Proceedings of the 35th European Safety and Reliability & the 33rd Society for Risk Analysis Europe Conference
Edited by Eirik Bjorheim Abrahamsen, Terje Aven, Frederic Bouder, Roger Flage, Marja Ylönen
 Published by Research Publishing, Singapore.
doi: 10.3850/978-981-94-3281-3_ESREL-SRA-E2025-P5658-cd

Data-driven Model Updating Solution for the NASA and DNV Challenge 2025 on Optimisation under Uncertainty with Flow-based Neural Networks

Tairan Wang, Sifeng Bi

Department of Aeronautics and Astronautics Engineering, University of Southampton, UK.
E-mail: tairan.wang; sifeng.bi@soton.ac.uk

Imre Gal, Sebastiano Fichera, John Mottershead

Department of Mechanical and Aerospace Engineering, University of Liverpool, UK.
E-mail: i.gal; seba84; j.e.mottershead@liverpool.ac.uk

Uncertainty quantification (UQ) remains critical in addressing complex engineering challenges, especially in safety-critical systems where scarce data and mixed uncertainties prevent robust decision making. This paper presents a data-driven model updating framework to address the NASA and DNV Challenge 2025 on Optimisation under Uncertainty using the invertible normalising flow-based neural networks, which emphasises high-dimensional systems with limited observational data and hybrid aleatory-epistemic uncertainties. Our methodology explicitly deals with the aleatory and epistemic uncertainties separately through a two-step model updating framework based on a preliminary sensitivity analysis. The aleatory variables are calibrated first globally and then the epistemic variables are calibrated locally. To process the time series response data, multihead transformer is adopted as the conditional network in the normalising flow-based model updating framework, which can summarise the complex data into fixed-length vector. The following design optimisation problems are tackled by the Particle Swarm Optimisation (PSO) with a Fully Connected Neural Networks (FCNNs)-based surrogate model. This work bridges machine learning with classical UQ methodologies, offering a practical pathway for safety-critical system design under aleatory-epistemic uncertainties.
Keywords: uncertainty quantification, model updating, design optimisation, normalising flow.

1. Introduction & Problem Description

The NASA and DNV Challenge on Optimisation Under Uncertainty (Agrell et al. 2025) is a challenge problem presented by both NASA Langley Research Centre and DNV Group Research and Development for researchers to explore the shared challenge in UQ across various sectors such as the practical development of safety-critical vehicle systems. The Challenge 2025 can be regarded as the enhanced version of the previous NASA Multidisciplinary Uncertainty Quantification Challenge 2014 (Crespo and Kenny 2019) and the NASA Langley UQ Challenge on Optimisation under Uncertainty (Crespo, Kenny, and Giesy 2014). All the three challenges share the similar key aspects of model calibration and uncertainty reduction given sparse data, and reliability-based design in the presence of both aleatory and epistemic uncertainty.

The problem framework centers on a black-box physical system characterized by an 8-dimensional input vector $X = (X_a, X_e, X_c)$, where:

- X_a (2 parameters) refers to aleatory variables with unspecified probability distribution f_a
- X_e (3 parameters) denotes epistemic variables with unknown true values X_e^*
- X_c (3 parameters) constitutes control variables with fixed user-defined values

All the three categories of parameters are standardised into a relevant range in $X \in [0,1]$. The system response comprises six time series $\left[Y_1^{(i)}(t), Y_2^{(i)}(t), Y_3^{(i)}(t), Y_4^{(i)}(t), Y_5^{(i)}(t), Y_6^{(i)}(t)\right]$ sampled at 60 time points ($i = 1,2,\cdots,60$), forming a 60×6 multivariate time series. As illustrated in Fig 1, the response depends on both deterministic inputs and stochastic elements governed by a random seed s, effectively rendering the output a stochastic process even under fixed input conditions.

Numerical simulations require simultaneous specification of X and seed s, whereas physical system observations are obtained by setting only

the control variables X_c. In the latter case, X_a values are automatically sampled from f_a, X_e assumes its true (but unknown) value X_e^*, and s follows its inherent probability distribution. This dual modeling framework creates distinct data generation paradigms for computational and experimental scenarios, establishing a realistic testbed for uncertainty quantification methodologies.

