Towards population-based structural health monitoring, Part II: Heterogeneous populations and structures as graphs

J. Gosliga, P.A. Gardner, L.A. Bull, N. Dervilis and K. Worden

Dynamics Research Group, Department of Mechanical Engineering
University of Sheffield, Sheffield S1 3JD, UK

Abstract

Information about the expected variation in the normal condition and various damage states of a structure is crucial in structural health monitoring. In an ideal case, the behaviour associated with each possible type of damage would be known and classification would be possible. However, it is not realistic to obtain data for every possible damage state in an individual structure. Examining a population of structures gives a much larger pool of data to work with. Machine learning can then potentially allow inferences across the population using algorithms from transfer learning.

The degree of similarity between structures determines the level of possible knowledge transfer between different structures. It is also useful to quantify in which ways two structures are similar, and where these similarities lie. This information determines whether or not certain the transfer learning approaches are applicable in a given situation. It is therefore necessary to develop a method for analysing the similarities between structures. First, it must be decided which properties of the structure to use when measuring the similarity. For example, comparing 3D CAD models or Finite Element models is not a suitable approach, since these contain a lot of irrelevant information. It is better to abstract this information into a form that contains only the relevant information.

This paper proposes Irreducible Element (IE) models, which are designed to capture the features that are crucial in determining whether or not transfer learning is possible. This information is then converted into an Attributed Graph (AG). The Attributed Graph for a structure contains the same information as the Irreducible Element model; however, the graph carries this information as a list of attributes attached to nodes. Organising the information in this manner makes it easier for graph-matching algorithms to perform a comparison between two structures. This comparison can then be used to generate a measure of similarity between the two structures and determine the most appropriate transfer learning method.

Keywords: Population-based structural health monitoring; Irreducible Element model; Attributed Graph

1 Introduction

Moving beyond detecting damage in a structure, to locating and diagnosing the type of damage, raises a *supervised leaning* problem, which requires data relating to each particular type of damage (also known as the damage states of the structure). There is a large cost associated with obtaining information on the damage states of a particular engineering structure. *Population-based SHM* (PBSHM) seeks to reduce this cost by developing methods to allow the sharing of data between structures. The concept of PBSHM has been introduced in [1, 2]. If the population of structures is *homogeneous* (i.e. composed of nominally-identical structures), then it may be possible to establish a normal condition which is common across the population of structures. Even if the population is *heterogeneous* (composed of disparate structures), it may be possible to use information regarding the damage state from one structure to diagnose the same type of damage across the population. The technology that appears most likely to allow the transfer of information across populations is found in the machine learning discipline of *Transfer Learning*.

Within a strongly homogeneous population of structures, all structures have nominally-identical materials, geometry, and topology. (*Topology* in this case refers to how the individual parts which make up the structure are attached to one another.) This usually implies that these are structures of the same make and model, and that the only variation within the population is due to manufacturing uncertainty. In addition, structures in a *strongly homogeneous* population must all have the same boundary conditions. It is often the case that populations do not meet all these criteria, but are still homogeneous to some degree. This situation is typically the case for wind farms, and so far there has been work taking the first steps towards PBSHM in the Lillgrund wind farm [3]. There are many ways of performing PBSHM for a homogeneous population with one approach being to represent a homogeneous population with a single model called a form [4], and another approach being the use of transfer learning.

For a population of heterogeneous structures, each structure can have completely different materials, geometry, or topology. Within such a population, it is necessary to assess, on a case-by-case basis, whether or not damage data from one structure can be used to diagnose damage in another. For example, it may not be possible to use damage data from an aeroplane to locate damage in a bridge; however, damage data from one aeroplane may prove useful when diagnosing damage in a different aeroplane. In addition, the correct tools for making these diagnostic inferences will depend on the level of similarity between the structures [5]. Therefore, it is necessary to develop a method for consistently and quantitatively assessing the degree of similarity between structures.

The first step in assessing the similarity between structures, in the framework proposed here, involves creating an *Irreducible Element* (IE) representation of the structure, which is closely tied to the physics of the structure. This IE representation is then converted to an *Attributed Graph*. The procedure for generating an IE representation of a structure, along with a brief description of how to generate the Attributed Graph is included in [6]. This paper will expand on the choice of properties included in the IE representation, and how these help to determine the level of inference possible between structures.

The AG representation makes it possible to quantitatively assess the degree of similarity between structures. Attributed Graphs have previously been used in manufacturing in [7, 8], to measure the degree of similarity in the geometry of manufactured parts. The attributes of the graph provide an efficient method for assessing any differences in the materials or geometry of the structures. The graph directly reflects the topology of the structure it represents and so can be used to assess any differences in the topology between structures. This paper will also describe the general procedure for converting the IE representation of a structure into an Attributed Graph, and highlight how a computer implementation of the ideas can be designed, with reference to the language *Python*.

