Book Section:
Wagg, D.J. orcid.org/0000-0002-7266-2105 (2024) Modelling, reductionism and the implications for digital twins. In: Touzé, C. and Frangi, A., (eds.) Model Order Reduction for Design, Analysis and Control of Nonlinear Vibratory Systems. CISM International Centre for Mechanical Sciences, CISM 614. Springer Cham, pp. 1-57. ISBN: 9783031674983. ISSN: 0254-1971. EISSN: 2309-3706.

https://doi.org/10.1007/978-3-031-67499-0_1

Modelling, Reductionism and the Implications for Digital Twins

David J Wagg*
University of Sheffield, S1 3JD, UK
The Alan Turing Institute, London, NW1 2DB, UK

Abstract In this Chapter we will discuss modelling and reductionism in science and engineering, and how this relates to the new idea of *digital twins*. In particular, we focus on the historical context of modelling and reductionism for dynamics and control of engineering systems. Both active and passive control methods will be discussed, including the novel ideas associated with the *inerter*. Based on a selected review of the philosophy of modelling, we consider the role of knowledge and complexity in model making. The related topics of systems engineering, uncertainty analysis and artificial intelligence are also briefly discussed in the context of digital twins. We will argue that utility, trust and insight are the three key properties of models that will ideally be extended to digital twins. We then consider how digital twins will require the *dynamic assembly* of digital objects in order to recreate emergent behaviours. In order to implement a digital twin, an operational platform is required. We briefly present an aircraft example of a digital twin operational platform. Lastly we consider digital twin knowledge models and ontologies, and how this topic might help shape digital twins in the future.

1 Introduction

> "To doubt everything, or, to believe everything, are two equally convenient solutions; both dispense with the necessity of reflection"
> — Henri Poincaré (see Gray, 2012).

*The author would like to acknowledge the support of UKRI via the grants EP/Y016289/1 & EP/R006768/1, The Alan Turing Institute Research and Innovation Cluster for Digital Twins (TIRC-DT) and the IOTICS Data for Good Programme. In addition, the author would also like to thank the DTNet+ Leadership Team members & wider community, the TRIC-DT Co-Directors & staff, Mark Girolami, Ben MacArther, Jason Shepherd, Mark Enzer, Rebecca Ward, Zack Xuereb Conti, Lawrence Bull, Tim Rogers, Nikos Dervilis, Matthew Bonney, Xiaoxue Shen, Ziad Gauch, Matthew Tipuric, Prajwal Devaraja, and Saeid Taghizadeh.

In this Chapter we will consider how reductionism and modelling have been used in the science and engineering associated with nonlinear systems. In particular, we will present an interpretation of this topic which looks forward to the development of *digital twins* for dynamical systems. For the purposes of our discussion we will consider that a digital twin is a virtual representation of a physical system, called the *physical twin* that evolves over time and is constructed from digitised information such as recorded data and the output of computational models.

Henri Poincaré can be considered as the founding father of the discipline of nonlinear systems. His initial work has lead to multiple fields of enquiry, most obviously the field of *dynamical systems theory*, that is also sometimes known by other names (e.g. during the 1970s and 80s it was known as *chaos theory*) — see for example Guckenheimer and Holmes (1983); Moon (1987); Glendinning (1994); Thompson and Stewart (2002); Strogatz (2019) for detailed overviews. However, Poincaré was more than just a brilliant mathematician, he was also a philosopher, and spent a considerable amount of time debating and discussing philosophical topics with others — as described in detail in the biography by Gray (2012). Therefore, we begin this Chapter with a quotation by Poincaré that reminds us of the importance of reflection, something that we will try to do in the discussion below. The quote from Poincaré also points out that extremes (or limiting cases) of an argument are often easy to adopt. The part in the middle, where most real-world applications lie, is the more difficult part to deal with, but is essential if we are to have relevance for physical applications.

1.1 Reductionism in Science and Engineering

The idea of reductionism in science and engineering is defined thus by Heylighen et al. (2007):

> "...to understand any complex phenomenon, you need to take it apart, i.e. reduce it to its individual components. If these are still complex, you need to take your analysis one step further, and look at their components. If you continue this subdivision long enough, you will end up with the smallest possible parts, the atoms (in the original meaning of "indivisibles"), or what we would now call "elementary particles". Particles can be seen as separate pieces of the same hard, permanent substance that is called matter."
>
> — Heylighen et al. (2007).

The concept of division in classical mechanics is based on the division of *material*, and so we say that the associated *ontology* is materialistic (e.g. related to physical matter). Ontology, is the branch of philosophy which examines the fundamental categories of things, and is becoming an important concept for digital twins (to be discussed later). Historically, the idea that some things, like the human mind, are *non-physical* extends back to the ideas of Greek philosophers, and *meta-physics* has become the established as the study of non-material phenomena. More specifically related to the human mind, Descartes developed the idea of *mind-body dualism* (Heylighen et al., 2007), which separates the physical matter of the brain from the (apparently) non-material human mind.

This reductionist approach has become the predominate method for creating models in scientific and engineering practices over history, and has been a highly successful approach. Reductionism has led to deterministic and mechanistic reduced problems to be used as models for a wide-range of applications, with a high degree of mathematical rigour. More than anyone else, this philosophy has become associated with Newton:

> "We are to admit no more causes of natural things than such as are both true and sufficient to explain their appearances. To this purpose the philosophers say that Nature does nothing in vain, and more is in vain when less will serve; for Nature is pleased with simplicity and affects not the pomp of superfluous causes."
> — Isaac Newton, Principia: The Mathematical Principles of Natural Philosophy (Newton, 1686).

In this classic quote Newton espouses the idea of avoiding "superfluous causes" and asserts that "Nature is pleased with simplicity" as an argument for simplification (and parsimony, which we discuss later).

Classical mechanics can be broadly divided into the study of solids, liquids and gases, with more advanced fields evolving to cover topics such as thermodynamics and electromagnetism. The overall reductionist approach is based on simplifying, when modelling apparently complicated physical processes. This method works particularly well for *ordered systems*, such as solid materials with lattice-like structures, or the dynamics of billiards (e.g. point-mass systems). In these cases the behaviours can be encoded into a set of deterministic "laws". To quote Descartes:

> "...reliable rules which are easy to apply, and such that if one follows them exactly, one will never take what is false to be true or fruitlessly expend one's mental efforts, but will gradually and constantly increase one's knowledge till one arrives at a true understanding of everything within one's capacity." — Descartes: Rules for the Direction of the Mind (reprinted; Descartes, 1985, first published in 1701).

Descartes is saying that not only can systems be reduced, but they can be explained by rules, and this is the underlying ethos of classical mechanics, where models are both *mechanistic* and *deterministic*. Our modern interpretation of determinism (in dynamical systems) has come to mean that the state of something in the future can be determined entirely from it's current state — an interpretation generally attributed to Laplace (2012) (first published in 1795).

The approach taken by classical mechanics didn't fit in the same way to *disordered systems*, such as a gas, consisting of molecules that act without any apparent constraints. To deal with this apparent disorder, a new field of study, called *statistical mechanics*, gradually developed, in which small particles (such as molecules in a gas) are treated statistically with probability theory and related techniques. This statistical approach allowed for average behaviours to be modelled, based on some basic assumptions about the independence of each particle and the nominally identical nature of the associated probability. Such simplifying assumptions allowed disordered systems to be analysed within an essentially mechanistic modelling framework as well. This was in large part due to the pioneering work of L. Boltzmann, J. Clerk Maxwell & J. W. Gibbs, and a modern introduction to the topic can be found, for example, in Pathria and Beale (2011).

Reductionism is very deeply embedded in much of science and engineering, for example the following techniques are all from a reductionist ethos:

1. Theory reductionism: One theory is reducible to another: e.g. Kepler's laws are reducible to Newtonian theories of mechanics.
2. Methodological reductionism: phenomena at one scale are determined by their underlying methods and processes.
3. Axiomatic reduction: Mathematics.
4. Atomistic reduction: The study of sub-atomic particles in physics.
5. Geometric reductionism: The separation of the parts of a system.
6. Separation of physical phenomena: Fluid flow, structural mechanics, thermodynamics etc.

However, it is also known that reductionist models, cannot represent the entire physical behaviour of the physical system. The difference between a reductionist model output and an observation is known as the *error* or *uncertainty* related to the model (Smith, 2013). For example, when a deterministic model does not capture the observed behaviour of the physical system, the model is considered to be "missing" some significant part of the physics. This missing knowledge is called the *model inadequacy* of the reduced model, (also called the *epistemic* uncertainty).

Regardless of uncertainty, there are other fundamental limitations with the reductionist approach. For example, in the 20th Century, new scientific ideas began to undermine the predominance of reductionist thinking. Specifically:

1. The Heisenberg uncertainty principle — a foundational concept in quantum mechanics that is philosophically different to classical mechanics.
2. Gödel incompleteness theorems — limits on the axiomatic reduction of mathematics.

Life sciences is a field of study for which reductionist modelling has long been recognised as not a good framework. This is particularly the case for *emergent behaviours*, which occur in a wide variety of biological systems, and have become particularly important when studying phenomena relating to the human mind, such as cognition, and intelligence (we discuss in more detail in Section 4).

Next we review the classic approach to mathematically modelling and controlling nonlinear dynamical systems.

1.2 Dynamics & Control of Nonlinear Systems

The study and understanding of nonlinear systems has been very important for the advancement of science and engineering. Phenomena exhibited by nonlinear systems are exhibited in real-world physical applications, but are not always possible to capture using linear (or linearised) modelling techniques. In many applications, there is a requirement to *control* the behaviour of the system in addition to understanding its nonlinear behaviour.

The history of control engineering stretches back to pre-industrial times, and an example of an early control system is shown schematically in Fig. 1. The idea of control is to adjust, regulate, or otherwise obtain a desirable response from a dynamic system as it evolves in time. For example, the governor system shown in Fig. 1 works for rotating machinery. As the central shaft rotates faster, the two masses move outwards which in turn lifts the collar and lever at the bottom of the shaft. The lever is connected to something that needs adjusting based on speed of rotation. For example, in windmills used to grind corn into flour, the gap between the grinding stones needed to be regulated to ensure the flour was evenly ground for all rotating speed. The governor can be tuned to achieve this purpose, and has a linkage which adjusts the gap between millstones to the desired distance.

