Generating High-Resolution Flight Parameters in Structural Digital Twins using Deep Learning-based Upsampling

Xuan Zhou[1]
Department of Mechanical Engineering
Politecnico di Milano
Milano, Italy
xuan1.zhou@mail.polimi.it

Michal Dziendzikowski
Airworthiness Division
Instytut Techniczny Wojsk Lotniczych
Warszawa, Poland
michal.dziendzikowski@itwl.pl

Krzysztof Dragan
Airworthiness Division
Instytut Techniczny Wojsk Lotniczych
Warszawa, Poland
krzysztof.dragan@itwl.pl

Leiting Dong[2]
School of Aeronautic Science and Engineering
Beihang University
Beijing, China
ltdong@buaa.edu.cn

Marco Giglio
Department of Mechanical Engineering
Politecnico di Milano
Milano, Italy
marco.giglio@polimi.it

Claudio Sbarufatti[3]
Department of Mechanical Engineering
Politecnico di Milano
Milano, Italy
claudio.sbarufatti@polimi.it

***Abstract*—The structural digital twin is a virtual representation of physical entities that accurately predicts the evolution of structural damage through multidisciplinary and multi-level probabilistic simulations. It provides crucial support for prognostic and health management. Flight parameters are important input data for airframe digital twin to support aerodynamic and structural simulations. However, many small aircraft or UAVs often suffer from insufficient sampling rates of flight parameters due to cost limitation or premature service. In this study, we propose a deep learning-based flight data upsampling method that effvbectively enhances the resolution of flight data. The method constructs an upsampling model using a one-dimensional super-resolution convolutional residual network, defines multiple loss functions associated with the flight data, and uses a highly sampled test aircraft dataset for training. The proposed method is validated using real UAV flight test data and several criteria, achieving good results with different upsampling factors. This approach is expected to facilitate the construction of structural digital twins in the future.**

Keywords—digital twin, flight parameter, upsampling, deep learning, prognostics and health management

I. INTRODUCTION

Digital twin has gained significant attention in the field of prognostics and health management, as it provides a virtual representation of physical assets [1], [2]. Originating from the aerospace field, a digital twin facilitates prognosis and health management by creating a multi-physical, multi-scale, and probabilistic virtual model of the system [3]. This model integrates multiple heterogeneous and uncertain information sources from models and data to support proactive fleet maintenance decisions [4]–[8]. For the Airframe digital twin, flight data plays a crucial role, as it provides flight conditions and serves as input to conduct aerodynamics and structural simulations [5].

However, a significant number of small aircraft or UAVs encounter limitations in the sampling rates of flight parameters due to cost constraints or premature entry into service [9]. For instance, the flight data captured by the Garmin 1000 avionics system in Cessna 172 operates at a mere 1 Hz, making it unsuitable for simulating the damage growth induced from flight aerodynamic loads.

To tackle this challenge, upsampling flight parameters presents a feasible solution. Nevertheless, limited research has been conducted in this area, with a primary focus on physical-based approaches. Öström et al. [10] proposed an inverse simulation method as a reliable means of completing coarse-resolution data before feeding it into a neural network that generates stress time histories of specific structural details. However, this approach necessitates the creation of higher fidelity flight dynamics models, which are frequently unavailable for small aircraft or UAVs. In such circumstances, data-driven approaches can play a vital role.

[1] He is also a Ph.D. candidate at the School of Aeronautic Science and Engineering, Beihang University.
[2] Co-corresponding author
[3] Corresponding author

This work has been developed based on the results from SAMAS project (SHM application to Remotely Piloted Aircraft Systems), a Cat.B project coordinated by the European Defense Agency (EDA) and financed by two nations, Italy and Poland. The project consortium includes the following parties: Italy (Politecnico di Milano, Leonardo S.p.A) and Poland (Instytut Techniczny Wojsk Lotniczych -AFIT,Military Aviation Works No. 1).