Section 2 will define the IE models for both an aeroplane and a wind turbine, with a discussion of how the geometry, topology (where joints form part of the description of topology), and material properties of the IE model affect knowledge transfer between two structures. Geometry is discussed in Section 2.1. Topology and joints are discussing in Section 2.2 and Section 2.2.1, respectively. Material properties are discussed in Section 2.3. The procedure for creating a AG from the IE model is described in Section 3. The approach for including boundary conditions in the IE and AG is described in Section 4. Formal definitions for homogeneous and heterogeneous populations are provided in Section 5.

2 Irreducible element representations of structures

To determine whether or not two structures share similarities which allow inferences to be made, it is not necessary to consider every property or dimension of the structure. For example, in order to determine geometric similarity (to a certain resolution), it would be computationally inefficient to compare 3D models—such as Finite Element (FE) models—of structures directly. Such 3D models contain a large amount of redundant information (i.e. the detailed construction of the mesh in an FE model) that have no strong effect on the overall geometry of the structure. It is is far more useful to consider which properties and dimensions are significant and abstract this information.

The proposed solution is an IE representation of the structure. For a given structure, an IE model attempts to simplify the geometry of various structural components into shapes that have well-defined dynamic behaviour, such as beams or plates, as illustrated in Fig. 1. The motivation for this is the belief that, for example, all plates exhibit similar behaviour (given a similar set of boundary conditions) and therefore would respond to damage in a similar way, or would have similar damage features. In the case where two IEs have the same shape and dimensions, it can be assumed that the behaviour exhibited is identical for a given set of boundary conditions. However, in reality it is likely that the geometry will be somewhat more complex and matches will not be exact. The complexity of the geometry and the degree of uncertainty within the match will influence the certainty with which inferences can be made. The IE representation can be used to determine which inference tools are appropriate, as well as the level of information that can be inferred. For example, if the method used to transfer information is transfer learning, the structures would need to be similar enough that the issue of *negative transfer* did not arise.

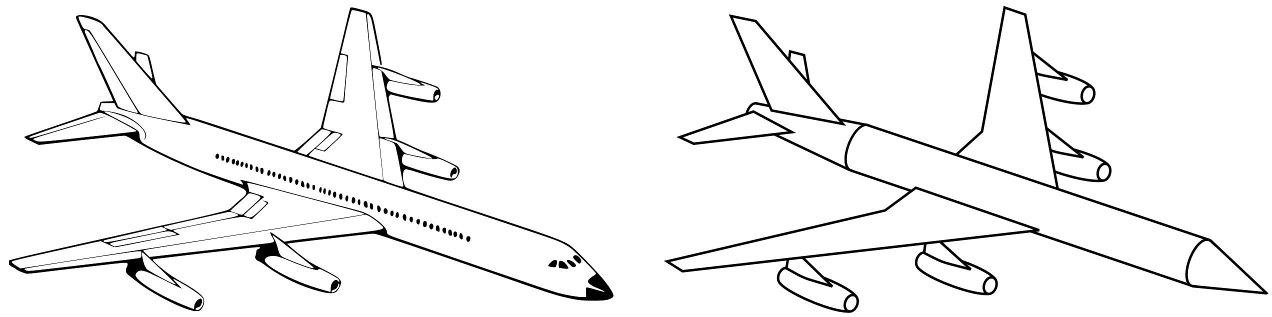

Figure 1: Irreducible Element representations seek to remove dimensions from a model which do not significantly affect the bulk dynamic properties of a structure, or have no relevance in determining the level of information transfer possible.

The IE representation captures important information about the geometry, topology, and materials in a structure that make it possible to determine the level of homogeneity between two structures. These aspects of a structure tie into the level of information transfer between two structures, based on Rytter's hierachy [9]. (The relation between differences in the IE representations of a structure and the level of inference possilbe are discussed further in [5].) The most extreme case would be if the IE representations of two structures were *identical*, which realistically would only occur when comparing a structure with itself. Within a population of structures that are all the same make and model, such as a wind farm, the IE representations would be *nominally-identical*. This condition implies that there is some probability distribution over the dimensions and material properties due to variation in manufacturing. If the population is described as strongly homogeneous, this also implies that the boundary conditions for the two structures are the same, for example all turbines in a wind farm with foundations on the same part of the seabed. A population where the IE representations are at least nominally-identical is a homogeneous population, and the idea of a *form* can be used in this case [4], in addition to other transfer learning methods.

Geometry, topology and materials have been chosen here as the main characteristics to use when judging the level of homogeneity of structures. Geometry is important when determining the overall dynamic behaviour of structures. If it is possible to assume that the dynamic behaviour is the same between two structures, then (given the same environmental conditions) any difference in the behaviour of the two structures must be due to damage. This can be seen in the Lilligrund windfarm example [3]. If the topology of two structures is the same, then it is possible to assume that any damage location labels can be applied in a consistent manner on both of them. This observation is true even if the entire structure is not similar. If two structures share a common substructure, then it may be possible to infer damage location between the two structures, provided damage occurs within this common substructure. This idea works because within this common substructure, feature and location labels should be consistent. Including materials in the description means one can determine whether inferences on the type of damage is possible. If two structures are made from the same materials, then the extent and classification labels for damage will be consistent, enabling the use of transfer learning. The following sections will detail the relevance of these properties further.