As control technology developed, the vast majority of the associated control theory was developed for linear dynamical systems — see for example Goodwin et al. (2000); Inman (2006) and references therein. As the interest in nonlinear dynamical systems has grown, so has the interest in the control of such systems — see Nijmeijer and van der Schaft (1990); Slotine and Li (1991); Isidori (1995); Krstić et al. (1995); Sastry (1999); Fradkov

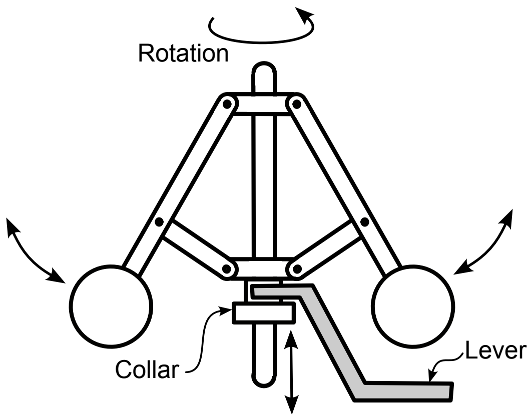


Figure 1. Schematic diagram showing an early governor system. These devices were used in windmills in the pre-industrial age to regulate the gap between millstones as the sails of the windmill rotated at different speeds.

et al. (1999); Wagg and Neild (2015). The classical approach to nonlinear

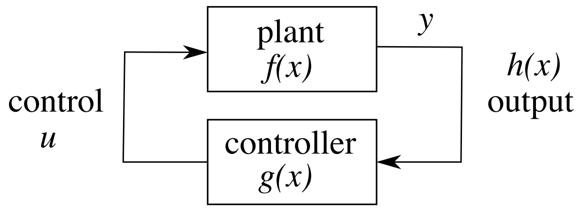


Figure 2. Nonlinear control schematic diagram.

control systems is shown schematically in Fig. 2. Here, the plant is the (dynamical) system to be controlled. The plant has a control input u and an output y, which in general are both vectors. The plant output is used by the controller to generate the next control input, and the system continues to evolve in time, t.

It is typical to define a mathematical model of the system in Fig. 2 as

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned} \tag{1}$$

where x is the state vector, $f(x)$ is the nonlinear function defining the plant behaviour, $g(x)$ is the nonlinear function defining the controller behaviour, and $h(x)$ is the nonlinear function defining the relationship between the output and the states.

There are many *hundreds* of books and papers that analyse this problem — see for example those mentioned above. Many different control design methods have been used for nonlinear systems including optimal and adaptive control methodologies.

This classical approach typically relies on “complete knowledge” of the functions f, g and h and assumed access to the states x. This is often not possible in practice. In addition to this, many other simplifications are often assumed, e.g. that the dynamics are deterministic. If noise or disturbance terms are included they are typically assumed to be stationary random signals that are also IID and Markovian (e.g. no memory). When designing a control system based on assumed *physics-based* models, then often multiple of these types of assumptions are needed to make the system modelling tractable.

It is an interesting point of note to compare classical nonlinear control to the much more recent technique of reinforcement learning (Graesser and Keng, 2019). The two concepts are shown schematically in Fig. 3. Here we

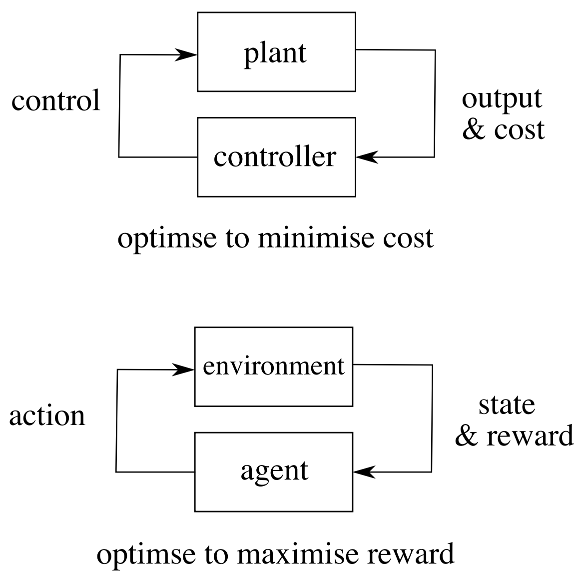


Figure 3. Nonlinear control (top) compared to reinforcement learning (bottom).

have introduced an optimal control version for the nonlinear control scheme (top diagram in Fig. 3) and the objective is to minimise cost. Conversely, in reinforcement learning, the actions are taken in an environment by an agent, and the objective is to maximise the reward for actions at each iteration.

In fact, control and reinforcement learning are two different (opposing) philosophical approaches to solve the same type of problem:

1. Nonlinear control is based on a physics-based (deterministic) approach: The problem can be modelled in a deterministic way using physics based models. All that is needed is to include all the relevant physics in the model(s) and increase the model fidelity until the required behaviour is obtained.
2. Reinforcement learning is based on a data-driven (dataist) approach: The problem can be solved by learning from large amounts of data using machine learning and associated methods. By increasing the amount of data covering all the relevant system behaviours, the required behaviour should be obtained.

Both these approaches have limitations and downsides. For example:

1. Physics-based (deterministic) approach: We need a lot of knowledge of the physics in advance. Lack of knowledge of the physics, or gradual changes over time, will compromise the reliability of this method.
2. Data-driven (dataist) approach: Even if enough data can be obtained, explainability, interpretability and reproducibility are often a major a problem. Fragility of learned models to a changing context is also a problem.

Later we will want to combine these opposing philosophical approaches when considering digital twins. This combination, will (ideally) maximise the strengths of both approaches, and minimise the limitations.

Next we consider how reduction may occur in control systems.

1.3 Reduction for Control

Control is normally about adding to a dynamical system. E.g. additional actuators, dampers or other devices to change behaviour However, reductions can happen in several ways for control system:

- Plant or controller model inadequacy, for example if there is a high degree of uncertainty about the physics of the plant, the assumed physics-based model may have a number of inadequacies.
- Transformations and truncations are often used, such as modal transformation — if the system is linear — or nonlinear normal models (or normal forms) see Chapter 2 of this book, when the systems is nonlinear.

- Process simplification, where the processes involved in the control actions are simplified to make the modelling tractable.

However there are consequences of reduction for control. These include:

- Incomplete or partial observations, which often happens for example when controlling flexible body structures, and the actuators and sensors cannot be co-located. The resulting non-collocation leads to partial observations and control actions.
- Control spillover, occurs for flexible body control systems, where untargeted models in the systems are excited by the control forces, and similarly observations are corrupted by the same process.
- Stiffness coupling often occurs for flexible body control, where actuators are attached to flexible structures leading to stiffness coupling.
- Actuator dynamics (delay particularly) can be a factor, and can have particular implications for the system under control — we will discuss a specific case later on that relates to a hybrid simulation system example.

Next we will review a classical example relating to nonlinear control.

1.4 Example, the Planar, Vertical, Take-off and Landing (PVTOL) Aircraft

We now consider a well known example of nonlinear control systems which is the planar, vertical, take-off and landing (PVTOL) aircraft, shown schematically in Fig. 4. Although this application was developed in the 1970s for military aircraft, it still retains relevance for applications such as the control of drones.

Here we follow the analysis in Sastry (1999) (see also references therein), from which the (scaled and simplified) equations of motion for the PVTOL aircraft are given by

$$\begin{aligned}
\ddot{x}_1 &= -\sin(\theta)u_1 + \varepsilon\cos(\theta)u_2 \\
\ddot{x}_2 &= \cos(\theta)u_1 + \varepsilon\sin(\theta)u_2 - 1 \\
\ddot{\theta} &= u_2
\end{aligned} \tag{2}$$

The states are x_1, x_2 and θ and the control inputs are u_1 and u_2. A scaled gravity term is represented by -1.

One method for solving this type of control problem is to use input-output linearisation. For this we need to find the Lie derivatives, which are obtained by differentiating the output, y with respect to time. This gives

$$\dot{y} = \frac{\partial h(x)}{\partial x}\frac{\partial x}{\partial t} = \frac{\partial h(x)}{\partial x}\dot{x}, \tag{3}$$

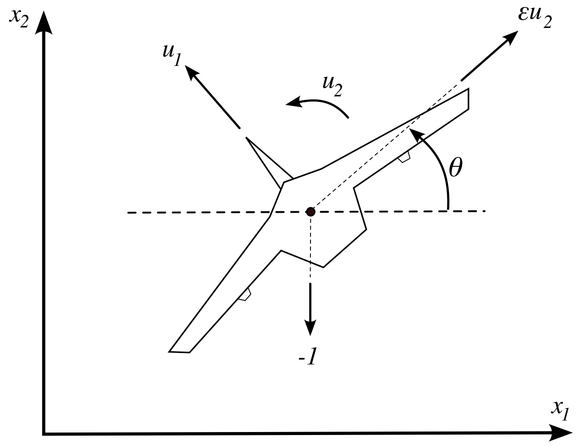


Figure 4. Nonlinear control example, the planar, vertical, take-off and landing (PVTOL) aircraft — see Sastry (1999).

where $x = [x_1, x_2, \theta]^T$. Substituting for $\dot{x}$ from Eq. (2) leads to

$$\dot{y} = \frac{\partial h(x)}{\partial x}(f(x) + g(x)u) = \frac{\partial h(x)}{\partial x}f(x) + \frac{\partial h(x)}{\partial x}g(x)u,$$

which can be rewritten as

$$\dot{y} = L_f h(x) + L_g h(x)u, \tag{4}$$

where $L_f h(x)$ and $L_g h(x)$ are the Lie derivatives of $h(x)$ with respect to $f(x)$ and $g(x)$.

To remove the system dynamics and replace them with a new control signal, $r(t)$, we choose a control input of the form

$$u = \frac{1}{L_g h(x)}(r(t) - L_f h(x)), \qquad L_g h(x) \neq 0, \tag{5}$$

to give $\dot{y} = r(t)$ which is a linear relationship between the input $r(t)$ and the output $\dot{y}$.

