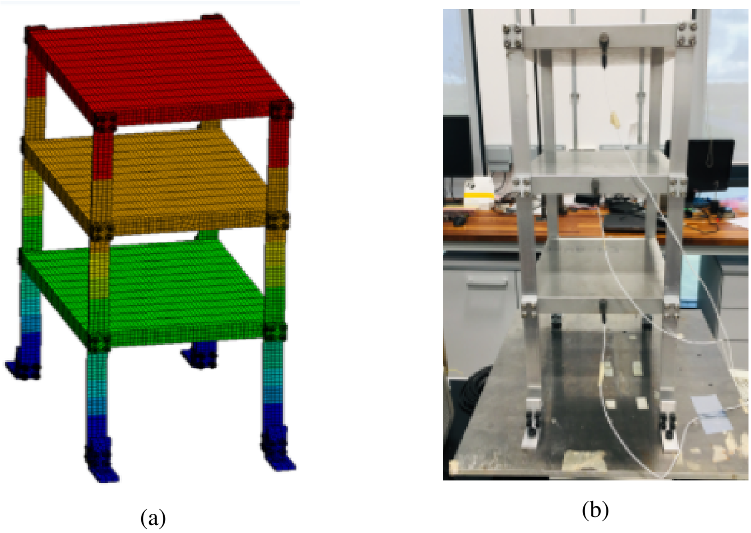
Once the inputs from the DTOP are added to the generalized script, ABAQUS is then called via a command prompt. In order for this to perform correctly in the current implementation, ABAQUS must be installed on the local machine and be callable via the command prompt "abaqus cae". In this current version, the code only performs the simulation and stores the results in the local scratch folder, which is set by default to "dtApp /dtData /ABAQUS_Scratch". The main outputs are the ".dat" and ".odb" files. These contain the frequency results in a text file, with an example shown in Fig. 12, and ABAQUS file respectively. Future versions will include a method to read these results in and display it to the user or use it for other simulation workflow such as model updating.

A.6. Experimental data cross-validation

The data flowing between the physical and digital twin has become a key part of the digital twin concept. It represents the initial step towards constructing a fully developed digital twin and DTOP. This tool presents a validation of the data from the experimentally measured and numerically simulated data obtained from the physical and digital twins.

The design of the physical twin structure for this model is a three-storey structure as shown in Figure 13a. Each floor is composed of a aluminum block with dimensions $12 \times 10 \times 1$ inch ($305 \times 254 \times 25$ mm) resulting in a mass of 11.7 lbs (5.36 kg). The three floors are joined by four vertical columns where each column is dimensions $22 \times 1 \times 0.125$ inch ($559 \times 25 \times 5$ mm) with a mass of 0.27 lbs (0.12 kg). To join the floors to the columns, blocks of aluminium are used that have dimensions of $1 \times 1 \times 0.5$ inch ($25 \times 25 \times 12.7$ mm) with a mass of 0.05 lbs (0.018 kg). 16 blocks were used in total for the three-storey structure. Aluminium 6082 with following parameters was assigned as the structure material;$E = 70$GPa, $\nu = 0.33$ and $\rho = 2.7$g/cm^3 where E is the Young's modulus, ν is the Poisson's ratio and ρ is the density.

Four prototypes of the three-storey structure are designed and manufactured in four different locations (Swansea University, University of Bristol, University of Southampton and University of Sheffield) and all data collected from the different prototypes and the models are presented in this tool. To maintain consistency between the prototypes, several aspects of the experimental setup were specified: The feet of the structure are bolted to a table with tapped holes, and a constant value of torque is used in all bolts of the structure. Three miniature PCB accelerometers with 0.5g mass, 10kHz frequency range and 10mg/s^2 sensitivity, were bonded to the middle of each floor. The first floor was excited by the impact hammer. The experimental measurements were acquired using a Data Physics DAQ system for a period of 8s with a sampling rate of 25.6kHz. The samples were averaged for 10 trials to smooth the noisy signals. A photograph of the experimental setup are shown in Fig. 13b. All measured responses come as frequency response signals. Similarly, a numerical model is performed for each prototype for validation purposes.


(a)

(b)

Figure 13. The three-storey structure shown: (a) FEA model design, and (b) the physical structure.

All of the collected data is presented in interactive figures equipped with a set of tools to inspect and explore the graphs. This tool is composed of four sections. The first section presents all experimental measurements for comparison, while the second section presents the numerical data. In the third section,

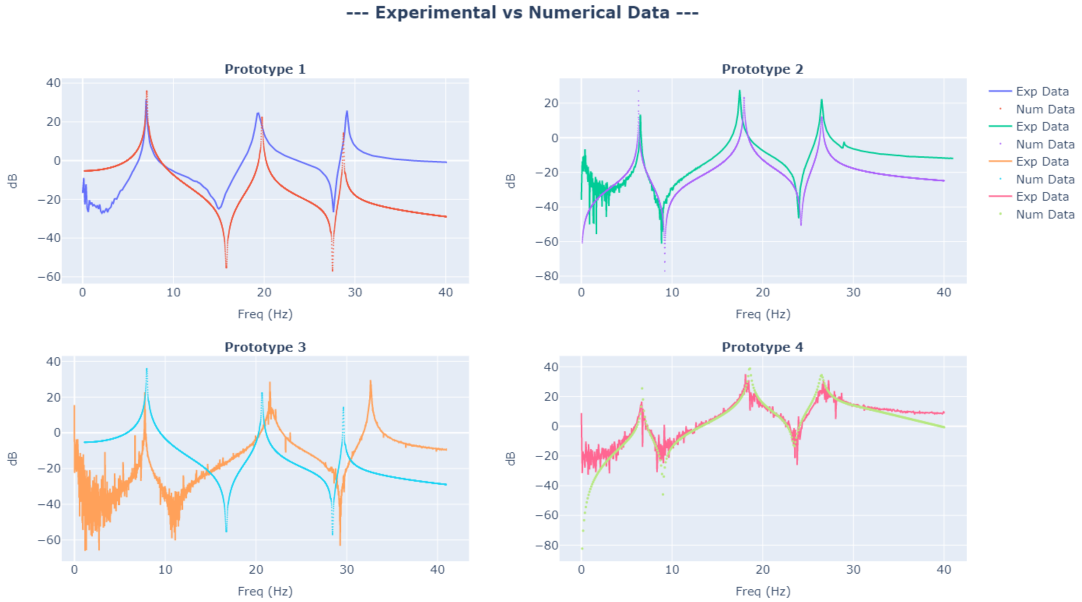


Figure 14. *Example comparison from the "Experimental Data Cross-Validation" tool.*

the numerical and experimental data for each prototype are compared in order to verify the accuracy of the numerical models. This section shown in Fig. 14 to show the type of figures found in this tool. Finally, in the fourth section, the experimental data collected from the four different prototypes are compared. This section can communicate the data to the various models for further studies such as uncertainty propagation or updating active control parameters.