2.1 Geometry

Since it may not be possible to obtain complete geometrical information for all structures, it is necessary to define a hierachy of properties which define the IE to increasing levels of precision. This can be seen in Tables 1 and 2, where the coarsest level of description is the *geometry* class, followed by *shape*. In the full IE model there will be a further level where the shape is fully defined with the major dimensions, for example length, breadth and width. This hierachy of properties encodes the fact that certain transfer learning approaches are valid at a coarser level than

Table 1: List of elements and their properties for Turbine 1

Element designations for Turbine 1				
Name	**Element ID**	**Material**	**Geometry**	**Shape**
Rotor blade	A	FRP	Beam	Aerofoil
Rotor blade	B	FRP	Beam	Aerofoil
Rotor blade	C	FRP	Beam	Aerofoil
Rotor hub	D	FRP	Complex	Rotor hub
Nacelle	E	FRP	Shell	Cuboid
Tower section 1	F	Metal	Beam	Cylindrical
Tower section 2	G	Metal	Beam	Cylindrical
Tower section 3	H	Metal	Beam	Cylindrical
Foundation	I	Concrete	Plate	Cylindrical
Name	**Element ID**	**Boundary**	-	-
Footing	1	Ground	-	-

Table 2: List of elements and their properties for Aeroplane 1

Element designations for Aeroplane 1				
Name	**Element ID**	**Material**	**Geometry**	**Shape**
Fuselage	A1	FRP	Shell	Truncated cone
Fuselage	A2	FRP	Beam	Cylindrical
Fuselage	A3	FRP	Shell	Cone
Wing 1	B	FRP	Beam	Aerofoil
Pylon 1	C	FRP	Complex	Pylon
Engine 1	D	Assembly	Shell	Cylinder
Pylon 2	E	FRP	Complex	Pylon
Engine 2	F	Assembly	Shell	Cylinder
Wing 2	G	FRP	Beam	Aerofoil
Pylon 3	H	FRP	Complex	Pylon
Engine 3	I	Assembly	Shell	Cylinder
Pylon 4	J	FRP	Complex	Pylon
Engine 4	K	Assembly	Shell	Cylinder
Vert stabiliser 1	L	FRP	Beam	Aerofoil
Vert stabiliser 2	M	FRP	Beam	Aerofoil
Horz stabiliser	N	FRP	Beam	Aerofoil
Front landing gear	O	Assembly	Complex	Assembly
Rear landing gear	P	Assembly	Complex	Assembly
Name	**Element ID**	**Boundary**	-	-
Tarmac	1	Ground	-	-

others, with some able to deal with different parameters in the source and target domains when transfer learning is applied [5]. Similarities become better defined at a lower level of the hierarchy and improve the chance of success knowledge transfer. Using the example of a beam, certain inferences will be possible across all beams, some will require that the cross-section is the same, and other inferences rely on the geometry matching exactly.

The first four proposed geometry classes are: *beam*, *plate*, *shell*, and *complex* (and these will be sufficient for the purposes of the discussion within this paper). Beam and shell are fairly self-explanatory and correspond to the definitions used in multi-body modelling. The class of plate is introduced to differentiate between shells which enclose a volume and those that do not. There are three main cases where the complex class becomes necessary. Firstly, the complex class is used to represent the components within the structure that could in theory be broken down further into simpler IEs[1], but doing so would mean creating multiple elements out of a single structural component. An example of this can be seen in the elements for the aeroplane in Table 2, where the fuselage has been broken into separate elements in order to better define the geometry. However, when it comes to similarity matching, there are

[1]There might appear to be a contradiction in the terminology here, in the idea that an *irreducible* element might be decomposed. However, the term is used here in the sense that an IE model can provide a description of *sufficient* complexity for the purposes of matching structures, but the user may choose a higher resolution.

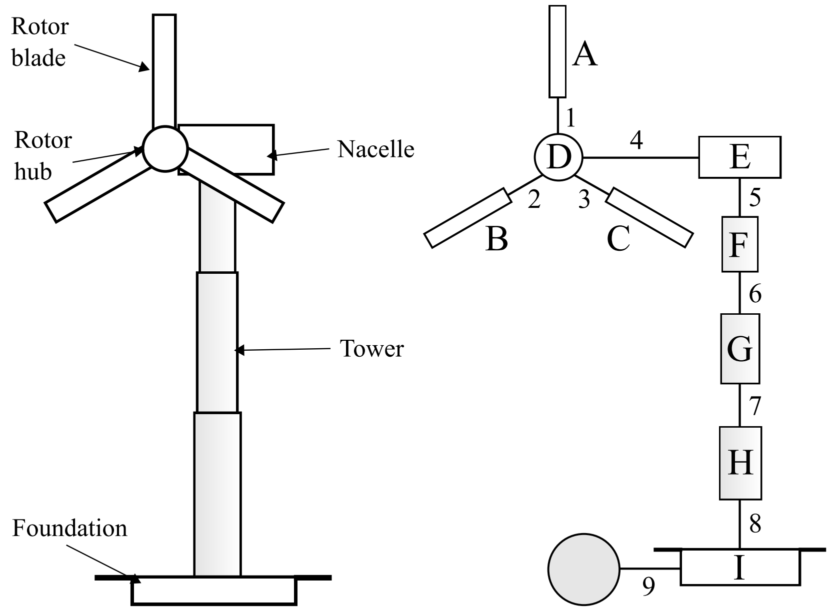

Figure 2: An expanded IE representation of a wind turbine with the elements labelled A to I and the connections labelled 1 to 9. The shaded node is a special element representing the ground, where the boundary condition is defined in the attributes for Joint 9.