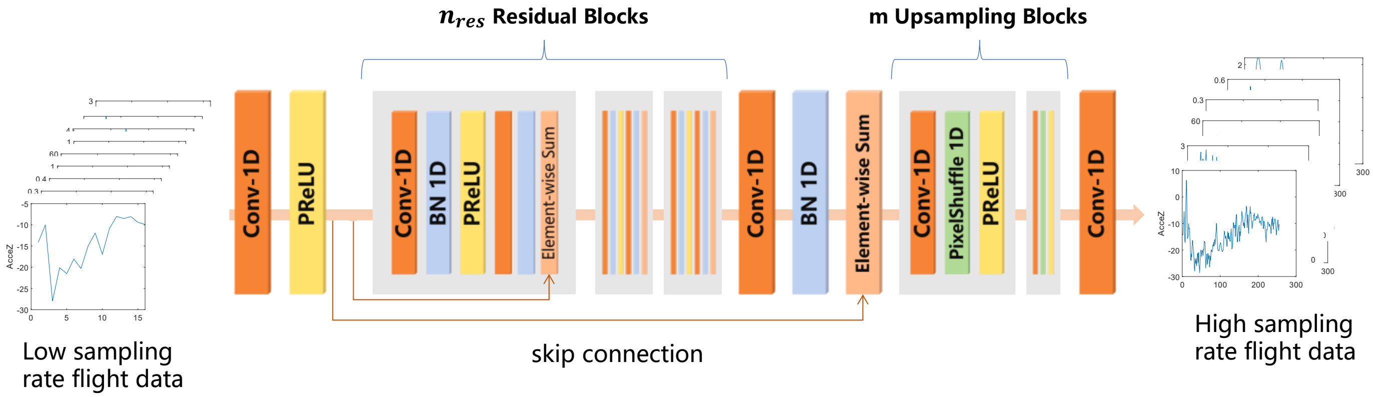


Fig. 1. Network Structure

The data-driven upsampling approach has found extensive use in the field of computer vision, with numerous methods proposed to improve the resolution of images and videos. These include convolutional residual network [11], generative adversarial networks [12], and recently popular diffusion models [13]. In the engineering domain, upsampling methods are also employed in non-intrusive load monitoring [14] and turbulent flows reconstruction [15], among others.

In this paper, we introduce a novel upsampling technique for flight parameters utilizing a deep convolutional neural network. The proposed method takes low-sampling-rate flight data as input, extracts features through a convolutional neural network, and generates high-sampling-rate data by the one-dimensional pixel shuffle block. Multiple loss functions are taken into account during the training process, and validation is performed using flight test data acquired from an unmanned aerial vehicle (UAV).

II. Upsampling by the Deep Learning-based Approach

Super-resolution algorithms have their origins in the field of image processing, where they are used to enhance the resolution of 2-dimensional images. In this study, we adapt and modify it to facilitate the upsampling of flight parameters as a time-series problem.

A. *Preprocessing of the flight data to create the dataset*

The test flight data acquired by the aircraft from various sensors possess varying sampling rates and thus cannot be utilized directly for the upsampling process. To input the data into the model, all the data must be converted into data with the same sampling rate. Subsequently, to account for high-frequency noise, all the data is resampled to the same rate.

By utilizing a fixed window of length n_w to traverse through the entire flight history, several high sampling rate flight sequences s_h with a length of n_{ph} can be obtained. To ensure a uniform representation of data with different units and ranges, the minmax transformation is applied to each sequence.

The low sampling rate sequence s_l, which serves as the input of the model, is generated through the down-sampling of s_h. The length of s_l is calculated as $n_{pl} = n_{ph}/\alpha$, where α represents the upsampling scale factor.

B. *Network Structure and setting*

In this study, the network architecture is adapted from the super resolution residual network, with all the blocks replaced by their 1D equivalents to better reflect the characteristics of flight parameters.

The input data consists of low-sampling flight and environment data, with the environment data serving as prior knowledge. Initially, the low-sampling data is input into a conv1d layer with a PReLU activation function. Subsequently, B residual blocks are stacked to extract additional features. Finally, m upsampling blocks are utilized, with the value of m being determined by the scale factor:

$$m = log_2\alpha \tag{1}$$

where m is the number of upsampling blocks and α is the upsampling scale factor.

The kernel size of the 1D convolutional layer in both the residual blocks and the upsampling blocks is defined as n_{sk}, while the kernel size of other convolutional layers is defined as n_{lk}. The number of residual blocks is defined as n_{res}, and the number of channels in the CNN is defined as n_c.