If the condition $L_g h(x) \neq 0$ is true, the system is said to have relative degree one and no more differentiation is required. However, if the output does not appear directly in the expression $L_g h(x) = 0$, we need to take the Lie derivative again. This will be the case in this example in the PVTOL system, where $\ddot{y}$ is required as the outputs are in terms of accelerations. So when $L_g h(x) = 0$ differentiate Eq. (4) again to give

$$\ddot{y} = \frac{\partial}{\partial x}(L_f h(x))\frac{\partial x}{\partial t} = \frac{\partial L_f h(x)}{\partial x} f(x) + \frac{\partial L_f h(x)}{\partial x} g(x) u = L_f^2 h(x) + L_g L_f h(x) u. \tag{6}$$

In this case, if $L_g L_f h(x) \neq 0$, the system is said to have relative degree two, and the control law is given by

$$u = \frac{1}{L_g L_f h(x)}(r(t) - L_f^2 h(x)), \qquad L_g L_f h(x) \neq 0. \tag{7}$$

Now take the system output to be $y = [x_1, x_2]^T$ because we want to control the hover of the aircraft. Then using Eq. (6) we have

$$\ddot{y} = L_f^2 h(x) + L_g L_f h(x) u \quad \rightsquigarrow$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} + \begin{bmatrix} -\sin(\theta) & \varepsilon \cos(\theta) \\ \cos(\theta) & \varepsilon \sin(\theta) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{8}$$

from which we can infer what $L_f^2 h(x)$ and $L_g L_f h(x)$ are, and substitute them into Eq. (7) to obtain the control signal. Now

$$\frac{1}{L_g L_f h(x)} = \begin{bmatrix} -\sin(\theta) & \varepsilon \cos(\theta) \\ \cos(\theta) & \varepsilon \sin(\theta) \end{bmatrix}^{-1} = \begin{bmatrix} -\sin(\theta) & \cos(\theta) \\ \frac{\cos(\theta)}{\varepsilon} & \frac{\sin(\theta)}{\varepsilon} \end{bmatrix} \tag{9}$$

and so

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} -\sin(\theta) & \cos(\theta) \\ \frac{\cos(\theta)}{\varepsilon} & \frac{\sin(\theta)}{\varepsilon} \end{bmatrix} \left(\begin{bmatrix} r_1 \\ r_2 \end{bmatrix} - \begin{bmatrix} 0 \\ -1 \end{bmatrix} \right) \tag{10}$$

Substituting for the control signals, u_1 and u_2 into Eq. (2) gives

$$\begin{aligned} \ddot{x}_1 &= r_1 \\ \ddot{x}_2 &= r_2 \\ \ddot{\theta} &= \tfrac{1}{\varepsilon}(\sin(\theta) + \cos(\theta) r_1 + \sin(\theta) r_2) \end{aligned} \tag{11}$$

Unfortunately this system has unstable zero dynamics, which makes this method an impractical approach for the control. To see the effect of the zero dynamics we substitute $r_1 = r_2 = 0$ in Eq. (11) which leaves

$$\ddot{\theta} = \frac{1}{\varepsilon} \sin(\theta) \tag{12}$$

which represents the part of the system dynamics that are uncontrollable. The dynamics of Eq. (12) is the same as dynamics that govern the *undamped* pendulum (with a saddle at the origin rather than a centre). In this case they correspond to the rolling and rocking motions of the aircraft, and as this can occur regardless of the control inputs, this situation is highly undesirable, and is therefore not a practical solution.

Many alternative methods of solution have been suggested that can avoid this problem, such as linearising and decoupling the planar degrees-of-freedom — for more details see the discussion in Sastry (1999). This approach may seem somewhat unsatisfactory, but we recall the quote at the beginning of the chapter from Poincaré about the need to deal with the 'inconvenient' realities of such problems.

Next we move on to consider how a different type of control mechanism can be used to take actions in the real-world.

2 Passive Control of Reduced & Low-Order Systems

> "While in 1934 a mechanical engineer was considered well-educated without knowing anything about vibration, now such knowledge is an important requirement"
> — J. P. Den Hartog, 1956 (see Den Hartog, 1934).

Thus far we have discussed what is known as *active* control systems, where actuators (or other mechanisms) are used to affect change in the plant via the control signal u. Passive control systems are the opposite of this. They do not use any actuators, instead the system of interest is *redesigned* to reduce the unwanted vibrations (or other behaviour), typically by the addition of a specific passive device(s) — where passive in this context means not actively adding energy into the system.

2.1 The Tuned-Mass-Damper

The most important and well known passive control device is the *tuned-mass-damper* (TMD) which was patented by Hermann Frahm (1909). The TMD concept is shown in Fig.5 (a). Here the primary (or host) system is the system we want to control that has parameters of mass M, stiffness k and zero (or very small) damping. To apply a passive control affect a smaller oscillator, called the tuned-mass-damper, is attached to the primary system — as shown in Fig. 5 (a). This smaller oscillator has parameters mass m_a, damping c_a and stiffness k_a.

In Fig. 5 (b) the response of the primary system both with and without the TMD is shown. The primary system simulation has a small amount of damping in this case, and the excitation across the frequency range is sinusoidal.

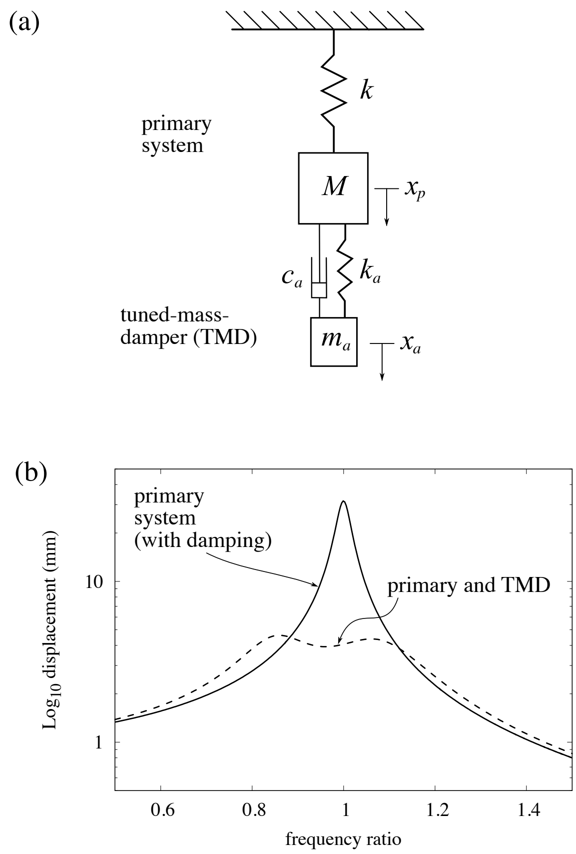


Figure 5. The tuned-mass-damper showing (a) a schematic diagram of the primary system, M, k, with the absorber, m_a, k_a, c_a, attached. (b) A simulation of the (damped) primary system without the absorber attached (solid line), and tuned-mass-damper (dashed line) subjected to sinusoidal excitation $F\sin(\Omega t)$ where $F = 1$N. The frequency ratio is Ω/ω_p where $\omega_p = \sqrt{k/M}$.

Following the work by Frahm, there where many refinements of the design process for the TMD. However, the basic design of the parameters for the TMD can be obtained using the "fixed point" method of Brock (1946). Using Brock's method, the design parameters for the simulation shown in Fig. 5 (b) were based on the following values $k = 1000$N/m, $M = 100$kg, (and $c = 10$kg/s i.e. damping "close to zero") and $m_a = 8$kg. Then the mass ratio, $\mu = m_a/M$, is calculated so that the TMD stiffness can be found via

$$k_a = \frac{k\mu}{(1+\mu)^2}$$

and damping ratio from

$$\zeta = \sqrt{\frac{3\mu}{(1+\mu)^3}}$$

which enables the calculation of the required TMD device damping as $c_a = 2\zeta m_a \omega_p$. The results of the TMD simulation are shown as the dashed line in Fig. 5 (b).

2.2 Vibration Isolation

The TMD is an example of vibration *absorption*. The other main type of passive vibration control is to design a vibration *isolation* system. Schematic examples of these basic passive vibration control concepts are shown in Fig. 6, where the excitation is from a base motion input (e.g. support motion). A vibration absorber is shown schematically in Fig. 6 (a), whereas two types of vibration isolator are shown in Figs. 6 (b) and (c).

Considering the vibration isolator system in Fig. 6 (b), the equation of motion for the case when the spring is linear can be written as

$$m\ddot{x} + c(\dot{x} - \dot{r}) + k(x - r) = 0 \tag{13}$$

where m is the mass in kg, k is stiffness in N/m and c is the damping in kg/s. This equation can be solved for the case when the excitation input, r is sinusoidal motion of the form $r = R\sin(\omega t)$ (full details can be found in Chapter 3 of Wagg and Neild (2015)).

The response of the linear vibration isolator is measured using transmissibility, which is the ratio of the response amplitude, X divided by the input amplitude R such that

$$\left|\frac{X}{R}\right| = \sqrt{\frac{(1 + 4\zeta^2(\frac{\Omega}{\omega_n})^2)}{(1 - (\frac{\Omega}{\omega_n})^2)^2 + 4\zeta^2(\frac{\Omega}{\omega_n})^2}}, \tag{14}$$

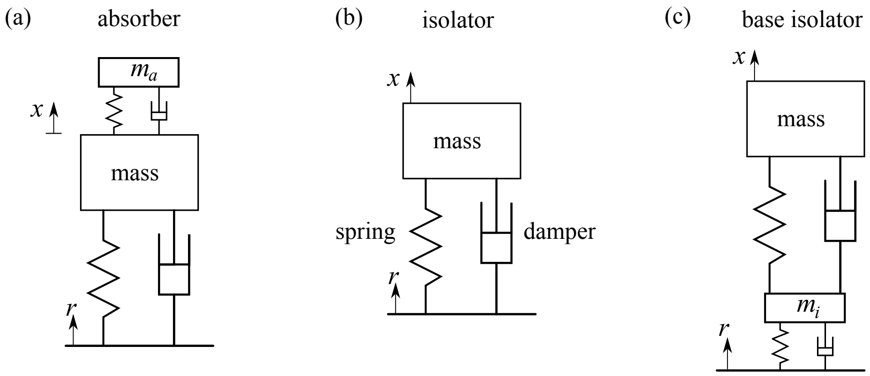


Figure 6. Three types of passive control with base motion input, showing (a) a vibration absorber, (b) a vibration isolator, and (c) a base isolation system.

and phase lag is

$$\phi = \arctan\left(\frac{2\zeta(\frac{\Omega}{\omega_n})^3}{(1-(\frac{\Omega}{\omega_n})^2)+4\zeta^2(\frac{\Omega}{\omega_n})^2}\right), \tag{15}$$

where $\omega_n = \sqrt{k/m}$ is the natural frequency and $\zeta = c/2m\omega_n$ is the damping ratio. An example for a linear passive vibration isolator is shown in Fig. 7, where (a) shows the transmissibility $|X/R|$, and (b) the phase shift, ϕ. Also shown in Fig. 7 is the affect of varying the viscous damping ration, ζ.