benefits to defining it as a single complex element. Using aeroplanes as an example, it may be beneficial to define the fuselage as a single element so that each aeroplane now shares a common feature: a fuselage element attached to two wing elements. Secondly, the complex class can be used in the case where it is beneficial to represent a collection of components as a single element, for example the landing gear in Table 2 is represented as a single complex element. The third case where the *complex* class is useful is when there exists an element which is not easily described by a simple shape, for example the rotor hub on a wind turbine in Table 1.

2.2 Topology

The topology of the structure plays a large part in determining the dynamic behaviour of a structure. The topology here is determined by physical connections between the various elements, as shown in Fig. 2. With reference to Rytter's hierachy [9], the topology has a strong link to damage location. Structures without corresponding topology cannot have fully consistent location labels.

The complex class provides flexibility to simplify the topology of structures where it is believed that the topology is not significant for the problem. For example, if the application was comparison of the overall dynamics of an aeroplane, it may be useful to simplify the landing gear or the fuselage into a single element so that differences in overall configuration, such as engine placement are emphasised. Alternatively, if the application was locating damage within the fuselage, it would be beneficial to break this complex element into constituent parts.

2.2.1 Joints

Joints are classed as capturing topological information since they define where connections between elements occur; they also have the same effect on the label consistency as topology [5]. Example lists of joints and their properties can be found in Tables 3 and 4 for a wind turbine and an aeroplane respectively. The element set describes which elements the joint forms a connection for. This is crucial in recovering the topology of the structure. The coordinates define the geometrical location of the joint; the joint coordinate aids in constructing physical models from the IE representation. These coordinates define the mid-point of the joint and are determined by examining the geometry of the structure. The joint type describes some of the key physical aspects of the joint, such as the type of joint and the degrees of freedom that are contsrained. The joint type and restricted degrees of freedom are again hierarchical, as it may not

Table 3: List of joints and their properties for Turbine 1

Joint designations for Turbine 1					
Joint ID	**Element set**	**Coordinate**	**Type**	**Disp. DoF**	**Rot. DoF**
1	A, D	8, 15, 235.75	Bearing	[x, y, z]	[y, z]
2	B, D	8, 14, 254	Bearing	[x, y, z]	[y, z]
3	D, E	10, 15, 253	Bearing	[x, y, z]	[y, z]
4	D, C	8, 16, 254	Bearing	[x, y, z]	[x, y]
5	E, F	15, 15, 250	Bearing	[x, y, z]	[x, y]
6	F, G	15, 15, 183	Bolted	-	-
7	G, H	15, 15, 105	Bolted	-	-
8	H, I	15, 15, 5	Bolted	-	-
9	I, 1	15, 15, 0	Soil	-	-

Table 4: List of joints and their properties for Aeroplane 1

Joint designations for Aeroplane 1					
Joint ID	**Element set**	**Coordinate**	**Type**	**Disp. DoF**	**Rot. DoF**
1	A1, A2	34.2, 14.68, 5.165	Perfect	-	-
2	A2, A3	34.2, 60.96, 5.165	Perfect	-	-
3	A2, B	32.2, 29.79, 2.89	Lug	-	-
4	B, C	13.2, 42.67, 4.74	Complex	-	-
5	C, D	13.2, 40.17, 4.74	Complex	-	-
6	B, E	23.2, 30.79, 3.57	Complex	-	-
7	E, F	23.2, 28.29, 3.57	Complex	-	-
8	A2, G	36.2, 29.79, 2.89	Lug	-	-
9	G, H	45.2, 30.79, 3.57	Complex	-	-
10	H, I	45.2, 28.29, 3.57	Complex	-	-
11	G, J	55.2, 42.67, 4.74	Complex	-	-
12	J, K	55.2, 40.17, 4.74	Complex	-	-
13	A3, L	33.2, 68.58, 7.55	Lug	-	-
14	A3, M	35.2, 68.58, 7.55	Lug	-	-
15	A3, N	34.2, 64.58, 9.16	Lug	-	-
16	A1, O	34.2, 7.75, 1.75	Complex	-	-
17	A2, P	34.2, 29.67, 1.75	Complex	-	-
18	O, 1	34.2, 7.75, 0	Plane	[z]	[x, y]
19	P, 1	34.2, 29.67, 0	Plane	[z]	[x, y]

be possible to obtain this information in every case. It should also be noted that for static joints it is assumed that all degrees of freedom are constrained and so this property is not specified for static joints. The properties contained within the joints are important both for constructing the Attributed Graph, as well as for creating physical models of the structure.

Anticipating the next section a little, the *nodes* in the Attributed Graphs will represent the IEs, the *edges* will represent the joints between them.