C. *Loss function*

Let θ be the parameter of the neural network. To determine an optimal θ, three losses are defined, which are Mean square error loss l_{mse}, down sampling loss l_{ds} and Fast Fourier Transform Error (FFTE) loss l_{fft}, respectively. Thus, the total loss L is defined by

$$\mathrm{L} = l_{mse} + \lambda_{ds} l_{ds} + \lambda_{fft} l_{fft} \tag{2}$$

where λ_{ds} is the weight of the l_{ds}, and λ_{fft} is the weight of l_{fft}.

1) *Mean square error loss*

The mean square error loss is a commonly used loss function and is defined as follows:

$$l_{mse} = \frac{1}{n_{ph}} \sum_{i=1}^{n_{ph}} (y_i - \hat{y}_i)^2 \tag{3}$$

where n_{ph} represents the length of the high sampling rate sequence. $\hat{\boldsymbol{y}}_i$ denotes the upsampled sequence obtained from the model, and $\boldsymbol{y}_i$ represents the true high sampling rate sequence.

2) *Down sampling loss*

To ensure the consistency of the resulting sequence with the original low-sampling-rate points, we define the down-sampling loss, which can be viewed as an additional weight applied to the low-sampling-rate points. The down-sampling loss is defined as:

$$l_{ds} = \frac{1}{n_{pl}} \sum_{i=1}^{n_{pl}} \left(y_i^l - \hat{y}_i^l \right)^2 \tag{4}$$

where $\hat{y}_i^l$ and y_i^l are the predicted and real value at the point of low sampling rate.

3) Fast Fourier Transform loss

The Fast Fourier Transform (FFT) loss function, as used in [14], is utilized to measure the dissimilarity of the sequence in the frequency domain. It is more focused on capturing the variations in the overall shape of the sequence, as opposed to the MSE loss function.

$$Fl_{fft} = \|\mathcal{F}(\hat{\boldsymbol{y}}_i) - \mathcal{F}(\boldsymbol{y})\|_1 \tag{5}$$

where $\mathcal{F}$ is the fast Fourier transform operator.

III. Hornet UAV and its Flight Test

In this study, two flight tests of a UAV are utilized to demonstrate the proposed approach.

A. Hornet

The deep learning-based upsampling approach described in Section II is tested on a full-scale Unmanned Aerial Vehicle (UAV) used as an aerial target during army training sessions of the Polish Air Force, namely the Hornet [16], [17]. The UAV, shown in Fig. 2, is a modified version of the originally developed system, designed and produced by the Air Force Institute of Technology (AFIT) and the Military Aviation Works number 1 (MAW1). Further specifications are accounted for in Table 1.

TABLE I. UAV technical specifications

Parameter	**Value**
Length	1.7 m
Wing span	3.2 m
Starting weight	38 kg
Fuel tank capacity	8 L (6.3 kg)
Payload	5 kg
Min–cruise–max speed	85 − 150 − 230 km/h
Operating range	40 km
Maximum climbing speed	16 m/s (at 140 km/h)

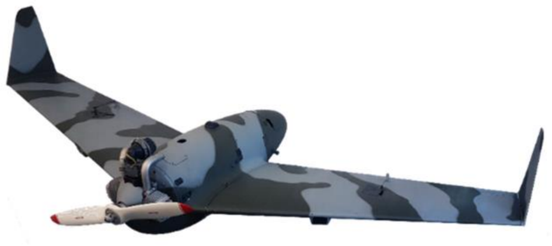


Fig. 2. The Hornet UAV

Fig. 3. The flight test of the Hornet UAV.

B. Flight Tests

In this study, two flight tests conducted on the same day but at different times were utilized to illustrate the proposed approach. The tests were carried out under thermic conditions with moderate wind, and each flight lasted for approximately 15 minutes, making some of the maneuvers more challenging to execute accurately.

C. Flight Parameters Preprocessing and Network Hypermeter Setting

In this study, 12 types of flight and environmental parameters are available for the data acquisition unit. All data are normalized to the range of [0,1]. The acceleration and angular velocity data are acquired at a frequency of 50 Hz, while other parameters are acquired at 10 Hz. To better illustrate the proposed upsampling approach and reduce the effect of noise, all data are resampled to 16 Hz.

Flight and Environment Parameters in the Hornet UAV

TABLE II.