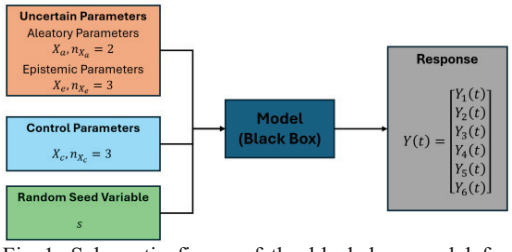


Fig 1. Schematic figure of the black box model for NASA UQ 2025 Challenge.

The NASA UQ Challenge 2025 consists of two categories of problems, namely: 1) uncertainty quantification and 2) design optimisation, which involve a series of subproblems. In the first problem about uncertainty quantification, the uncertainty models (UMs) are initialised by the given range of the input parameters. The initial UMs are then calibrated in two steps given the observation data from the real system. In the second problem about design optimisation, a cost function and another constraint are determined for optimisation. A series of optimisation process are carried out to define different X_c^* about performance-based design, reliability-based design and the constrained design.

2. Methodology Overview

The goal of Problem 1 is to obtain the UMs of the aleatory variables and the epistemic variables against the response data from the real system. Thus, the calibrated UMs will constitute a two-dimensional probability distribution f_a for X_a and 3 reduced intervals E for X_e. Because the response data is six-dimensional time series, the traditional model updating approaches such as the Bayesian approach might not be compatible or might need to be integrated with other data pre-processing methods in order to build up the likelihood function (Bi et al. 2022).

In this study, a novel conditional normalising flow (CNF)-based model updating framework is employed. The conditional normalising flow is a type of flow-based deep generative model consists of a conditional network and an invertible neural network (INN). The conditional network is responsible for summarise the observations into fixed size condition vectors and inject them into the INN and the INN is accountable for bijective transformation between complex input space and latent space (generally standard Gaussian distribution) with the guidance of the conditions.

The CNF-based model updating framework (as shown in Fig. 2) establishes a bidirectional mapping between input parameter distributions and latent representations through an integrated architecture combining an Invertible Neural Network (INN) with a conditional feed-forward network (Ardizzone et al. 2019). During forward training, input parameters sampled from prior distributions are processed through numerical simulations to generate corresponding outputs, which the conditional network encodes into transformation parameters for the INN. This enables the INN to learn a bijective mapping between the input parameter space and a Gaussian latent space while preserving simulation data relationships through conditional coupling. In the inverse inference phase, the trained network probabilistically inverts this mapping by propagating latent samples through the inverted INN conditioned on observed data features, thereby recovering posterior parameter distributions that synthesize prior knowledge with observational evidence. This dual-directional framework achieves efficient model updating bypassing the likelihood estimation in Bayesian model updating, simultaneously maintaining computational tractability and preserving complex parameter-data interdependencies learned during joint training.

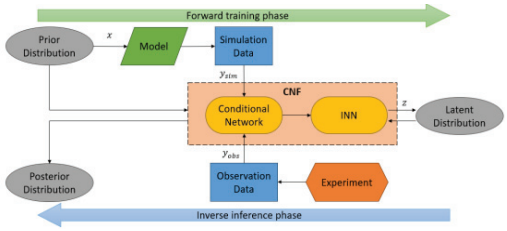


Fig 2. Architecture of CNF-based model updating framework.

The so-called conditional affine coupling layers (cACLs) (Ardizzone et al. 2019) serves as fundamental units in the architecture of the CNFs, as shown in Fig 3. The input vector is split into

two parts with random size and propagated through the cACL to the latent space. The conditional network is integrated in the cACL to assure the information of observation data is included in the training process and can guide the inverse generation process. The $s_{1,2}$ and $t_{1,2}$ indicate scaling and translation functions, which can be arbitrary functions or neural networks.