2.3 Materials

The topology and geometry of the structure are most strongly linked with the transfer of damage detection and damage location labels. The material properties, on the other hand, determine the assesment and classification labels. A hierachical set of materials properties are used to describe the materials within an IE, as again it cannot be guaranteed that detailed information on materials used will be available for each structure. The coarsest level of detail will be the *material* class (cermaics, metals). The next level of detail will be a specific description of the material within the class, for example brass and steel belong to the material class of metals. The finest level of detail will be the specific properties of the material, for example Young's Modulus and density. Material grade may also form part of the attributes. Once again, to be confident of an exact match between two structures, all of the material properties must be known down to the bottom of the hierarchy, otherwise there will be a degree of uncertainty.

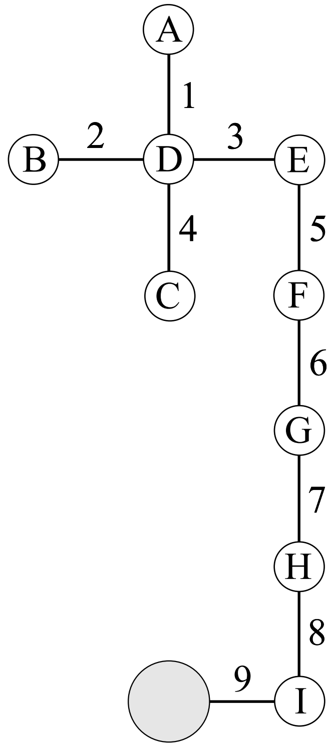

Figure 3: The resulting Attributed Graph from the IE representation shown in Fig. 2. It is possible to see how the topology is defined by the element set in the joints.

The material properties determine whether it is possilbe to make inferences using damage assessment and classification labels between two structures. Two materials of the same material class will experience similar failure modes, and could be expected to exhibit a similar response to a particular type of damage giving more confidence in the classification of damage. However, to build confidence that the assessment (extent) of the damage will be the same between two structures, the material would ideally be the same. For example, aluminium and steel will both suffer corrosion (damage classification), but there is little guarantee that the change in material properties will be the same for the same extent of corrosion. If the materials are identical, then transferring damage assessment and classification labels is trivial.

3 Producing an Attributed Graph

For the Attributed Graph (AG) the same information is embedded as for the IE representation, but whereas the properties for the IE representation are organised in a tabular format to improve readability by humans, the information in the Attributed Graph is organised so that it can be more efficiently processed by a graph-matching algorithm.

The structure of the elements and joints is extracted from the properties in the tables to create the graph itself. A graph G consists of a set of nodes V and edges E. The set of nodes contains all of the elements from the IE representation. The element set for each joint is used to construct the set of edges. One can define the graph of the turbine $G_{turbine}$ from the set of nodes $V_{turbine} = \{A, B, C, D, E, F, G, H, I, 1\}$ and set of edges $E_{turbine} = \{(A, D), (B, D), (D, E), (D, C), (E, F), (F, G), (G, H), (H, I), (I, 1)\}$. The graph is represented as a Python dictionary, which is a datatype where values are stored with a lookup key, e.g. dictionary = {‘key’ : ‘values’}. The dictionary representation of a graph contains the list of nodes (labelled with their IE designation as shown in Fig. 3) and the values for each node in the dictionary are a list of neighbouring nodes.

One can define a dictionary for a subgraph G_1 of the turbine graph $G_{turbine}$ shown in Fig. 3:

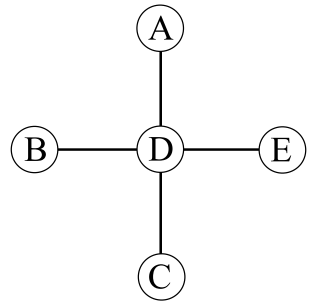

Figure 4: A sample subgraph of Fig. 3.

```
graph =   { ‘A’ :   [‘D’]
            ‘B’ :   [‘D’]
            ‘C’ :   [‘D’]
            ‘D’ :   [‘A’, ‘B’, ‘C’, ‘E’]
            ‘E’ :   [‘D’]  }
```

where the curly brackets are used to specify that the datatype will be ‘dictionary’. The quotation marks are used to denote strings, which are used as node labels in the code. The square brackets are used to specify a ‘list’. The edges of the graph can be obtained by examining the dictionary ‘key’ (to the left of the colon) and creating a pair with any single entry in the list of neighbours (to the left of the colon). For example, the edge (A, D) can be created by pairing the dictionary key A with its neighbour D. Alternatively, the equivalent edge (D, A) could be created by pairing the dictionary key D with its neighbour A.