Parameter	***Type***	***Unit***	***Frequency***
GyrX, GyrX, GyrZ	Rotation rate about X, Y, Z axis	deg/s	50 Hz
AccX, AccY, AccZ	Acceleration along X, Y, Z axis	m/s^2	50 Hz
IAS	Indicated Air Speed	m/s	10 Hz
AOA	Angle of Attack	deg	10 Hz
SSA	Angle of Sideslip	deg	10 Hz
g-load	Vertical Load Factor	1	50 Hz
PRESSURE	Atmospheric pressure	MPa	10 Hz
Alt	Attitude	m	10 Hz

The data from the two flight tests were preprocessed using the procedure outlined in Section II.A. A fixed window size of $n_w = 10$ was used, resulting in 1171 sequences for the first flight and 1263 sequences for the second flight. The data from the first flight was divided into a training set, validation set, and test set with a ratio of 0.6: 0.2: 0.2, respectively.

TABLE III. MODEL HYPERMETERS FOR THE FLIGHT DATA UPSAMPLING OF THE HORNET UAV

Hypermeter	**Value**
n_{lk}	9
n_{sk}	3
n_{res}	8
n_{cnl}	128
lr	1e-4
Batch size	32
λ_{ds}	0.1
λ_{ffte}	0.01

The model takes all 12 parameters as input and outputs the first 10 parameters without PRESSURE and Alt. Three types of upsampling with scale factors of α = 4,8,16 were performed. Notably, the $\alpha = 16$ case was used to simulate a scenario where only 1Hz data is available.

D. Baseline and Performance Metric

To measure the performance of the proposed approach, two metrics are adopted in this study. The first metric is the root mean square error (RMSE), which measures the point-by-point error between the upsampled data and the high sampling rate data. The RMSE is defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n_p}\sum_{i=1}^{n_p}(\boldsymbol{y}_i - \hat{\boldsymbol{y}}_i)^2} \tag{6}$$

where n_{ph} is the length of the high sampling rate sequence, $\hat{\boldsymbol{y}}_i$ is the upsampled sequence from the model, and $\boldsymbol{y}_i$ is the true high sampling rate sequence.

The second metric used to evaluate the performance of the proposed approach is the Fast Fourier Transform Error (FFTE), which measures the error in the frequency domain of the upsampled sequence compared to the high sampling rate sequence.

$$FFTE = \frac{1}{N}\left\|\mathcal{F}(\hat{\boldsymbol{y}}_i) - \mathcal{F}(\boldsymbol{y})\right\|_1 \tag{7}$$

where $\mathcal{F}$ is the fast Fourier transform operator.

The baseline method employed in this study is a simple linear interpolation technique.

IV. RESULTS AND DISCUSSION

This section presents the upsampling results of the Hornet UAV flight data using the proposed approach, and compares them with the baseline method. Two test sets are used for evaluation: the first one is the test set obtained by splitting the data from the first flight test, while the second one consists of the entire data from the second flight test.

A. Training process

Fig. 4 illustrates the changes in the training and validation loss functions during the training process. It can be observed that both loss functions decrease as the epoch increases. However, when the epoch exceeds 500, the training loss function continues to decrease while the validation loss function remains relatively constant. This suggests that the model will become overfitted if training continues beyond this point. For each upsampling scale factor α, we select the epoch with the smallest validation loss function as the final model.

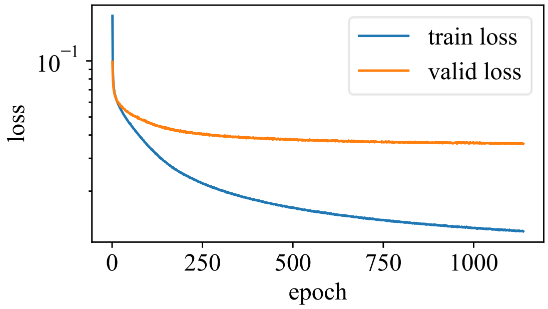


Fig. 4. Training and test losses during the network training. α = 8.

B. Comparison with baseline on the test set in the first flight test

The results on the test set of the first flight test are presented first. The obtained upsampling results are shown to be good for different upsampling factors, indicating that the network effectively learns the flight parameter variation patterns between adjacent low sampling intervals. It is observed that the RMSE and FFTE increase with increasing α, which is in line with reality since upsampling becomes more challenging with the increase of α.