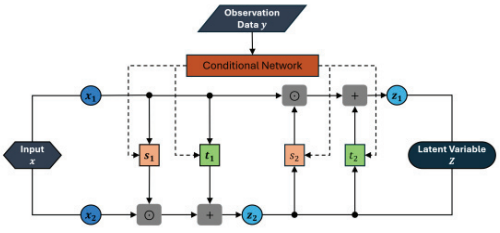


Fig 3. Conditional affine coupling layers (cACLs) in the forward training direction.

3. Calibration of the Uncertainty Models (Problem 1)

3.1. *Sensitivity Analysis*

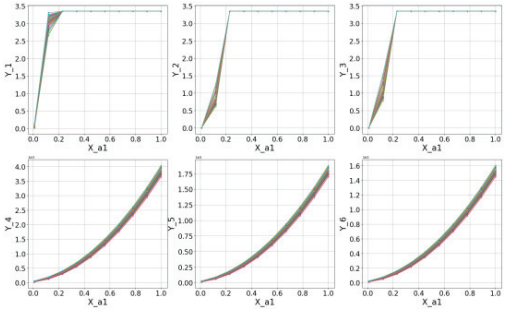
Before carrying out the calibration process, a sensitivity analysis was performed to observe how the output responses vary based on different input parameters including $[X_{a_1}, X_{a_2}, X_{e_1}, X_{e_2}, X_{e_3}, X_{c_1}, X_{c_2}, X_{c_3}, s]$. By taking the initial input vector of the black box model as $X = [0.1,0.2,0.3,0.4,0.5,0.533,0.666,0.5,10]$, where the first two parameters refer to the aleatory parameters, the following three refer to the epistemic parameters, the three after refer to the control parameters and the random seed parameter at last, one input parameter is adjusted at a time and run the forward black box model provided to collect the corresponding output responses to discover any changes caused by the variation of the input. The input parameters X are adjusted from 0 to 1 with a step length of 0.1 and the random seed variable s is selected as integers from 1 to 10. The different plots of various colours in Fig 4 represent response data of the 60 time-steps.

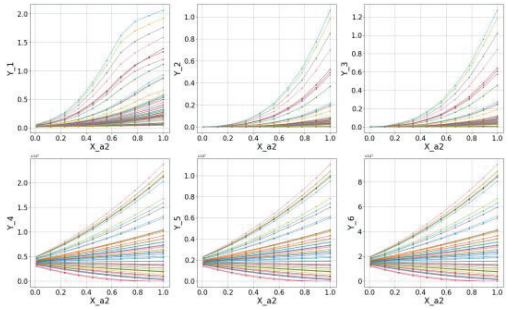
The results indicate that, the aleatory parameters have significant impact on response data. For the first aleatory variable, X_{a_1}, all six output features are affected strongly by this parameter. It causes an increase from 0 to around 3.35 in first three response features and an increase from 0 to 4000, 1750, and 3400 to the latter three response features, respectively. The first three responses reach the upper bound of around 3.5 when X_{a_1} reaches roughly 0.3 or larger. For the second aleatory variable, X_{a_2}, the first three responses on all time steps increase as the increase of the X_{a_2}, while the response data on some time steps of the latter three responses decrease along the increase of X_{a_2}.

The epistemic parameters, on the other hand, have impact on response data but is considerably smaller than the aleatory parameters. All three epistemic parameters have almost no effect on the response $Y_1(t)$ and $Y_4(t)$. The first two epistemic parameters, X_{e_1} and X_{e_2}, have little effect on the other 4 responses, while X_{e_3} a negligible effect on the other 4 responses either.