Examining the subgraph G_1 shown in Fig. 4 with the full turbine graph $G_{turbine}$ shown in Fig. 3, it is clear that this is indeed a valid subgraph. By defining the node set $V_1 = \{A, B, C, D, E\}$ and edge set $E_1 = \{(A, D), (B, D), (D, E), (D, C)\}$ for the subgraph, it can be stated that $V_1 \subseteq V_{turbine}$ and $E_1 \subseteq E_{turbine}$. Since E_1 contains all edges from $E_{turbine}$ that have both endpoints in V_1, G_1 is an *induced* subgraph of $G_{turbine}$. For determining similarity between engineering structures, induced subgraphs are the only subgraphs of interest, since to preserve topology it is necessary to include all connections that exist within the two parent graphs when examining any sub-structures. In addition, only connected subgraphs will be considered, since in a structure, every part must be physically connected to at least one other part of the structure.

The dictionary data type in Python is also used to store the properties from the IE representation as node and edge attributes. This datatype allows the graph-matching algorithm to easily query the attributes of a given node, since the node label can be used as a dictionary key. The organisation of the information for elements consists of using the element ID as the dictionary key, and the element attributes are stored as a list, with nested lists containing the geometrical and material attributes. For example the node attributes for the elements in Table 1 appear as:

```
turbine.elements =  { ‘A’  :  [‘FRP’,        [‘Beam’, ‘Aerofoil’]],
                      ‘B’  :  [‘FRP’,        [‘Beam’, ‘Aerofoil’]],
                      ‘C’  :  [‘FRP’,        [‘Beam’, ‘Aerofoil’]],
                      ‘D’  :  [‘FRP’,        [‘Complex’, ‘Rotor hub’]],
                      ‘E’  :  [‘FRP’,        [‘Shell’, ‘Cuboid’]],
                      ‘F’  :  [‘Metal’,      [‘Beam’, ‘Cylindrical’]],
                      ‘G’  :  [‘Metal’,      [‘Beam’, ‘Cylindrical’]],
                      ‘H’  :  [‘Metal’,      [‘Beam’, ‘Cylindrical’]],
                      ‘I’  :  [‘Concrete’,   [‘Plate’, ‘Cylindrical’]],
                      ‘1’  :  [‘Ground’]  }
```

For edges, the label is exchanged with the element set. This is because when looking up edges in resulting subgraphs, the easiest way to query if an edge exists between node v_1 and node v_2 is to check whether $(v_1, v_2) \in E$ and so naturally edges are best labelled by their node pair. The coordinates and joint information (for the turbine joints in Table 3) is then organised in a similar fashion to the node attributes, where the hierachical information is contained within a nested list:

```
turbine.joints = { (‘A’, ‘D’)  :  [‘1’,  [8, 15, 235.75],  [‘Bearing’,  [‘x’, ‘y’, ‘z’],  [‘y’, ‘z’]]],
                   (‘B’, ‘D’)  :  [‘2’,  [8, 14, 254],     [‘Bearing’,  [‘x’, ‘y’, ‘z’],  [‘y’, ‘z’]]],
                   (‘C’, ‘D’)  :  [‘3’,  [8, 16, 254],     [‘Bearing’,  [‘x’, ‘y’, ‘z’],  [‘y’, ‘z’]]],
                   (‘D’, ‘E’)  :  [‘4’,  [10, 15, 253],    [‘Bearing’,  [‘x’, ‘y’, ‘z’],  [‘x’, ‘y’]]],
                   (‘E’, ‘F’)  :  [‘5’,  [15, 15, 250],    [‘Bearing’,  [‘x’, ‘y’, ‘z’],  [‘x’, ‘y’]]],
                   (‘F’, ‘G’)  :  [‘6’,  [15, 15, 183],    [‘Bolted’]],
                   (‘G’, ‘H’)  :  [‘7’,  [15, 15, 105],    [‘Bolted’]],
                   (‘H’, ‘I’)  :  [‘8’,  [15, 15, 5],      [‘Bolted’]],
                   (‘I’, ‘1’)  :  [‘9’,  [15, 15, 0],      [‘Soil’]]  }
```

where the regular parentheses represent a *tuple* in Python. While organising the information in this way reduces the readability from a human perspective, it makes it easier for a graph-matching algorithm to search for node or edge attributes. Furthermore, there are no empty rows or columns when presented with an incomplete set of attributes.

Given the set of nodes V and edges E for a structure, it is possible to plot the graph using Python. Plotting the graphs is not necessary for graph comparison, but can aid in assessing whether or not the code is producing valid subgraphs. An example of the graphs produced using the Python code is shown in Fig. 5. From these two graphs of the turbine and the aeroplane, it is possible to visually determine that G_1 (Fig. 4) is a valid subgraph (compare the subset of nodes $\{A, B, C, D, E\}$ in the turbine with subset of nodes $\{A2, A3, N, M, L\}$ in the aeroplane) of both $G_{turbine}$ and $G_{aeroplane}$. Using the node attributes, it is possible to align the aerofoil elements in each node set, giving the corresponding node set for the common subgraph $V_1 = \{(A, N), (B, M), (C, L), (D, A3), (E, A2)\}$. This node re-labelling requires an update of the edge set $E_1 = \{((A, N), (D, A3)), ((B, M), (D, A3)), ((C, L), (D, A3)), ((E, A2), (D, A3))\}$.