In the case where the isolator spring is nonlinear, then there are additional parameters that can be used to design the transmissibility response. For example, one type of nonlinear passive vibration isolator uses a high static, low dynamic stiffness function, and the equation of motion is given by

$$m\ddot{x} + c(\dot{x}-\dot{r}) + k(x-r) + k_3(x-r)^3 = 0 \tag{16}$$

where, in this case, a nonlinear cubic stiffness has been included in addition to the linear stiffness.

The combination of a linear and a nonlinear stiffness can be used to design a *quasi-zero stiffness* (solid line in Fig. 8 (a)) that is a combination of linear and nonlinear stiffnesses (Shaw et al., 2013b). The transmissibility response of a nonlinear vibration isolator with quasi-zero stiffness, compared to a linear vibration isolator is shown in Fig. 8 (b) (Shaw et al., 2013a). It can be seen for this example, that the affect of the nonlinear stiffness term is to increase the isolation region.

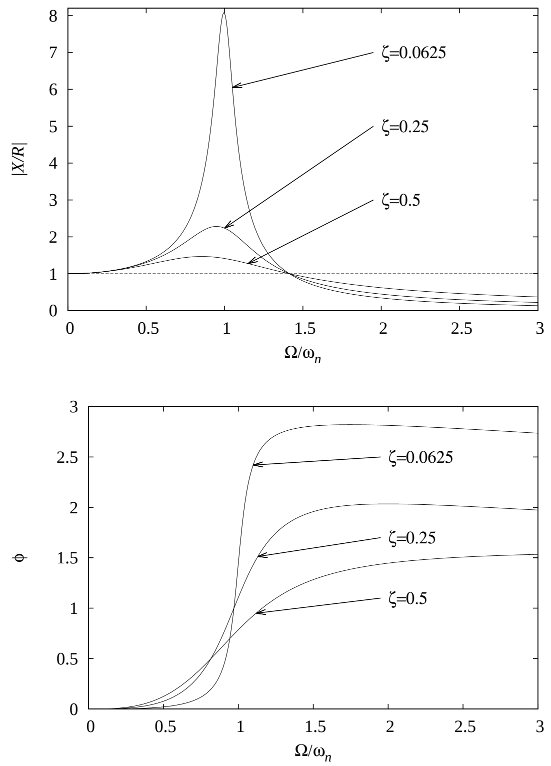


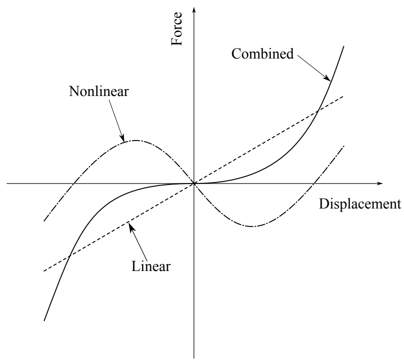
Figure 7. Steady state response of linear vibration isolator showing, (a) displacement amplitude divided by input amplitude, and (b) phase shift of response compared to input.

In addition to adding masses, or designing stiffness or damping parameters, in recent years another concept has come to light. This is the idea of the *inerter*, and we discuss this next.

2.3 The Inerter

The inerter was a term defined by Malcolm Smith (Cambridge), to complete the force-current analogy between mechanical and electrical networks (Smith, 2002). In fact, inerter type devices had been in use for many previous decades, but just known by other names. For example the dynamic antiresonant vibration isolator (or DAVI) was developed in the 1960s — see

(a)



(b)

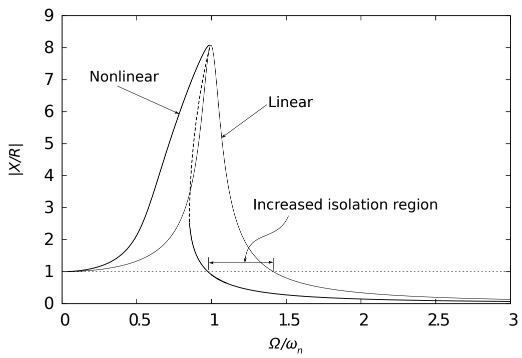


Figure 8. Nonlinear vibration isolator showing, (a) the quasi-zero stiffness (solid line) that is a combination of linear and nonlinear stiffnesses, and (b) transmissibility response (displacement amplitude divided by input amplitude) for both linear an nonlinear passive vibration isolator examples.

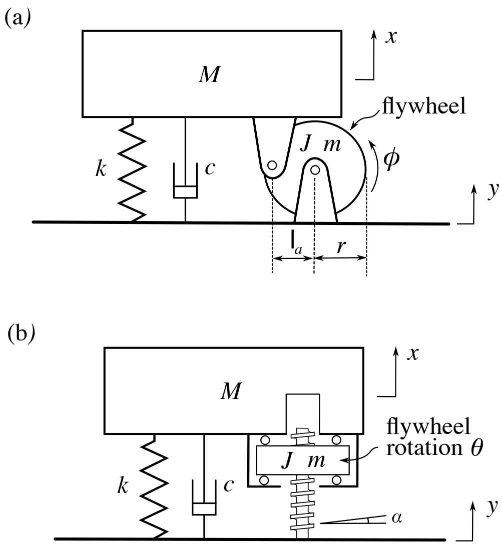


Figure 9. Schematic diagrams of the flywheel inerter showing (a) the case where flywheel is directly connected, and (b) the flywheel operates on a threaded rod, often referred to as a “ball-screw” system.

Wagg (2021) and references therein for more details.

One of the simplest ways to design an inerter device is to use a mechanical flywheel. Two variations of this idea are shown in Fig. 9, where in (a) the flywheel is directly connected, and in (b) the flywheel operates on a threaded rod, often referred to as a “ball-screw” system. Inerters can be used to reduce vibrations by using them to create devices that enable vibration isolation and absorption. This is typically done assuming linear behaviour, to design a series of inerter-based devices that mitigate vibrations. The most well-known devices of this type are the tuned-viscous-mass-damper (TVMD) (see Ikago et al. (2012)), the tuned-inerter-damper (TID) (proposed by Lazar et al. (2014)), and the tuned-mass-damper-inerter (TMDI) — see Marian and Giaralis (2014). These three inerter-based devices (e.g. the TVMD, TID & TMDI) and multiple variants have been applied to a wide range of applications and examples in engineering — see for example Wagg (2021) and references therein.

An example of the characteristics of a tuned-inerter device is shown in Fig. 10 where the tuned-inerter-damper applied to a 3-storey structure (dot-dash line), is compared with the uncontrolled system (dashed line) and a

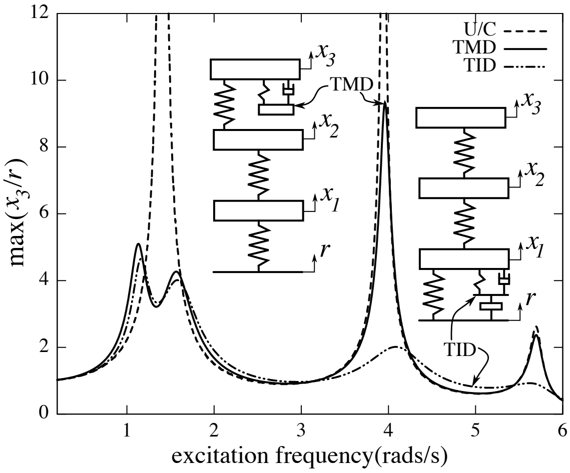


Figure 10. Tuned-inerter device example, showing the response of a tuned-inerter-damper applied to a 3-storey structure (dot-dash line), and compared with the uncontrolled system (U/C, dashed line) and a tuned-mass-damper (solid line).

tuned-mass-damper response (solid line). Note that the optimum position for the TID is at the bottom of the structure, which is the opposite of the TMD which is optimally positioned at the top of the structure.

The vibration mitigation affect of the TMD and TID are very similar for the first (e.g. lowest frequency) resonance peak. However, it is important to notice that the TID is a relative motion device, and so unlike the TMD it affects all modes, and the response of second and third resonance peaks are also reduced by the TID.

Inerters have also been applied to nonlinear vibration systems as well, and an example is shown in Fig. 11. In this example a nonlinear quasi-zero-spring is used in a vibration isolator design where mass, M, is to be isolated from input $y(t)$, and the spring has a quasi-zero nonlinear restoring force function, $f(z)$, where $z = x - y$. The equation of motion for the resulting nonlinear quasi-zero & inerter isolator shown in Fig. 11 (a) can be written as

$$(m + b)\ddot{z} + c\dot{z} + f(z) = -m\ddot{y}, \tag{17}$$

where z is the relative displacement $z = x-y$. We can "design" the nonlinear stiffness function to be an odd polynomial in z such that

$$f(z) = k_1 z + k_3 z^3 + k_5 z^5, \tag{18}$$

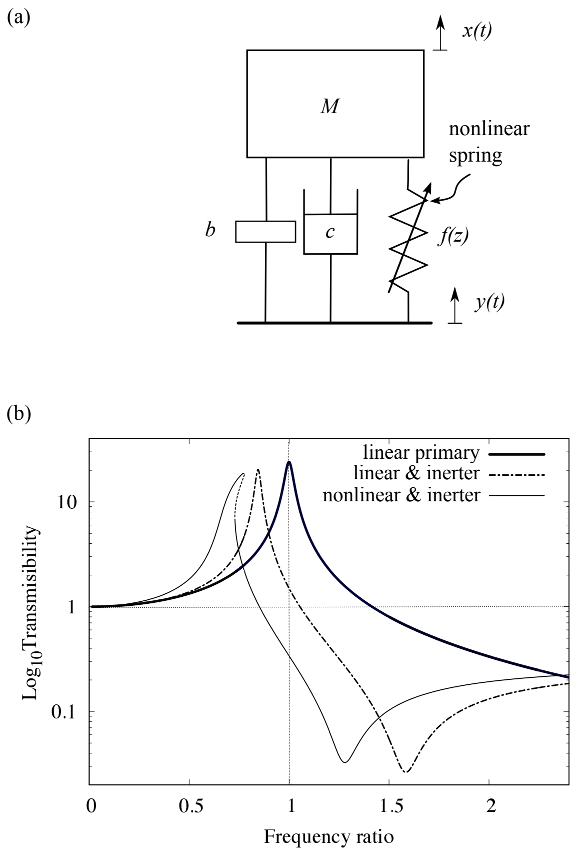


Figure 11. Nonlinear inerter isolator example, showing (a) a schematic diagram of a nonlinear inerter isolator, and (b) a comparison of linear- and nonlinear-inerter isolators.

