A Digital Twin Framework for Detecting the Damage Location

Ali Awan,[*] Enixe Aguliar,[†] Roya Salei,[‡] Markus Eger[§] Zahra Sotoudeh [¶]

California State Polytechnic Institute, Pomona, CA 91768

Extended Abstract

This paper reports on the progress of the Cal Poly Pomona Digital Twin Technology for Aerospace Engineering (CPP-DiTTA) ongoing effort. The project aims to create a technology demonstrator for a digital twin. The CPP-DiTTA demonstrator predicts the location of the damage on a plate structure using real-time sensory data and produces a three-dimensional visualization of the deformed configuration, including the damage under the given specific loads.

Digital Twin is a concept of high interest to the aerospace community, including the US Air Force and several private companies.[1,2] A digital twin is a tail number-specific computational model of an individual aircraft (or other physical assets).[3] The digital twin will be updated based on sensory data over the life of the physical asset. A digital twin decreases maintenance costs by acting as a live health monitoring system. It also helps engineers and operators make better decisions based on running *what-if* analysis on the digital twin.[4] Many papers explain the concept of the digital twin,[5–8] but not many articles include the details of creating a digital twin.[9] The CPP-DiTTA goal is to create a technology demonstrator for a digital twin for educational purposes. Our digital twin system includes the following:

- a physical asset (a plate structure),
- a computational tool that updates at given time intervals based on sensory data,
- a virtual reality module.

In addition to advanced finite element analysis, a machine learning (ML) algorithm is at the heart of this project.

The Content of the Full Paper

The full paper will include a literature review on digital twins for aerospace engineering. It will present the complete framework of the CPP-DiTTA and predictive results. *This Digital twin uses sensory data to predict the location of damage in a plate structure.* Fig. 1 shows the system architecture. In creating this framework, we tested two machine learning algorithms: a decision tree and a k-Nearest Neighbors (kNN). The full paper includes a comparison of the results using each algorithm.

CPP-DiTTA System Architecture

In the this section, we briefly explain each component of the system architecture (Fig. 1). The details of implementation and extensive results and verifications will be reported in the full paper.

[*]Graduate Students, Computer Science Department, California State Polytechnic Institute, Pomona, CA 91768, USA.

[†]Undergraduate Students, Aerospace Engineering Department, California State Polytechnic Institute, Pomona, CA 91768, USA.

[‡]Former Undergraduate Students, Computer Science Department, California State Polytechnic Institute, Pomona, CA 91768, USA.

[§]Assistant Professor, Computer Science Department, California State Polytechnic Institute, Pomona, CA 91768, USA.

[¶]Associate Professor, Department of Aerospace Engineering, California State Polytechnic Institute, Pomona, CA 91768, USA, AIAA Associate Fellow.