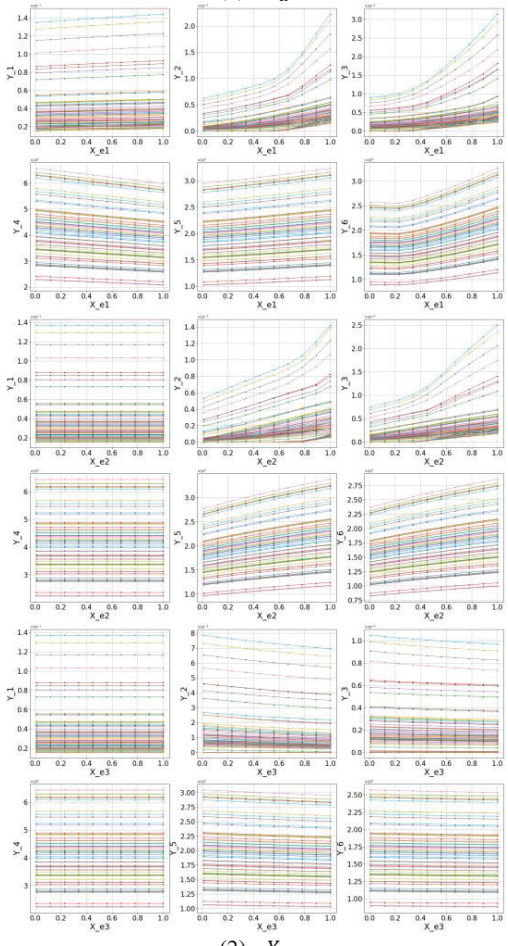
For the control variables, unlike the other parameters lead to a trend of either increase or decrease monotonically on the response data, they will make the response differently incremental and decremental before and after a certain point. As the X_{c_1} changes it causes $Y_1(t)$ increase first and then decrease and the other 5 responses decrease and later increase, where the inflection point occurs at approximately $X_{c_1} = 0.4$. The second control variable has similar impact on $Y_2(t), Y_3(t), Y_5(t), Y_6(t)$, where the inflection point occurs at roughly $X_{c_2} = 0.6$. However, it barely has effect on $Y_1(t)$ and $Y_4(t)$. The X_{c_3} has monotonical impact on $Y_2(t), Y_3(t), Y_5(t), Y_6(t)$, with $Y_2(t)$, $Y_5(5)$ increase and $Y_3(t)$, $Y_6(t)$ decrease along the increase of X_{c_3}. It nearly has no effect on $Y_1(t)$ and $Y_4(t)$ either.

The random seed parameter s serve as a random noise generator that add extra noise on the model response. Its variation will have some effect on the response, but there is no obvious regularity.