So, assuming k_1 is fixed, we now have three parameters, k_3, k_5 and b with which to "tune" the desired behaviour.

Results from an example system are shown in Fig. 11 (b), where the unisolated linear primary system with $M = 7.2$kg, linear spring $k = 80$N/m, $c = 1.0$Ns/m subject to a support displacement of $y = 0.04\sin(\Omega t)$ m and $b = 0$ (thick solid line) can be compared to the linear spring plus inerter case with the same parameters except inertance $b = 2.9$kg so that $\mu = b/M = 0.4$ (dot-dash line). In addition, the quasi-zero plus inerter case with the same parameters as the previous case except a nonlinear quasi-zero spring function designed using the method in Shaw et al. (2013a) with $k_s = 52$N/m & $\hat{z}_r = 0.577$ is used (thin solid line). Note that for this simulation curve the thin dashed line represents the part of the solution branch that is unstable. It is clear from Fig. 11 (b), that using both inerter and nonlinear stiffness parameters, enables the antiresonance to be moved closer to the position of the linear primary system's original resonant peak.

Having considered the concepts of control and passive redesign to reduce vibrations, we now return to the idea of modelling and reductionism for digital twins.

3 Modelling, Reductionism and Complexity

> "All models are wrong, some are useful," — George Box (Box, 1982).

> "No one trusts a model except the man who wrote it; everyone trusts an observation except the man who made it," — Harlow Shapley.

The quotes from George Box and Harlow Shapley introduce some important ideas for modelling. The context for Box's comment comes from a discussion regarding the level of validation a model can have when compared to the real world system (Vining, 2013). Box's main point is that no (statistical) model can ever be "correct" in the sense that there is a "perfect" match with the physical system. Box's statement also introduces the idea of usefulness (or utility) of a model, and that models can have a *useful purpose even though they can never be perfect.*

Shapley introduces the idea of trust which is in practice linked to uncertainty. Shapley's quote also reveals two human biases; (i) the tendency for humans to trust observations over a model, and (ii) the difference between model makers (and data collectors) and users. Why might this be important? What is the "useful purpose" of a model?

It will be argued here that a primary useful purpose of a model is to gain (or enhance, extend, and/or clarify) knowledge. Furthermore, the statements above are key to understanding the limitations of *all* theoretical and computational models. In the authors opinion, these limitations are broadly aligned to the idea of "model dependent realism" expressed by Hawking & Mlodinow's Grand Design (Hawking and Mlodinow, 2010).

Model dependent realism is based on the philosophy that absolutely certainty is an unobtainable goal, and therefore the most important thing is the usefulness of the model. In the context of digital twins, discussed later, the usefulness will be particularly relevant in terms of *explanatory capability.* More specifically, for both models and digital twins, we will contend that the primary useful purpose is to gain (or enhance, extend, and/or clarify) knowledge/insights that will ultimately lead to explanatory capability. We also acknowledge (following Shapley) that unbiased and trustworthy models (and digital twins) are crucial, alongside utility, in order to gain this new knowledge and insight.

Therefore we claim that *utility, trust and insight* are the three key *generic* requirements (or properties) of models that we would like to extend to digital twins. How about other important characteristics like fidelity, parsimony, cost or optimality? For digital twins, we argue that these characteristics will depend on the specific *context* of the model (or digital twin). The context means the specific application, objectives and other details relating to the physical system under consideration. It is important to emphasise that our ultimate aim is to create digital twins that are of course *not* models (at least in the most direct interpretation — explanation given later). Therefore, characteristics like fidelity, parsimony, cost or optimality will be considered to be context dependent, whereas utility, trust and insight are *generic.*

That said, parsimony relates to simplification and therefore reductionism, so we will consider that next. Firstly the parsimony principle states that: a simpler model with fewer parameters is favoured over more complex models with more parameters, provided the models fit the data similarly well. This follows the advice of many prominent scientists, for example, didn't Einstein say

> "Everything should be made as simple as possible, but no simpler"?

In fact, it's difficult to find the source of this quote, more likely he said:

> "It can scarcely be denied that the supreme goal of all theory is to make the irreducible basic elements as simple and as few as possible without having to surrender the adequate represen-

tation of a single datum of experience." — Einstein lecture 1933 (Einstein, 1934).

For more complicated systems, and particularly living systems, cognitive science and AI research, there is some evidence from that points to the opposite conclusion. For example:

> "AI researchers were beginning to suspect — reluctantly, for it violated the scientific canon of parsimony—that intelligence might very well be based on the ability to use large amounts of diverse knowledge in different ways," — Pamela McCorduck 2004 (McCorduck, 2004).

Therefore, if we want to try and recreate some of these more complex and sophisticated physical phenomena, it is unlikely that reductionism will be a sufficient tool to help us. Something further would be needed. We will now take a more detailed look at the process of making models for scientific and engineering applications.

3.1 The Model Making Process

A flow diagram of a model making process is shown schematically in Fig. 12. In this example, the first part of the model making process is to make observations from a physical system. This obviously assumes that there is already a physical systems in existence, which is often *not* the case in engineering, e.g. when designing something not previously built. However, for the purposes of this discussion, we assume that the physical system is available for observation. These observations are then used to make a model based on a set of assumptions, and the assumptions capture all the

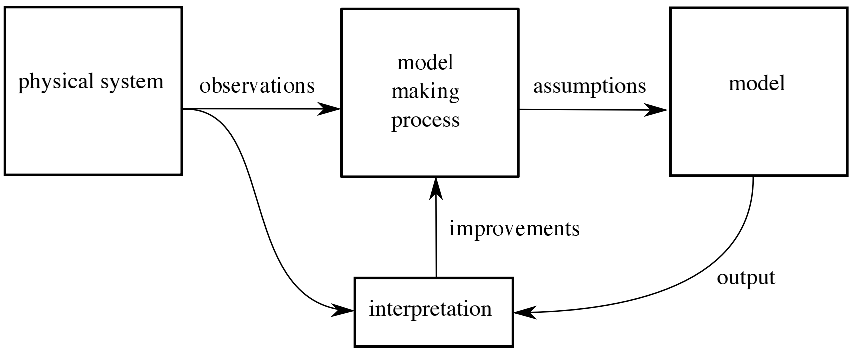


Figure 12. Schematic diagram showing the typical method of making a model of a physical system.

reductions, simplifications, and other approximations made. The resulting output(s) from the model are then interpreted, and typically this leads to improvements being made to the model, before the process is repeated.

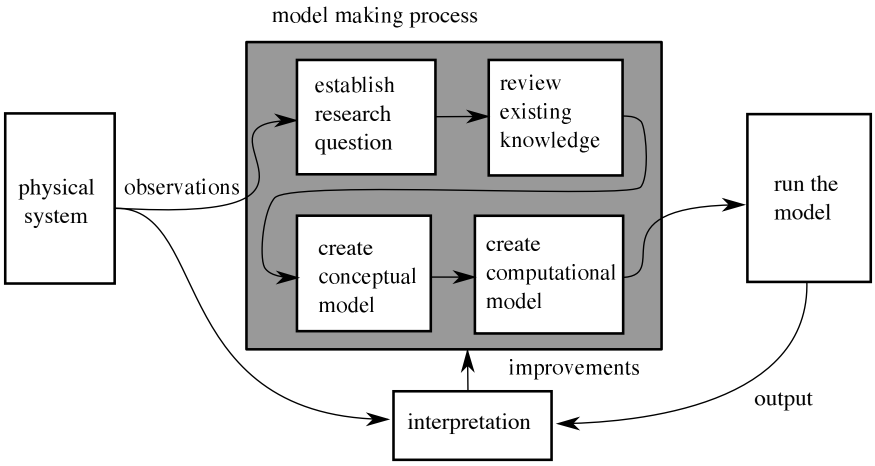


Figure 13. Schematic diagram showing the typical method of making a model of a physical system for a computational model.

It will be important to our later discussion to consider the actual model making process in more detail, and an example of this is shown in Fig. 13. Although there are multiple different ways to approach the model making process, the steps shown in Fig. 13 capture the most important parts of the process.

First, there is a requirement to establish an overall research question that the model is designed to answer. This will typically require expert knowledge and understanding of the problem at hand in order to formulate a meaningful question to answer. Next, or often as part of the same process, a review of the existing knowledge would typically be carried out, which also requires a high level of expertise and research skills. Existing knowledge also includes a review of existing modelling assumptions that can be used to inform the assumptions that are made for the current modelling process.

Typically in an engineering setting, the next step is to create a conceptual model, as shown in Fig. 13, then assumptions will need to be made at this point in order to define the precise form of the conceptual model. This process of assumption and refinement will continue into the last stage of developing the computational model, and it would be expected that further assumptions, either explicit or implicit would be adopted at this stage.

Note that this discussion of model making is not intended to be comprehensive in any way, and readers are referred to Oberkampf and Roy (2010) for a comprehensive overview of modelling in the scientific and engineering domains. That said, there are several important points that can be understood from the (simplified) model-making process shown in Fig. 13 that will be important for our later discussion on digital twins. In particular, the role of knowledge and expertise in the process of creating the model and making the associated assumptions.

This is particularly important when trying to address systems with significant complexity, which is a topic we address next.

3.2 Complexity in Engineering Systems

> "Engineering is the art of modelling materials we do not wholly understand, into shapes we cannot precisely analyse, so as to withstand forces we cannot properly assess, in such a way that the public has no reason to suspect the extent of our ignorance"
> — Dr. A. R. Dykes, from the British Institution of Structural Engineers President's Address,1978.

In engineering practice there are requirements to design, build, commission, operate, maintain, manage and decommission a wide range of different engineering systems. The quotation from A. R. Dykes gives a sense of the trade-offs necessary in the engineering process. It's typical that multiple complex and uncertain factors, e.g. the materials, geometries, forces and public expectations, have to be combined in order to achieve an engineering task. A list of some of the types of complex (and/or complicated) behaviours that occur in engineering applications in Table 1.