Once an AG is defined, it immediately induces a basic modal model – the *Canonical Modal Model* (CMM). This is constructed as a lumped-mass model as follows: first, one replaces each node (IE) in the model by a lumped mass computed using the appropriate node attributes (i.e. density and dimensions). Secondly, one replaces each edge by a spring with stiffness computed using the edge attributes. Taking care to preserve ground nodes, this procedure defines a basic model from which modal invariants (natural frequencies) can be computed.

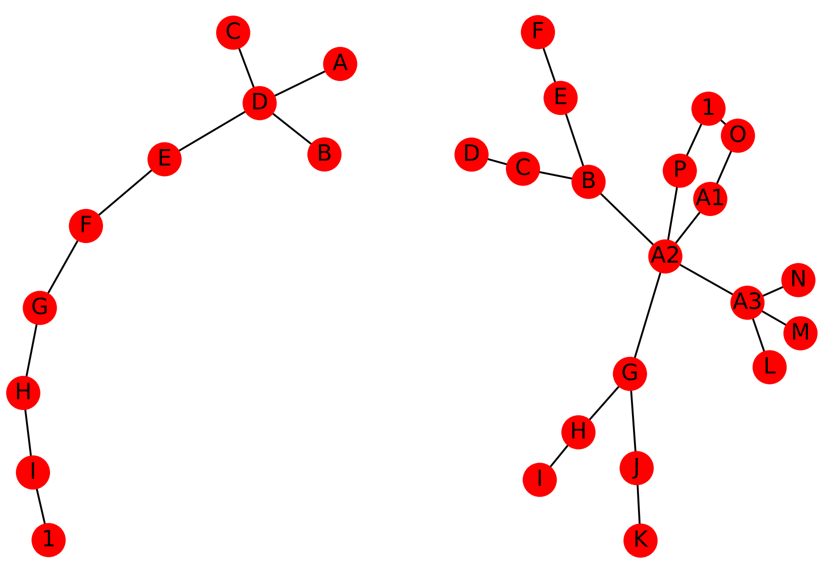

Figure 5: The graphs prodcued from list of elements in Tables 1 and 2 and list of joints in Tables 3 and 4. The graph shown on the left is from the wind turbine (compare with Fig. 3) and the graph on the right is from the aeroplane.

4 A note on boundary conditions

Because boundary conditions are essentially a type of joint, they are defined using special edges, which also determine which coordinate degrees of freedom are actually constrained. In fact, a complete boundary condition will be specified by an edge-node pair; defining the boundary condition and the object to which the boundary is connected. There are two main cases to deal with here; the first is when the connection is to 'ground', the second is when the structure is connected to another structure which is deemed to be distinct. Boundary conditions imposed by the same root cause can be represented by the same node. For example, in the case of the aeroplane described in Tables 2 and 4, both sets of landing gear as restricted by the tarmac (ground), as the aeroplane sits on the runway[2]. As such, these two boundary conditions can be represented by the same sort of edge, connected to the same node, which represents the runway (ground). The ground also features as part of a boundary condition for the wind turbine in Table 1, although as can be seen from Table 3, the wind turbine has a permanent static connection to ground. The same type of boundary condition can manifest in different ways depending on how the structure is connected, and so a complete match in boundary conditions requires that both the type of node, and the edge connected to the boundary are the same.

The justification for this is partially physical – all connections to 'ground' are modelled as rigid connections to the earth – but also practical. Having single edge-ground combinations for boundary conditions makes it possible to calculate the number of connections to ground (or other boundary conditions) by simply examining the degree of the ground nodes. (Degree is the number of edges connected to a node.)

'Ground' will be regarded as a very distinguished node in the formal definitions which will follow.

5 Some formal definitions for PBSHM

The details of the IE and AG constructions in this paper will now allow formal definitions for some of the terms that have been used up to now, like *homogeneous* and *heterogeneous* populations.

First of all a *population* of structures will simply be some set of N structures, $\{S_1, \dots, S_N\}$, which exist in reality. With each structure S_i, one can associate/construct an appropriate IE model $IE(S_i)$ or IE_i. Each of the IE models will induce an Attributed Graph representation $AG(S_i)$ or AG_i.

Two graphs G_1 and G_2 will be said to be *topologically equivalent* if they are topologically equivalent, when considered as simple graphs, i.e. there are the same number of nodes in each, and the pattern of connectivity is the same.

Now, one can exploit the existence of the special *ground* node that can appear in AGs, and say that the two graphs are *structurally equivalent* if they are topologically equivalent, and the ground nodes in both AGs occur in corresponding places.

Now, one defines two *structures* to be topologically (resp. structurally) equivalent, if they both have IE representations, that induce AG representations which are topologically (resp. structurally) equivalent as graphs.

Two structures will be said to be *strictly equivalent* if they are structurally equivalent, but also have corresponding IEs at corresponding nodes in the AG; this means that the attribute vectors for the nodes must have the same entries, with the same physical meanings.

Now, a population of structures $P = \{S_i\}$ is *topologically (resp. structurally, strictly) homogeneous*, if all structures in P are pairwise topologically (resp. structurally, strictly) equivalent. If *any* pair of structures fail in equivalence, the population P is *heterogeneous*.