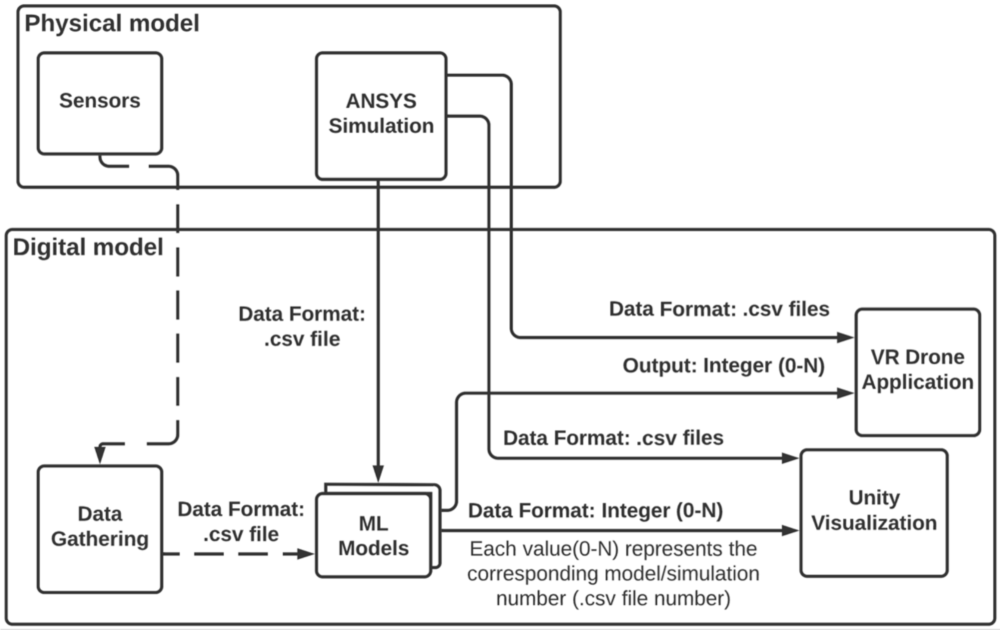
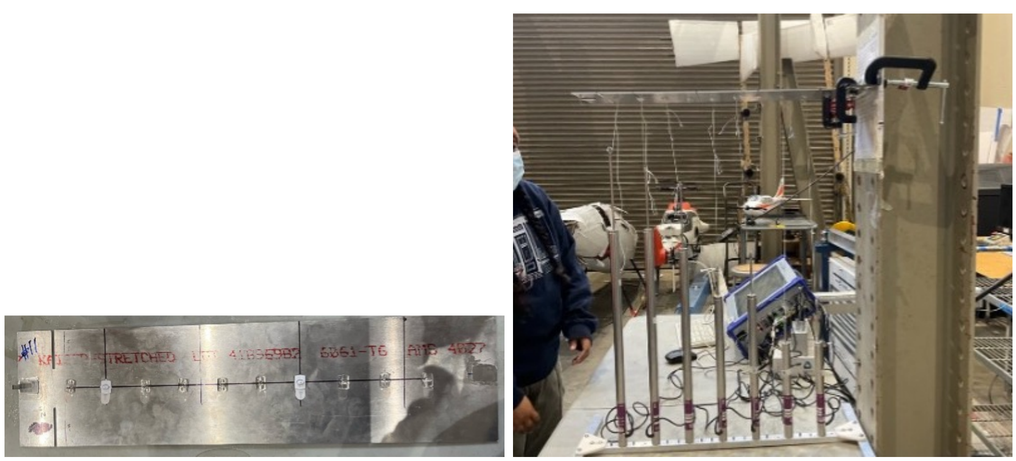


Figure 1: *The SystemArchitecture for CPP-DiTTA*

The Physical Asset

The physical asset is a rectangular aluminum plate. At this phase, we will only focus on the ground experiments. We start with known structural and geometric properties for the perfect wing (no damage), which is the state of an aircraft at the start of its service life. We install sensors on this plate, which allow us to read strain and stresses on the plate. We will use two sets of sensors, fiber optics to measure stress and linear variable differential transformers (LVDTs) to measure deformation. We update the digital twin based on this data at given intervals. In an experimental setup, we build similar plates with different cuts at different locations to create the structure's accumulated damage during its lifetime. These changes simulate the effect of environmental conditions in a laboratory setting. *The digital twin predicts the location of the cuts based on only the available sensor data.* Fig. 2 (a) shows one of such plates with cuts (damage) two inches from the plate's tip. Fig. 2 (b) shows the experimental setup using LVDTs. The sensors are placed one inch away from each other. *The full paper includes the results with both LVDTs and fiber optics.*



(a) A plate with two cuts at two inches from plate's tip

(b) Experimental setup with LVDTs

Figure 2: *The damaged physical asset and the test setup with LVDTs*

Numerical Simulations

We use ANSYS to model plates with the cuts at different locations. First, we verify our ANSYS model using the experimental data. Next, the ANSYS results are used to generate data sets to train the machine learning models. Fig. 3 shows a mesh for ANSYS simulations for the beam in Fig. 2.

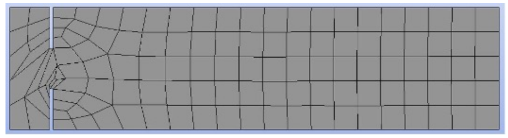


Figure 3: *FEM model of the plate in ANSYS*

The Digital Componenets

Three digital subcomponents are Data Gathering, Machine Learning (ML) models, and Visualization. Data Gathering involves cleaning the data sent by the sensors attached to the physical asset, ensuring that the data forwarded to the ML model resembles the training data. In the full paper, we will present results for two ML models, a decision tree and a k-Nearest Neighbor (kNN).

Decision trees and kNN are popular machine-learning algorithms for classification and prediction tasks. Decision trees recursively partition the input space based on the features that maximize the class separation. kNN assigns labels to input points based on the most common class among their nearest neighbors.