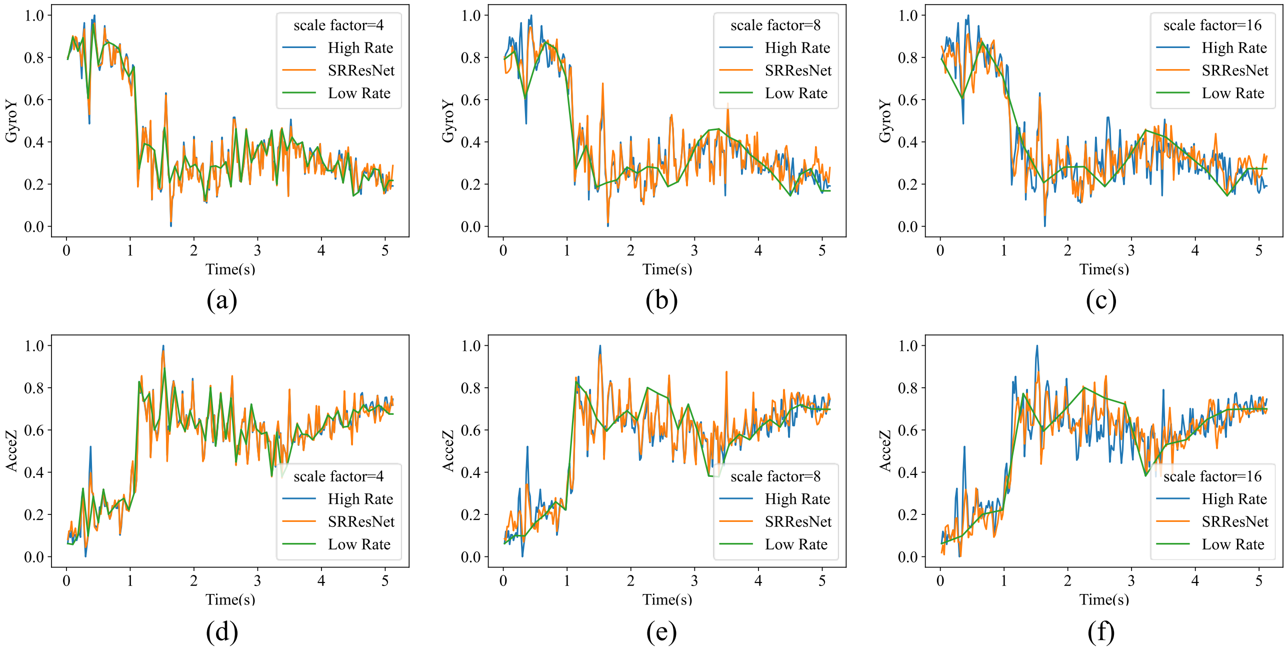
TABLE IV. TEST ERROR IN THE TEST SET OF THE FIRST FLIGHT TEST

α	**RMSE**		**FFTE**	
	SRResNet 1D	***Baseline***	***SRResNet 1D***	***Baseline***
4	**0.0248**	0.0663	**0.197**	0.547
8	**0.0478**	0.0914	**0.354**	0.742
16	**0.0908**	0.122	**0.581**	0.899

C. Test result on another flight test

TABLE V. ERROR ON THE FLIGHT DATA OF THE SECOND FLIGHT TEST

α	**RMSE**		**FFTE**	
	SRResNet 1D	***Baseline***	***SRResNet 1D***	***Baseline***
4	**0.0588**	0.0633	**0.415**	0.514
8	**0.0866**	0.0879	**0.564**	0.704
16	0.117	**0.116**	**0.680**	0.855