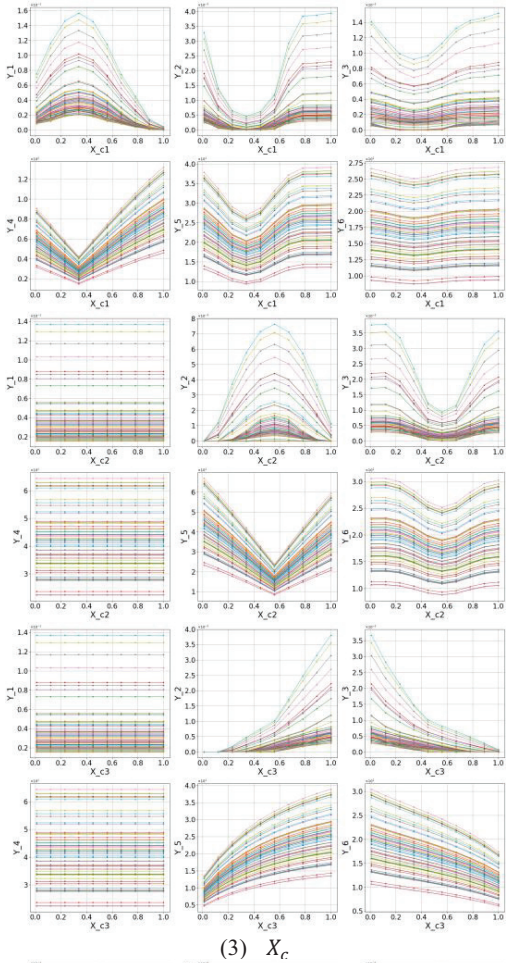




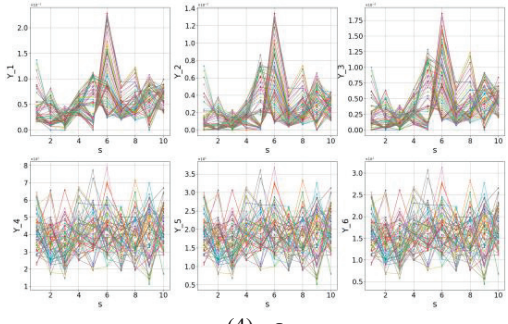
(1) X_a



(2) X_e



(3) X_c



(4) s

Fig. 4. Sensitivity analysis for the input parameters (1) X_a, (2) X_e, (3) X_c and (4) s.

3.2. *Observation Data Acquisition*

To obtain the response data from the real system, which serves as the target in the model updating process, the three control variables need to be determined manually. In this case, a total number of 10 groups of control variables are selected

based on the sensitivity analysis above. For each of the three control variables, 10 values are chosen based on the sensitivity analysis above. For X_{c_1} and X_{c_2}, the samples are selected symmetrically of the inflection points, and the intensity of sampling is determined by the variation rate of the response. Thus, the samples result in vectors X_{c_1}= [0.075, 0.15, 0.225, 0.35, 0.4, 0.45, 0.575, 0.65, 0.725, 0.9],X_{c_2}= [0.1, 0.275, 0.35, 0.425, 0.55, 0.6, 0.65, 0.775, 0.85, 0.95]. The X_{c_3} are sampled uniformly as X_{c_3}= [0.05, 0.15, 0.25, 0.35, 0.45, 0.55, 0.65, 0.75, 0.85, 0.95]. The Latin Hypercube Sampling is adopted to determine the final 10 combinations according to the sampled vectors. The response data is generated accordingly by entering all the chosen control variables to the system provided by the organisers and 100 sets of 6-dimensional time series data are generated based on one single set of control variables.

Table 1. Sampled 10 groups of control variables for observation data acquisition.

Number	X_{c_1}	X_{c_2}	X_{c_3}
1	0.225	0.1	0.05
2	0.65	0.775	0.65
3	0.9	0.425	0.15
4	0.725	0.275	0.75
5	0.15	0.95	0.85
6	0.075	0.6	0.25
7	0.4	0.55	0.95
8	0.45	0.35	0.45
9	0.35	0.65	0.55
10	0.575	0.85	0.35

3.3. *Model Updating (Problem 1)*

Based on the understanding of the problem, a two-step model updating is carried out with the aleatory parameters to be calibrated firstly and the epistemic parameters to be calibrated afterward. The initial UMs for the two aleatory parameters are set as uniform distributions from 0 to 1, and the epistemic parameters are set as intervals from 0 to 1. The two UMs for the aleatory parameters are first updated with the initial UMs as prior and given the real response data from the real system as the target. The three UMs for the epistemic parameters are then calibrated by setting the prior distribution follow the updated UMs in the first step and given the real response data from the real system as the target.

The CNF architecture is designed to contain 6 cACLs with the multihead transformer as the conditional network. The multihead transformer (Vaswani et al. 2017) architecture processes 60-time-step sequences of 6-dimensional inputs through two encoder layers, each featuring a 4-head self-attention mechanism and a feedforward network with a size of 256, with model dimensions unified to 64. Residual connections, layer normalisation, and 0.1 dropout maintain stability while capturing hierarchical patterns through scaled attention weights and nonlinear feature transformations.

In the training phase of the first step of model updating, the training data is generated by using the local computational model provided by the organisers. The aleatory variables and the epistemic variables are sampled from their initial UM introduced before. The control variables are sampled randomly from the interval from 0 to 1 as well. The random seed parameter is fixed because the involvement of extra noise may affect the training of the CNF. A number of 30,000 sets of training data were collected. After training, the networks are optimised automatically so that they can build a bijective mapping between the input and latent variables accurately.