Typically, most engineering applications have more than one type of complexity contained within it from the list in Table 1. Furthermore, engineers are often required to create something new, or deal with a socio-technical system that is highly complex/uncertain and is changing over time. This is in contrast to the scientific approach, where the focus is on understanding and explaining the behaviour we observe (as in complexity science). In order to address the challenges of complexity, *systems engineering* has become an important methodology, and we discuss this next.

3.3 Systems, Uncertainty and Artificial Intelligence

Systems engineering has been developed alongside the related fields of systems research and complexity throughout the later part of the 20th Century (Schlager, 1956). The systems engineering approach has now developed

Table 1. Some examples of complex (and/or complicated) behaviours that occur in engineering applications.

Type	Examples (non-exhaustive list)
Environmental	Temperature, pressure humidity & climatic effects; physical location; geographical effects
Geometric	Multiple compliments of varied shape & geometries; joints and jointing between components; mechanisms & interactions
Material	The physical & chemical properties of matter; combined & composite materials; wear, ageing & damage
Behavioural	Mechanistic behaviour of solids & fluids; vibrations & time-dependent behaviours; emergent behaviour; multi-physics; length-scales
Operational	Control & feedback; updates & changes; faults & failures; networks & connectivity; computational hardware & software
Computational	deterministic vs non-deterministic; time & memory requirements; processing resources; data size & formats, Kolmogorov complexity
Processes	Design; decisions & interventions; sequencing & workflow; human behaviour; communications; heuristics
Organisational	Structure & hierarchies; practices & organisation culture; rewards & incentives
Social	Attitudes; motivations; culture; education level; religion; beliefs; gender etc.

into a well established methodology for dealing with complex engineering projects (Walden et al., 2015; Hirshorn et al., 2017). The early developments were driven in large part by NASA and the space programme, and space engineering continues to be an important application area for the further development of the methods (Hirshorn et al., 2017). State-of-the-art systems engineering is built largely on the concept of *processes*, to enable the design, implementation and management of a specific engineering application or project.

The systems engineering framework enables interlinked uncertainties and complexities to be managed simultaneously, and for the technical processes to be aligned with the decision, management and wider related business processes. Systems engineering also takes account of the lifecycle of a system, in addition to requirements analysis and hierarchies of systems that lead to systems-of-systems applications (Adams and Meyers, 2011).

When dealing with complex system, it is important to acknowledge uncertainties involved in both physical observations and computer models. In terms of classifying uncertainties, we follow Kennedy and O'Hagan (2001), who define the following:

- Parameter uncertainties: computer models contain of parameters which may be measurable (in which case there is parametric variability) but in most cases are not fully known or accessible.
- Model discrepancy (or inadequacy): following the famous quote by Box it is understood that there will always be mismatches between the model output and the physical process.
- Residual variability: given the same set of inputs the process may produce different outputs, due to stochastic processes (or in some circumstances deterministic chaos).
- Parametric variability: outputs may vary because inputs cannot be fully controlled or specified.
- Observational uncertainty: measuring any real world system will result in a level of measurement error or noise.
- Code uncertainty: uncertainty associated with the computer code.

The process of uncertainty quantification requires the measurement (or estimation) of these different sources in order to quantify uncertainty for predictions in a given context. It's also possible to categorise uncertainties into the classifications of aleatoric and epistemic, depending on the philosophical approach.

The processes of quantifying uncertainties in models may be achieved via a variety of approaches. However, a key distinguishing feature of a digital-twin is that it evolves over time. This will mean that any uncertainty quantification technique may need to operate in, or close to, real time — a large constraint on many current technologies. A digital-twin could require an offline process of uncertainty quantification whilst also having some component of online parameter estimation.

Artificial intelligence (AI) began with the development of formal logical methods and the early attempts to create mechanical computation machines — see for example Nilsson (2009); Russell and Norvig (2010); Haenlein and Kaplan (2019); Marcus (2020) and references therein. Early developments led by Alan Turing (Turing, 1950) were a starting point for the current research field, and the name *artificial intelligence* came from a meeting at Dartmouth in 1956 organised by John McCarthy.

The drive to create AI (e.g. Nilsson (2009)) has multiple different, interlinked approaches. Those include the desire to replicate human intelligence and other biological examples, attempts to create intelligent machines, and using AI to solve applied problems. The unifying theme (if there is one) appears to be the use of *intelligent agents* (Russell and Norvig, 2010).

For example, deep reinforcement learning, where agents are used to solve *sequential decision-making problems*, such as autonomous driving vehicles (Kiran et al., 2021) is a current topic of interest that overlaps with digital twins concepts, and was mentioned above when we discussed nonlinear control. Such methods can be used to address sequential decision-making problems, which are also highly relevant to digital twins.

Another categorisation of different components of AI research is to distinguish between:

1. symbolic AI, such as logical reasoning, knowledge models and expert systems (Krishnamoorthy and Rajeev, 2018);
2. sub-symbolic AI (connectionism), which includes all types of machine learning (ML) (Bishop, 2006);
3. neuro-symbolic AI, which is the fusion of the other two categories (Dingli and Farrugia, 2023).

Broadly speaking, it could be said that symbolic AI was the earliest to develop leading to applications such as expert systems (Krishnamoorthy and Rajeev, 2018). This has more recently been eclipsed by sub-symbolic which is not the dominant force in AI, particularly deep learning (LeCun et al., 2015) and most recently large language models (Teubner et al., 2023). Recently, some AI experts have been pointing out the limitations of sub-symbolic AI, (Marcus, 2018), and promoted the idea of combining the two approaches in the form of neuro-symbolic AI. Note that this is a overly simplistic summary, but readers who are interested can find more detail in the associated references.

In summary, we can reflect on three trends from the 20th Century — since the time of Poincaré. Firstly, following the development of quantum mechanics, the philosophy of science underwent a major shift in perspective, resulting in far less certainty of what can be "proven" objectively — scientists had to reassess the Newtonian worldview. Secondly, the 20th Century saw the development of computational power that has given birth to high-powered software models that have surpassed all previous human capacities to simulate the physical world. Lastly, the rise of data-driven techniques, particularly machine learning, have provided far greater inferential capacity without corresponding explanatory power. Thus, it would appear that one of the legacies of the 20th Century has been to *leave us with greater philosophical uncertainty about models, but far greater capacity to compute and infer from them!*

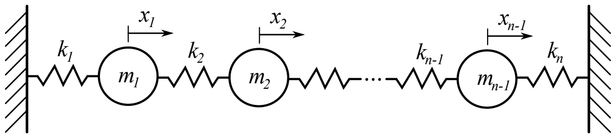


Figure 14. A simple chain of masses and springs, similar to that used by Enrico Fermi and collaborators to model the interactions of molecules in a crystal.

We now consider an example of an important computational development from the 20th Century.

3.4 Example: The Fermi-Pasta-Ulam-Tsingou Paradox

In the early 1950s a group of researchers working at Los Alamos led by Enrico Fermi decided to study the problem of molecular dynamics using numerical computations. This was a new idea, that would take advantage of the newly available Maniac1 computer that had recently been commissioned at Los Alamos. The idea was to use the computer to test the theories of molecular physics developed via statistical mechanics.

For example, in statistical mechanics, it had already been well established that molecules in a crystal lattice could be approximately represented as oscillating masses with springs between them representing the molecular interaction forces. If these forces were linear, then the theory applicable for a multi-degree-of freedom linear (undamped, unforced) systems can be applied. This states that energy put into a single (linear) vibration mode of the systems will remain in that mode for all time. However, if the springs were weakly nonlinear, the energy would gradually redistribute (or equipartition) into all the modes as time increased.

Fermi with his co-workers, wanted to test this assumption using the Maniac1 computer. So they built a simulation that used an initial condition of all vibration energy in the lowest (frequency) mode of vibration, and included weakly nonlinear coupling from the springs. Initially they observed what they expected, however, they also found that if the simulation was run for long enough, the energy flowed back out of all the other modes into the mode where it started.

This phenomena was completely surprising to the research team at the time, and was not what they expected at all. It was called the Fermi-Pasta-Ulam (FPU) paradox (Weissert, 1999; Berman and Izrailev, 2005; Dauxois, 2008), and gave rise to a large field of research, particularly for Hamiltonian dynamical systems. Note that it has been more recently recognised that

Mary Tsingou made significant contributions to the work, and so now it is called the Fermi-Pasta-Ulam-Tsingou paradox.

This work was also one of the first pioneering activities in scientific computation, or otherwise now known as numerical simulation. This powerful methodology has transformed the way we perform science and engineering computations right up to the present day.

4 Assembly, Emergence and Anti-Reductionism

> "The axiomization and algebraization of mathematics after more than 50 years has led to the illegibility of such a large number of texts that the threat of complete loss of contact with physics and the natural sciences has been realised," — Vladimir Arnold. (1988).

In order to create digital twins, we will need to be able to "assemble" models together. There are several possible ways to do this, and we will present one example of how this might be achieved in the next subsection.

4.1 Assembly of Models

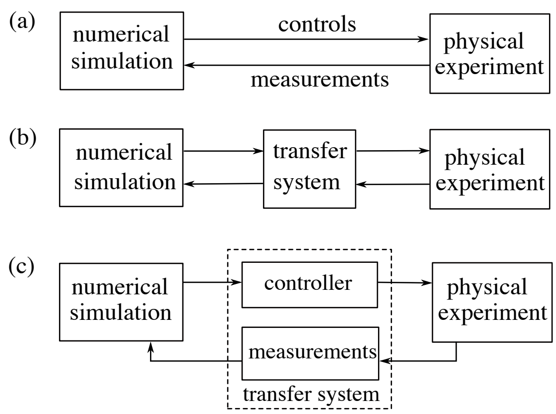


Figure 15. Schematic diagram showing (a) the underlying concept of hybrid simulation, (b) the use of a transfer system, and (c) the two main components in the transfer system.

One way to assemble models is to use a technique called hybrid sim-

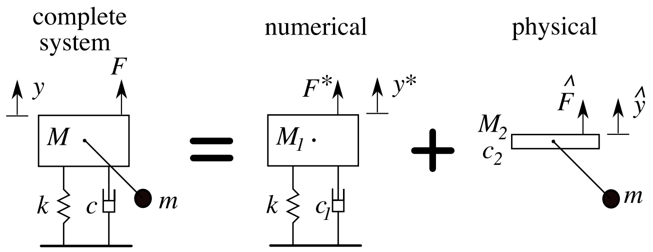


Figure 16. Assembly of models: Spring-mass-pendulum example: system is shown (labeled as “complete system” on the left of the subfigure) which is decomposed into the addition of two subsystems (labelled as “numerical” and ”physical” on the right of the subfigure).

ulation[1]. Hybrid simulation is a technique where a physical experiment and numerical simulation are combined using control & data acquisition hardware, typically in real time — see for example Wallace et al. (2005a); Carrion (2007); Carrion et al. (2009); Chen and Ricles (2009); Gao et al. (2013); Tsokanas et al. (2021) and references therein. The concept is shown schematically in Fig. 15.