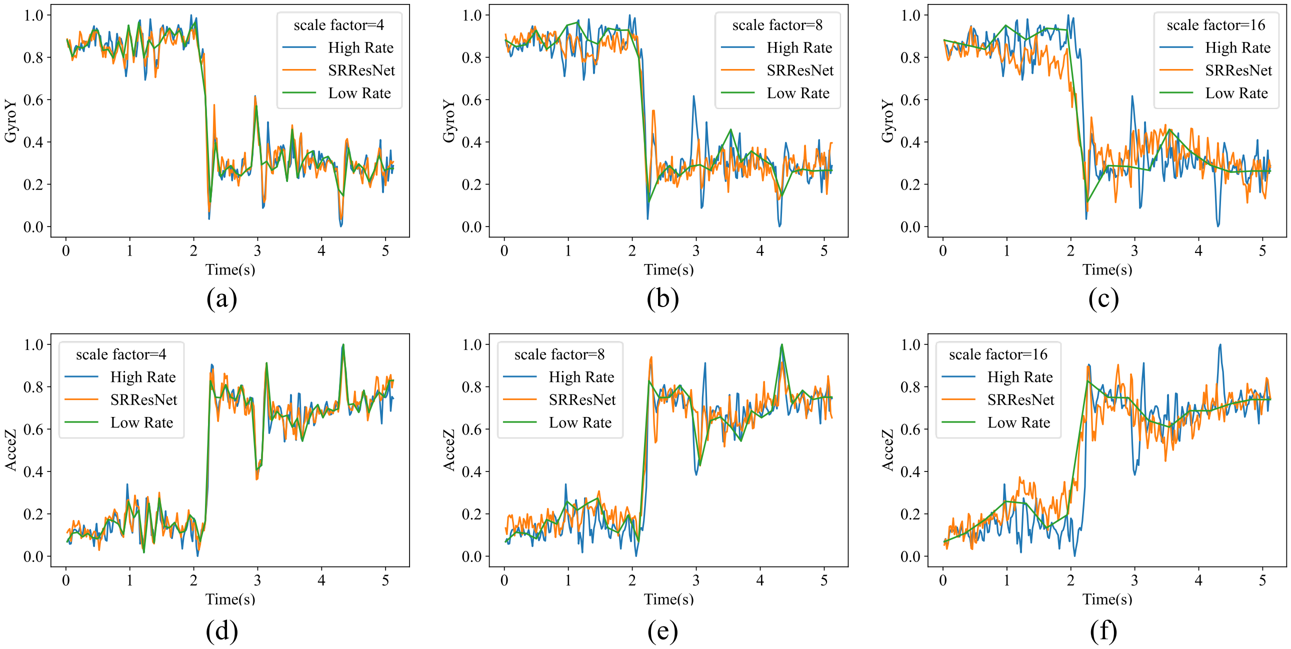
(a) (b) (c)

(d) (e) (f)

Fig. 5. Upsampling of flight parameters on the test set of the first flight data.

In this subsection, the data of the second flight test were used for testing. It is worth noting that the distribution of the data is different due to the difference in time and environment between the two flight tests.

Table IV presents a comparison of the errors on the flight data from the second flight test. It can be observed that the RMSE error is slightly higher than the baseline, while the FFTE is lower than the baseline. However, the improvement is not as significant as that achieved in the first flight test. This indicates that differences in data distribution have a considerable impact on the performance and should be taken into account in future studies. Fig. 6 shows the results of upsampling with typical the errors.



(a) (b) (c)

(d) (e) (f)

Fig. 6. Upsampling of flight parameters of the second flight data.

V. Conclusion

In this study, we proposed a one-dimensional super-resolution residual convolutional neural network-based upsampling method for flight parameters to provide high sampling rate data for structural digital twin simulation. Our method achieved upsampling through multiple pixel shuffle layers and multiple tandem residual blocks, which extracted the features of flight and environmental parameters. We defined several loss functions related to the flight parameter, including time and frequency domains. Validation on flight test data of a flying wing UAV showed that the proposed method can effectively improve the sampling rate of flight data with multiple upsampling factors such as 4, 8, and 16. The error in both time and frequency domains can be reduced compared to the baseline method. However, the test on another flight test data showed that the method needs to improve its generalization ability under different data distributions in the future. Overall, our study provides a promising approach for enhancing the accuracy of structural digital twin simulation with high sampling rate flight data.

References

[1] A. Thelen *et al.*, “A comprehensive review of digital twin — part 1: modeling and twinning enabling technologies,” *Struct Multidisc Optim*, vol. 65, no. 12, p. 354, Nov. 2022, doi: 10.1007/s00158-022-03425-4.

[2] A. Thelen *et al.*, “A comprehensive review of digital twin—part 2: roles of uncertainty quantification and optimization, a battery digital twin, and perspectives,” *Struct Multidisc Optim*, vol. 66, no. 1, p. 1, Dec. 2022, doi: 10.1007/s00158-022-03410-x.