Consider a strictly homogeneous population $P = \{S_i : i = 1, \dots, N\}$, where each structure S_i has parameters representing the node attributes, denoted by $\underline{\theta}^i_j$, where j is the node index, and parameters $\underline{\theta}^i_{jk}$, where j and k indicate that the edge is between nodes j and k. By the strict homogeneity assumption, the parameters in the vectors for each structure are in one-to-one correspondence in terms of physical meaning. A population now, will be termed *fully homogeneous* if the attributes can be considered to be random draws from distributions $p_j\underline{\theta}_j)$ and $p_{jk}(\underline{\theta}_{jk})$, which are *independent* of the structure. Clearly, this definition encompasses the idea that a homogeneous

[2] If the aircraft is stationary, this joint represents a static friction; if the aircraft is in motion, one would need to consider the joint as a rolling friction; if the aircraft were in flight, the joint would disappear

population is comprised of structures that are exactly the same make and model, with variations arising only from manufacturing/embodiment variations.

In order to fully motivate the idea of a *form*, it may be necessary to add a stronger condition, even than this. For example, one might say that a population is *strongly homogeneous* if it is homogeneous, and the attribute densities p_j and p_{jk} are unimodal, or convex etc.

A wind farm with all the turbines the same model, but subject to manufacturing variations would be expected to be strongly homogeneous, assuming that some parameterised foundation model was appropriate to all the turbines.

These definitions are part of the essential machinery that will be needed to assess if transfer of inferences is likely to succeed between two different structures. Part VI of the papers in this sequence, will return to this issue [5].

6 Conclusions

The properties of a pair of IE models of two structures determines the level of inference possible between the structures. This is due to the fact that consistent damage labels are required for transfer learning. Matching topology and element geometry gives consistent location labels. Matching the material properties for elements gives consistent damage classification and assessment labels. Therefore, when looking for exact matches, it is important to consider all three aspects of a particular structure: topology, materials and geometry. Accordingly, these three categories are captured in the information contained in the AG. The nodes contain attributes which correspond to these properties in the IE, however the information is arranged in a way that makes it more accessible to any graph-matching algorithms. Joints carry important topological information, particularly in terms of connections to ground, which is necessary for not only creating the AG, but also when searching for common subgraphs. By carefully choosing the information and data structure, it is possible to compare the AG for two structures in order to determine the most appropriate transfer learning tool. Comparing the attributes of the graphs (which correspond to the properties of the IE model) allows one to group structures based on the level of knowledge transfer possible between any two given structures.

7 Acknowledgements

The authors would like to thank the UK EPSRC for funding through the Established Career Fellowship EP/R003645/1 and the Programme Grant EP/R006768/1.

References

[1] I. Antoniadou, N. Dervilis, E. Papatheou, A.E. Maguire, and K. Worden. Aspects of structural health and condition monitoring of offshore wind turbines. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 373, 2015.

[2] K. Worden, E.J. Cross, N. Dervilis, E. Papatheou, and I. Antoniadou. Structural health monitoring: from structures to systems-of-systems. *IFAC-PapersOnLine*, pages 1–17, 2015.

[3] E. Papatheou, N. Dervilis, A.E. Maguire, I. Antoniadou, and K. Worden. A performance monitoring approach for the novel Lillgrund offshore wind farm. *IEEE Transactions on Industrial Electronics*, 62:6636–6644, 2015.

[4] L.A. Bull, P.A. Gardner, J. Gosliga, A.E. Maguire, C. Campos, T.J. Rogers, M. Haywood-Alexander, N. Dervilis, E.J. Cross, and K. Worden. Towards population-based structural health monitoring, Part I: Homogeneous populations and forms. In *Proceedings of IMAC XXXVIII – the 38th International Modal Analysis Conference, Houston, TX*, 2020.

[5] P.A. Gardner and K. Worden. Towards population-based structural health monitoring, Part IV: Heterogeneous populations, matching and transfer. In *Proceedings of IMAC XXXVIII – the 38th International Modal Analysis Conference, Houston, TX*, 2020.

[6] J. Gosliga and K. Worden. A general representation for assessing the similarity of structures. In *Proceedings of the 12th International Workshop on Structural Health Monitoring, Palo Alto, CA*, 2019.

[7] M. El-Mehalawi and R.Ãllen Miller. A database system of mechanical components based on geometric and topological similarity. part I: representation. *Computer Aided Design*, 35:83–94, 2001.

[8] M. El-Mehalawi and R.Ãllen Miller. A database system of mechanical compoennts based on geometric and topological similarity. part II: indexing, retrieval, matching, and similarity assessment. *Computer Aided Design*, 35:94–105, 2001.

[9] A. Rytter. *Vibrational Based Inspection of Civil Engineering Structures*. PhD thesis, Deptartment of Building Technology and Structural Engineering, Aalborg University, Denmark, 1993.

[10] J. Gosliga, P.A. Gardner, L.A. Bull, N. Dervilis, and K. Worden. Towards population-based structural health monitoring, Part III: Graphs, networks and communities. In *Proceedings of IMAC XXXVIII – the 38th International Modal Analysis Conference, Houston, TX*, 2020.