In Fig. 15 (a) the idea of hybrid simulation is shown, where a physical experiment and a numerical simulation are combined in real-time using control algorithms and measured observations. For most physical experiments, a *transfer system* is required to achieve this as shown schematically in Fig. 15 (b) and (c). It is important to notice that the objective in hybrid simulation is to get the transfer system to act like an *identity transformation* between the two systems being connected.

To consider a simple illustrative example of hybrid simulation, in Fig. 16 we show a mass-spring-damper-pendulum example originally developed in Gonzalez-Buelga et al. (2005). The complete system (on the left of Fig. 16) is the mass-spring-damper-pendulum system. The idea is that the nonlinear part is “difficult” to model because it is nonlinear (in this example the pendulum), and is therefore taken to be the physical experiment part, because then all of the nonlinear physics will be captured (e.g. no assumptions are taken to make a model of this part). It should be remembered that this is a toy problem used to show the concept, in fact the pendulum is not that difficult to model. The pendulum is labelled as “physical” in Fig. 16. The

[1] Also known by numerous other names such as hybrid testing, hardware-in-the-loop, real-time dynamic substructuring, and pseudo-dynamic testing.

remaining linear part is modelled numerically (labelled as "numerical" in Fig. 16) and is assumed to be relatively "easy" to model, because it is linear.

During hybrid simulation, the numerical system output, y^*, is used as the setpoint in a control algorithm that controls the input to the physical system so that $\hat{y}$ tracks y^*. At the same time, the physical force, $\hat{F}$, from the experiment is measured and feedback to be applied in the next computation of the numerical model. To remove the effects of latency in the control and measurement hardware, a delay compensation scheme is typically used.

Measuring the synchronisation errors, $|\hat{y} - y^*|$, shown in Fig. 17 (b), can be used as a measure of the fidelity of the hybrid solution, (and ideally we want $|\hat{y} - y^*| \to 0$ (Wallace et al., 2005a)). It is interesting to note that the complete system output, y only "exists" during the hybrid simulation. In other words, "reconstructed" outputs from the assembled system only exist whilst the control algorithm is working to connect the two systems together such that $\hat{y} \to y^* \to y$ (and $\hat{F} \to F^* \to F$.). Without the control system connection, the output of the two systems would not reconstruct the desired outputs of the combined system (e.g. $\hat{y} \neq y^* \neq y$).

4.2 Emergent Behaviours

> "What does this mean? That the essential reality of a system is indescribable?...Or does it mean, as it seems to me, that we must accept the idea that reality is only interaction?" — Carlo Rovelli (2016).

In the previous section we showed an example of two systems being "assembled" to reconstruct the dynamical behaviour of the combined system. For such a simple example, the reconstructed dynamic behaviour that resulted from the assembly process was also a well known type of behaviour. In the example shown above (in Fig. 17) the *interaction* consisted of synchronising variables from the two subsystems. This is a type of time dependent emergent behaviour, generally a subset of evolutionary dynamics known as *synchronisation* (Jensen, 2022).

As well as synchronisation, there are other types of emergent behaviour, and multiple authors have described how different types might be categorised — see for example Ashby (1956); Holland (2007); Frei and Serugendo (2012); Fernández et al. (2014); Holland (2018); Tadić (2019); Jensen (2022) and references therein. Broadly speaking, the types of emergent behaviours range from types of *self-organisation*, (Jensen, 2022), through to *evolutionary* forms of emergence (Kauffman, 2000). The ability to simulate emergent behaviours is a significant capability that is seen as a very desirable functionality (Gershenson, 2013), including for digital twins. Here

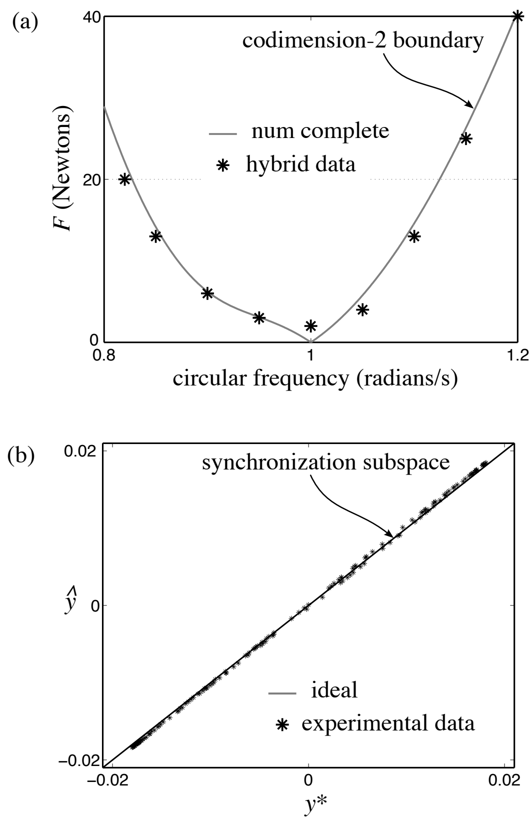


Figure 17. Assembly of models: Spring-mass-pendulum example: The hybrid numerical-physical system is used to reconstruct a nonlinear codimenson-2 bifurcation boundary. This behaviour is only exhibited by the reconstructed output of the combined system, and the resulting behaviour is shown in panel (a) where a complete simulation of the combined system (solid line labelled as "num complete") is shown in comparison to the hybrid simulation data (stars, labelled as "hybrid data"). The control algorithm is configured to ensure that $\hat{y}$ tracks y^{*}, and that if they are synchronised as closely as possible then the hybrid system will reconstruct the required behaviour to some level of fidelity. The "synchronisation subspace" for the test in subfigure (a) is shown in subfigure (b). For full details of these and related results see Gonzalez-Buelga et al. (2005); Kyrychko et al. (2006); Gonzalez-Buelga et al. (2007); Gawthrop et al. (2009).

(a)

Type of system behaviour

What is predicted

	Desired	Undesired
Predicted	Predicted Desired	Predicted Undesired
Unpredicted	Unpredicted Desired	Undesired Unpredicted

(b)

What could be known

What is known

	Knowns	Unknowns
Known	Known Knowns	Known Unknowns
Unknown	Unknown Knowns	Unknown Unknowns

Figure 18. Schematic diagram showing (a) how the outputs from a digital twin might be able to predict emergent behaviours proposed by Grieves and Vickers (2017), and (b) the "Rumsfeld" matrix.

we are interested in how a digital twin might be expected to produce such behaviours, especially for very complicated applications. In the example in Fig. 16 we *knew in advance* what behaviour to expect, and could therefore validate the hybrid result quite easily (e.g. the validation between a complete numerical computation and the hybrid system results is shown in Fig. 17 (a)). However, what happens in cases where we cannot know what to expect in advance?

Work by Grieves and Vickers (2017) studied how the outputs from a digital twin might be used to predict emergent behaviours. Grieves & Vickers proposed a categorisation of outcomes for the digital twin that is shown in Fig. 18 (a), where there are four categories of outcome. Each category depends on the digital twin prediction and whether the predicted behaviour was desirable in a design context (meaning the intended design) or undesirable (problematic and/or unwanted designs). This method is then used iteratively to try and minimise the undesirable and unpredicted aspects as much as possible.

Unfortunately, this approach also suffers from the problem of the need to know in advance what to include in the digital twin to get a desired outcome. Kauffman (2000) for example, has pointed out that this need to know things in advance is a particular problem in the field of emergent behaviour. In fact, problems relating to prior knowledge are well known in other fields, for example in the domain of uncertainty and risk management (e.g. Okashah and Goldwater (1994); Lanza (2000)). The "Rumsfeld" matrix (made famous by Donald Rumsfeld in 2002 and an adaptation of the Johari window) captures the key issue as shown in Fig. 18 (b).

The Rumsfeld matrix defines four categories based on what *is* known (e.g. meaning what is known at this present time) and what *could be* known (e.g. all possible knowledge, if there was a way to access it). Clearly, if something is not known at the present moment, then it cannot be included it in our digital twin, and therefore (using this type of framework) the "unknown unknowns" category can never be accessed. Note that the unknown unknowns category which is associated with so-called *black swans* (Taleb, 2007; Aven, 2013). Knowing in advance is a practical necessity for modelling, but will therefore exclude the more advanced behaviours such as evolutionary forms of emergence — see for example Kauffman (2000); Tononi et al. (2016) and references therein.

One way of trying to mitigate this limitation, could be including real-time data, but the same constraints apply. If we have never experienced an event before, it won't be in any of our previously recorded data sets, or associated data-based models. This principle would include the most recent data-driven computations such as the Deepmind AlphaGo algorithm (Silver et al., 2016; Chouard, 2016). In our interpretation, even new learned behaviours of these types of simulations would fall into the unknown knowns category.

Now, let us consider what can be reasonably expected from a digital twin in terms of emergent behaviours. We propose that an important property for a digital twin will be so-called *object-property inheritance*. E.g. if a digital twin has a model as one of its components (e.g, the model is the object), then the digital twin will directly inherit (at least) some of the properties of that model. In other words, *a digital twin is something more than a model, but can be used to perform functions that have been previously carried out using models.*

Object-property inheritance can be interpreted as both related to individual components (objects) in the digital twin, and relational combinations of the components. The relational combinations of the components are achieved using *component-connector interactions*, all of which we assume are prescribed in advance.

As a result, if a digital twin consists of n objects it would have a number (say d) of *directly inherited properties* which come from the n objects without any interactions. In addition, there would have a combinatoric number (say r) of *relational properties*, including any emergent behaviours, which are generated from the *component-connector interactions* (and where the combinatoric metric is chosen based on the context of the digital twin).

An example for three components and three connectors is shown in Fig. 19, where a simple graphical model is used to represent the component-connector interactions. Directly inherited properties are shown to come from the components, and relational properties come from the connectors. Both direct and relational properties can be then used as digital twin outputs.