In the inference phase of the first step of model updating, 10 groups of obtained observation data are concatenated together as the conditional data. The well-trained CNF can operate inversely to infer the distribution of input parameters by sampling randomly from the latent distribution with given observation data as the condition. There are 1000 observation data utilised in the model updating process, implying 10,000 samples of the aleatory parameters are obtained via the inverse application of the CNF architecture, where 10 samples are generated inversely based on a single set of observations. The PDFs of the aleatory parameter UMs, f_a, are estimated and illustrated in Fig 5.

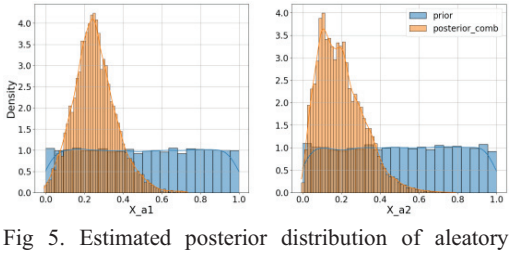


Fig 5. Estimated posterior distribution of aleatory variables X_a.

In the second step of the model updating, the sampling strategy of the input parameters to obtain the training dataset is slightly different from the last step. The aleatory variables are systematically drawn from the posterior distribution obtained in the last step, while control variables remain fixed according to the predefined combinations outlined in Section 3.2. All other model parameters maintain their initial configuration from the first updating step. Notably, distinct training datasets are generated for each unique combination of control variables, resulting in a comprehensive collection of 10,000 training data instances. Therefore, 10 rounds of model updating according to various training data and combinations of control variables are carried out to obtain 10 sets of various posterior distributions of epistemic variables.

For each observational dataset, the inference process is executed under the constraint of identical control variable configurations used during corresponding training phases. The optimised CNF generates posterior distributions for the three epistemic parameters through inverse propagation, producing 100 samples per observational dataset. This step results in 10,000 posterior samples, enabling robust estimation of probability density functions as demonstrated in Fig 6.

In this case, three groups of results with good performance (the predictions are in good agreement with the true value in the validation) by adopting control variables from No. 2, 7, and 9 in Table 1 are plotted in Fig 6 below. The resultant posterior distributions from these selected configurations are demonstrated in Fig 6 to illustrate differential inference outcomes.

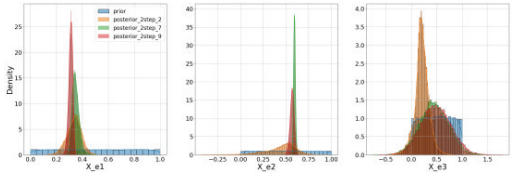


Fig 6. Estimated posterior distribution of epistemic variables X_e.

According to the two-step model updating results, the calibrated UMs for the aleatory variables and epistemic variables are determined. The aleatory variables follow a distribution f_a and their PDFs are shown in Fig 5. The intervals for the epistemic variables are determined manually by balancing model updating results to make sure the most probable value is in the interval and the interval is narrow but informative. The estimated intervals are denoted in Table 2 below with the best guess of epistemic variables.

Table 2. Calibrated uncertain models (E) of epistemic variables X_e.

Parameters	Uncertain Models	Best Guess
X_{e_1}	$X_{e_1} \in [0.26,0.41]$	0.323
X_{e_2}	$X_{e_2} \in [0.5,0.6]$	0.571
X_{e_3}	$X_{e_3} \in [0,0.5]$	0.223

3.4. *Prediction Intervals Determination (Problem 1.3)*

The overall goal for this subproblem is to determine the tightest prediction intervals of all six outputs corresponding to the identified uncertainty models before for a baseline design X_c, where the baseline design is set as $X_c = [0.533,0.666,0.5]$. The aleatory variables and the epistemic variables are generated randomly from the UMs obtained in the last subproblem. For all X_e in the interval of E, X_a are generated from their uncertainty model $X_a \sim f_a$. A number of 50,000 sets of response data are generated by sampling input parameters from their UMs using MCS in each case. The corresponding response data are collected and flattened to six-dimensional arrays that are sorted from smallest to largest without considering the time sequence and dimension of response. The infimum of the upper bound and the supremum of the lower bound of all six-dimensional response data are evaluated based on these data. The obtained results for both $\alpha = 0.999$ and $\alpha = 0.95$ are indicated in Table 3.