Decision Tree Algorithm

The decision tree algorithm recursively partitions the input space into smaller regions. Each partition is associated with a specific prediction, mainly used for classification tasks. The algorithm chooses a feature to split upon based on a criterion that maximizes the separation between the classes, thereby achieving these partitions. This process continues until a stopping criterion is achieved or the data has been completely partitioned.[10]

k-Nearest Neighbors (kNN)

The k-nearest neighbors (kNN) is a non-parametric lazy learning algorithm for classification and regression. The algorithm finds the k-nearest neighbors of a given input point and assigns the label of the most common class among those neighbors. The choice of k is a hyper parameter that can be tuned to optimize the algorithm's performance. The kNN is a simple but powerful instance-based learning algorithm.[10]

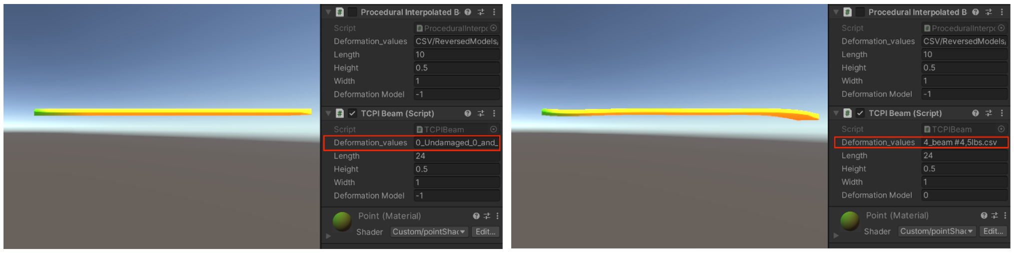
Visualization

One of our key goals is to provide an intuitive environment for users to understand the simulation results in a 3D virtual environment. We will be using the 3D game engine Unity, which has been applied to various game and non-game visualization tasks, including for industrial processes.[11] Using this game engine, we can create a 3D environment containing the physical asset as a 3D model that users can inspect from any side and angle and interact with intuitively.

The Visualization module is a Unity application that displays a 3D model closely resembling the actual beam's condition. The Unity visualization accepts numeric data from the machine learning models, ranging from 0-N via a socket connection, where each value corresponds to a specific model number. The model number sent by the machine learning model represents the closest approximation to the actual asset. The Unity visualization continuously updates based on the latest sensor data, accurately reflecting the physical beam's current state.

References

[1]Committee, A. D. E. I. et al., "Digital Twin: Definition & Value—An AIAA and AIA Position Paper," *AIAA: Reston, VA, USA*, 2020.



(a) Visualization of an undamaged beam

(b) Visualization of a damaged beam

Figure 4: *Samples of Unity Simulations of CPP-DiTTA*

[2]Singh, M., Fuenmayor, E., Hinchy, E. P., Qiao, Y., Murray, N., and Devine, D., "Digital twin: Origin to future," *Applied System Innovation*, Vol. 4, No. 2, 2021, pp. 36.

[3]Tuegel, E., "The airframe digital twin: some challenges to realization," *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA*, 2012, p. 1812.

[4]James, S. C. and Hale, B., "What Do Virtual V&V and Digital Twins Have in Common?" *AIAA Scitech 2021 Forum*, 2021, p. 0176.

[5]Kraft, E. M., "The air force digital thread/digital twin-life cycle integration and use of computational and experimental knowledge," *54th AIAA aerospace sciences meeting*, 2016, p. 0897.

[6]Kobryn, P., Tuegel, E., Zweber, J., and Kolonay, R., "Digital thread and twin for systems engineering: EMD to disposal," *Proceedings of the 55th AIAA Aerospace Sciences Meeting*, 2017.

[7]Aydemir, H., Zengin, U., and Durak, U., "The Digital Twin Paradigm for Aircraft Review and Outlook," *AIAA Scitech 2020 Forum*, 2020, p. 0553.

[8]Harper, D. J., "Effectiveness-Based Design as an Important Part of the Conceptual Digital Twin: Observations from AFRL's EXPEDITE Program," *AIAA Scitech 2021 Forum*, 2021, p. 1353.

[9]Kapteyn, M. G., Knezevic, D. J., and Willcox, K., "Toward predictive digital twins via component-based reduced-order models and interpretable machine learning," *AIAA Scitech 2021 Forum*, 2021, p. 0418.

[10]Alpaydin, E., *Introduction to machine learning*, MIT press, 2020.

[11]Wang, J., Phillips, L., Moreland, J., Wu, B., and Zhou, C., "Simulation and visualization of industrial processes in unity," *Proceedings of the Conference on Summer Computer Simulation*, 2015, pp. 1–7.