[3] E. Glaessgen and D. Stargel, “The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles,” in *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Honolulu, Hawaii: American Institute of Aeronautics and Astronautics, Apr. 2012, p. 1818. doi: 10.2514/6.2012-1818.

[4] C. Li, S. Mahadevan, Y. Ling, S. Choze, and L. Wang, “Dynamic Bayesian Network for Aircraft Wing Health Monitoring Digital Twin,” *AIAA Journal*, vol. 55, no. 3, pp. 930–941, Mar. 2017, doi: 10.2514/1.J055201.

[5] W. Sisson, P. Karve, and S. Mahadevan, “Digital twin for component health- and stress-aware rotorcraft flight control,” *Struct Multidisc Optim*, vol. 65, no. 11, p. 318, Oct. 2022, doi: 10.1007/s00158-022-03413-8.

[6] X. Zhou, S. He, L. Dong, and S. N. Atluri, “Real-Time Prediction of Probabilistic Crack Growth with a Helicopter Component Digital Twin,” *AIAA Journal*, vol. 60, no. 4, pp. 2555–2567, Apr. 2022, doi: 10.2514/1.J060890.

[7] F. Zhao, X. Zhou, C. Wang, L. Dong, and S. N. Atluri, “Setting Adaptive Inspection Intervals in Helicopter Components, Based on a Digital Twin,” *AIAA Journal*, pp. 1–14, Feb. 2023, doi: 10.2514/1.J062222.

[8] T. Li, L. Lomazzi, F. Cadini, C. Sbarufatti, J. Chen, and S. Yuan, “Numerical simulation-aided particle filter-based damage prognosis using Lamb waves,” *Mechanical Systems and Signal Processing*, vol. 178, p. 109326, Oct. 2022, doi: 10.1016/j.ymssp.2022.109326.

[9] C. Keryk, R. Sabatini, K. Kourousis, A. Gardi, and J. M. Silva, “An Innovative Structural Fatigue Monitoring Solution for General Aviation Aircraft,” *J.Aerosp. Technol. Manag.*, vol. 10, Feb. 2018, doi: 10.5028/jatm.v10.779.

[10] J. Öström, “Enhanced Inverse Flight Simulation for a Fatigue Life Management System,” in *AIAA Modeling and Simulation Technologies Conference and Exhibit*, Hilton Head, South Carolina: American Institute of Aeronautics and Astronautics, Aug. 2007. doi: 10/ghkdpx.

[11] J. Kim, J. K. Lee, and K. M. Lee, “Deeply-Recursive Convolutional Network for Image Super-Resolution,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 1637–1645. doi: 10.1109/CVPR.2016.181.

[12] C. Ledig *et al.*, “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network,” presented at the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society, Jul. 2017, pp. 105–114. doi: 10.1109/CVPR.2017.19.

[13] C. Saharia *et al.*, “Palette: Image-to-Image Diffusion Models,” in *ACM SIGGRAPH 2022 Conference Proceedings*, in SIGGRAPH ’22. New York, NY, USA: Association for Computing Machinery, 2022, pp. 1–10. doi: 10.1145/3528233.3530757.

[14] L. Song, Y. Li, and N. Lu, “ProfileSR-GAN: A GAN Based Super-Resolution Method for Generating High-Resolution Load Profiles,” *IEEE Transactions on Smart Grid*, vol. 13, no. 4, pp. 3278–3289, Jul. 2022, doi: 10.1109/TSG.2022.3158235.

[15] Y. Xie, E. Franz, M. Chu, and N. Thuerey, “tempoGAN: a temporally coherent, volumetric GAN for super-resolution fluid flow,” *ACM Trans. Graph.*, vol. 37, no. 4, p. 95:1-95:15, 2018, doi: 10.1145/3197517.3201304.

[16] L. Colombo *et al.*, “Numerical and experimental verification of an inverse-direct approach for load and strain monitoring in aeronautical structures,” *Structural Control and Health Monitoring*, vol. 28, no. 2, p. e2657, 2021, doi: 10.1002/stc.2657.

[17] L. Colombo, C. Sbarufatti, W. Zielinski, K. Dragan, and M. Giglio, “Numerical and experimental flight verifications of a calibration matrix approach for load monitoring and temperature reconstruction and compensation,” *Aerospace Science and Technology*, vol. 118, p. 107074, Nov. 2021, doi: 10.1016/j.ast.2021.107074.