Table 3. Predicted intervals of all response data.

Output Dimension	Scenario	Infimum of upper bound	Supremum of lower bound
1	$\alpha = 0.999$	3.35	0
	$\alpha = 0.95$	3.35	0
2	$\alpha = 0.999$	3.35	0
	$\alpha = 0.95$	3.35	0
3	$\alpha = 0.999$	3.35	0
	$\alpha = 0.95$	3.35	0
4	$\alpha = 0.999$	1841.96	0
	$\alpha = 0.95$	722.01	9.61
5	$\alpha = 0.999$	872.46	0
	$\alpha = 0.95$	341.41	4.51
6	$\alpha = 0.999$	760.01	0

$\alpha = 0.95$	296.82	3.93

4. Design Optimisation (Problem 2)

To execute optimisation for the black box model directly will lead to a massive number of model evaluations during the iterations which will cause the optimisation computationally expensive and time intensive. Therefore, a surrogate model is built based on the knowledge of the system. Distinct from general surrogate modelling tasks, the output features in this term are time sequences, which is considered the challenge.

In this case, a fully connected neural network (FCNN) with two hidden layers is adopted as the surrogate. The output features of different time steps are concerned separately. Thus, one single surrogate is built between the inputs and the outputs for one single time step and result in 60 surrogate models in total. The FCNN surrogate model is very time efficient compared to the original black model, but the accuracy is cursed by the complex correlation between the input and output with a $R^2 = 0.91$ and relatively poor generalisation capability. Consequently, the performance of the following optimisation process is directly influenced by the precision of the surrogate model.

4.1. *Performance-based Design (Problem 2.1)*

The objective of this subproblem is to obtain a group of performance-based design of X_c^* that can maximise the objective function below.

$$X_c^* = \arg\max_{X_c} \left(\min_{X_e \in E} \int_0^1 \sum_{i \in I_1} \mathbb{E}[y_i(X_a, X_e, X_c, s, t)] \right) dt \quad (1)$$

where $\mathbb{E}[\cdot]$ refers to the expected value operator with respect to random processes X_a and s.

In this case, the optimisation is carried out in the output space I_1, which is considered as $I_1 = \{i | i = 1,2,3\}$. In Eq (1), the optimisation problem is described as double-loop optimisation, including determining the X_e that minimise the integral in the time domain of the sum in the output space of the expectation of the i-th response given a certain X_c. Then, the control variables X_c are optimised maximising the minimum mentioned above. The first step of optimisation is simplified with numerical model evaluations based on input parameters X_a and X_e sampled from an interval of $[0,1]$ and E using MCS. Afterward, Particle Swarm Optimisation (PSO) is adopted to prescribe the optimal value X_c^*, iteratively. The captured performance-based design is $X_c^* = [0.683, 0.377, 0.692]$.

4.2. *Reliability-based Design (Problem 2.2)*

To obtain the reliability-based design X_c^*, the greatest value of system failure probability expressed in Eq (2) is minimised as X_e varies in E.

$$\text{pof}_{\text{sys}}(X_e, X_c) = \mathbb{P}\left[\min_{i \in I_2} g_i(X_a, X_e, X_c, s) < 0\right] \quad (2)$$

where $\mathbb{P}[\cdot]$ is the probability with respect to the random processes X_a and s; and the event $g_i(X, s) < 0$ expressed in Eq (3) refers to a failure in the i-th physical response and the individual failure probability are defined as shown in Eq (4).

$$g_i(X, s) = c_i - \max_{0 \le t \le 1} |y_i(X, s, t)|, \forall i \in I_2 \quad (3)$$

$$\text{pof}_i(X_e, X_c) = \mathbb{P}[g_i(X_a, X_e, X_c, s) < 0] \quad (4)$$

where the c_i are fixed constants.

The output space I_2 is considered as $I_2 = \{i | i = 4,5,6\}$. The constants c_i are defined as $c_4 = 2750, c_5 = 2000, c_6 = 1000$. The probability of failure of each dimension is determined by computing the probability of the maximum response among all time steps of each dimension smaller than zero while X_e varies in E and X_a randomly generated. The system failure probability is determined by computing the probability of the minimum of the maximum response among all time steps of all dimensions smaller than zero. The PSO is adopted to solve the problem and results in a reliability-based design $X_c^* = [0.695, 0.381, 0.629]$.

4.3. *ϵ-constrained Design (Problem 2.3)*

This subproblem describes a constrained optimisation problem that solves Problem 2.1 in the output space I_2 while regarding the constraint in Eq (5).

$$\max_{X_e \in E} \text{pof}_{sys}(X_e, X_c) \le \epsilon \quad (5)$$

where ϵ represents the failure probability and $\epsilon = 10^{-3}$ and $\epsilon = 10^{-4}$ are considered.

The optimisation is carried out the same way as in Sec 4.1 while ensuring that the maximum system failure probability is no larger than the constraint ϵ. The constrained PSO algorithm returns the results for both scenarios, as shown in Table 4.

Table 4. ϵ-constrained design of X_c.

Scenario	X_c^*
$\epsilon = 10^{-3}$	[0.685.0.378,0.674]
$\epsilon = 10^{-4}$	[0.686,0.374,0.681]

5. Conclusion

In this paper, the NASA and DNV Challenge on Optimisation Under Uncertainty 2025 is successfully tackled and the results are presented accordingly. The first problem of the challenge regarding uncertainty quantification is solved by a two-step data-driven model updating framework based on the CNFs, which allows calibrating the aleatory and epistemic parameters separately. In the process of model updating, the aleatory variables are calibrated primarily to obtain their UMs f_a and the epistemic variables are updated subsequently based on the f_a to obtain their UMs E. In order to process the time-series response data effectively, the multihead transformer is employed as the conditional network of the CNF. On the basis of a preliminary sensitivity analysis, 10 groups of control variables are selected for observation data generation to ensure that a full scale of response data is included and the model updating framework is built accordingly. In addition, the CNF-based model updating can operate based on observation data of any size even though its performance might be affected.

In the second problem regarding design optimisation, a surrogate model of the original black box model is established to replace the time-consuming model evaluation. The inner optimisation of epistemic variables is simplified by MCS to obtain the ergodic of epistemic variables in their UMs. The Particle Swarm Optimisation (PSO) is adopted to capture the performance-based, reliability-based, and ϵ - constrained design of the control variables. However, the performance of the optimisation process is limited by the accuracy of the surrogate model. The establishment of an accurate and efficient surrogate model for a real system with time series response remains a challenge.

References

Agrell, Christian, Luis G Crespo, Vegard Flovik, Erik Vanem, and Sean Kenny. 2025. 'The NASA and DNV Challenge on Optimization Under Uncertainty'.

Ardizzone, Lynton, Carsten Lüth, Jakob Kruse, Carsten Rother, and Ullrich Köthe. 2019. 'Guided Image Generation with Conditional Invertible Neural Networks'. arXiv. http://arxiv.org/abs/1907.02392.

Bi, Sifeng, Kui He, Yanlin Zhao, David Moens, Michael Beer, and Jingrui Zhang. 2022. 'Towards the NASA UQ Challenge 2019: Systematically Forward and Inverse Approaches for Uncertainty Propagation and Quantification'. *Mechanical Systems and Signal Processing* 165 (February):108387. https://doi.org/10.1016/j.ymssp.2021.108387.

Crespo, Luis G, and Sean P Kenny. 2019. 'The NASA Langley Challenge on Optimization Under Uncertainty'.

Crespo, Luis G., Sean P. Kenny, and Daniel P. Giesy. 2014. 'The NASA Langley Multidisciplinary Uncertainty Quantification Challenge'. In . Reston, VA, United States. https://ntrs.nasa.gov/citations/20140007349.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. 'Attention Is All You Need'. In *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.