It should be emphasised that all the emergent (and non-emergent) behaviours in the digital twin outputs will fall into the categories of *known knowns*, *known unknowns*, and *unknown knowns*, shown in Fig. 18 (b). The *unknown unknowns*, shown in Fig. 18 (b) *are not accessible to the digital twin by definition.*

As a result, assuming that the known knowns category is already well understood, it is the known unknowns, and particularly the unknown knowns categories where value can be obtained from using a digital twin.

4.3 Anti-Reductionism or Holism

> "The whole is greater than the sum of the parts," — Aristotle.

Holism is the opposite of reductionism, where the physical system is not reduced but instead treated as a whole. This concept is attributed to Aristotle, and the quote above shows where this thinking originated.

For engineering systems, the holistic approach has been developed primarily through the field of *systems engineering* with inputs from other subjects, such as complexity science (Waldrop, 1993) and artificial intelligence (Russell and Norvig, 2010). Systems engineering defines a hierarchy of systems staring with "closed" systems that can be modelled using deterministic (Newtonian) mathematical models. Next is the possibility of closed systems-of-systems, when many deterministic systems can interact with each other. Beyond closed systems are "open" complex systems, such as biological or social systems, where the complexity of the underlying processes cannot be represented by closed, mechanistic models.

For complex interacting systems, emergent behaviours can be induced by interactions between different parts of the overall system. Emergent behaviours have also been central to the topic of complexity theory, which (typically) uses coupled systems of dynamic models acting as "agents" to create models of emergent behaviours — typically in a deterministic sense (Jensen, 2022), although a non-deterministic framework can also be adopted. Sys-

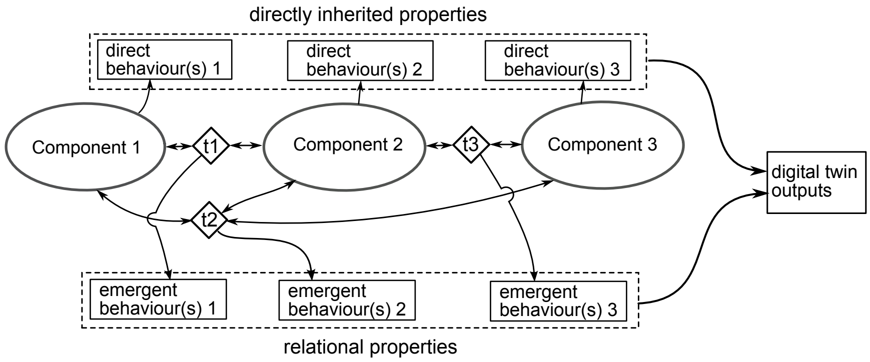


Figure 19. Schematic diagram showing how the outputs from a digital twin might be created using three components. In this case the Components 1 & 2 are connected by connector $t1$, and Components 2 & 3 are connected by connector $t3$. All three components are connected together with connector $t2$. Note that this is an arbitrarily chosen series of connections, there are other configurations. The directly inherited properties come from each of the three components and are grouped together. The relational properties, such as any emergent behaviours, come from the connections, $t1, t2$ and $t3$, and are also grouped together. Both the directly inherited and relational properties can be used to form digital twin outputs. Note also that the design and initialisation of the digital twins is omitted from this schematic diagram.

tems engineering has already been discussed in the context of digital twins — see for example Madni et al. (2019).

Engineers make extensive use of numerical simulation tools that are element-based (or similar) that essentially break up complex geometries and behaviours into an assemblage of simpler elements for which the behaviour can be defined. These methodologies, such as the finite element method and computational fluid dynamics, have evolved into sophisticated tools that are widely used to simulate the behaviour of complex/complicated systems.

Although we tend not to think in these terms, the outputs from element-based methods are emergent behaviours. Typically, field quantities such as stress, displacement or temperature are approximated as a form of "self-organisation" within the element-based method. Or in other words, the overall field behaviour arises from local interactions between the elements. A simple example is shown in Fig. 20.

The other domain where emergent behaviour is often used in an engineering context is when agent-based models are used. Examples of agent-based modelling include, passenger flow through transport hubs, such as railway stations or airport terminals (Abar et al., 2017).

Digital twins are anti-reductionism by definition, and the quote from Aristotle captures the key idea: some important phenomena cannot be captured by considering just the component parts. That said, there is a philosophical question about whether assembling a series of interacting parts will capture all the holistic behaviours — this is an open question which is strongly context dependent.

4.4 Importance of Data: Bayesian Calibration Example

"The goal is to turn data into information and information into insight", former Hewlett-Packard CEO, Carly Fiorina, 2004.

In recent years it has become much easier to record, store and accumulate data — leading to the concept of so-called "big data" (Chen et al., 2014). This has led to a huge growth in data-driven models, especially machine learning methods and statistical inference (Bishop, 2006; Girolami, 2011). Data-based models can be used for tasks such as regression, clustering, calibration and classification. Data-driven models are typically "black-box" models and cannot be used to extrapolate beyond the regime in which they were trained.

Despite these limitations, there is huge value to be obtained from data-based models, as the quote from Carly Fiorina indicates. Many of the advancements in engineering in recent years are built of the increasing availability of data. Key to the understanding of data-based models are statistical models, and in the next subsection we shown a short example of how model calibration can be used to compensate for inadequacy (e.g. lack of

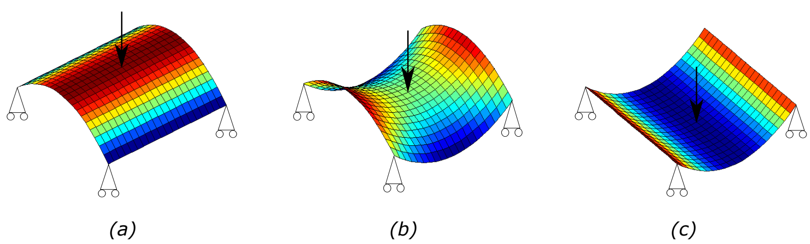


Figure 20. Finite element simulation of a bistable plate, showing the displacement fields of (a) & (c) the stable states and (b) the unstable state.

knowledge) in a physics based model.

4.5 Example: Model Calibration Due to Lack of Knowledge

The importance of data-based modelling is demonstrated here using a simple numerical (toy) example of a mass, tension wire system, shown schematically in Fig. 21, which was originally developed in Wagg et al. (2020a). The objective is to create a model that can predict the natural frequency of the system f (Hz), for a range of different tensions T in the wire. To reflect the concept of model discrepancy it is assumed that the "true" observed physical system has an off-centred mass where, L=1.1m, a=0.2m and m = 5.1kg (Fig. 21 (a)). However, the model of the system does not include the ability in incorporate an offset, instead modelling the system with a centred mass, representing a level of missing physics (Fig. 21 (b)).

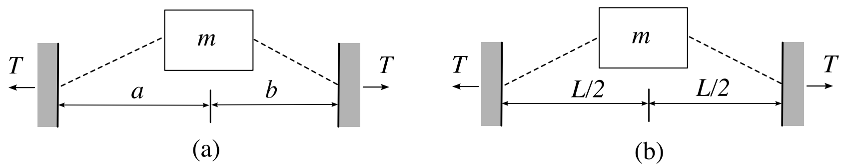


Figure 21. Mass, tensioned wire system schematic. Panel (a) the 'true' system; off-centred mass, tensioned wire, and (b) shows the model assuming a centred mass, tensioned wire. Parameters assumed are L=1m and a=0.2m.

$$f_{phys} = \frac{1}{2\pi}\left(\frac{T(a+b)}{mab}\right)^{\frac{1}{2}} \qquad \text{and} \qquad f_{model} = \frac{1}{\pi}\left(\frac{T}{mL}\right)^{\frac{1}{2}} \tag{19}$$

As a result when observations are made the data generated is that of f_{obs} in Eq. (19). Whereas, when the model is used, the results obtained are from f_{model} in Eq. (19). Even if the mass parameter is the same in both cases, because of the differences in the expressions for f in Eq. (19). The results are shown in Fig. 22 where a small amount of Gaussian noise has been added to the observation, so that $f_{obs} = f_{phys}+e$ with $e \sim \mathcal{N}(0, 0.01^2)$.

This raises the question of how to *calibrate* the model in order to align the model results with the observed data. There are multiple ways to do this — see for example discussions in Oberkampf and Roy (2010); Arendt et al. (2012). One approach is to express the problem in the form of a statistical model for the ith observation as

$$f_{obs,i}(T_i) = f_{phys,i}(T_i) + e_i = f_{model,i}(T_i, \boldsymbol{\theta}) + \delta_i(T_i) + e_i \tag{20}$$

where $\boldsymbol{\theta}$ is a vector of the physical parameters (and/or hyperparameters, depending on the specific methods being used (Arendt et al., 2012)) and δ_i is a discrepancy function that can be used to compensate for differences between the observations and the model.

There are a couple of "naive" approaches to solving this problem. First we can ignore the discrepancy by setting $\delta_i = 0$ and use the parameters (or hyperparameter) to get $f_{model} = f_{obs}$. This often leads to non-physical parameter values and other issues. Secondly we can ignore the effect of the parameters (in some cases, like this toy example we know what they are exactly) and use δ_i as a calibration function. For example, treating the components of Eq. (20) as random variables and then by taking the expected values over a suitably large range of observations we obtain

$$E[f_{obs,i}(T_i)] = E[f_{model,i}(T_i, \boldsymbol{\theta}) + \delta_i(T_i)] \tag{21}$$

assuming $E[e_i] = 0$, which is considered a reasonable assumption in many cases. From this we can compute a discrepancy function $\delta(T)$ for all the T values, and this is the result shown in Fig. 22.

Obviously, more realistic applications do not tend to lend themselves to these types of simplistic approaches, and it is more typical that both the parameters and the discrepancy needs to be used in order to calibrate the models. In those cases a range of approaches are applicable including Bayesian calibration (see for a Bayesian treatment of this example Wagg et al. (2020a)) and Gaussian Processes Brynjarsdóttir and O'Hagan (2014); Gardner et al. (2020), amongst other methods Oberkampf and Roy (2010); Arendt et al. (2012).

5 Digital Twins

> "It ought to be remembered that there is nothing more difficult to take in hand, more perilous to conduct, or more uncertain in its success, than to take the lead in the introduction of a new order of things" — Niccol Machiavelli, The Prince, 1532.

Digital twins have been promoted as a way to solve multiple problems. For example, by increasing our ability to understand engineering and other complex systems at previously unmatched levels of performance. As the quote from Machiavelli reminds us, introducing something new is often a difficult thing to do and there is a need to be cautiously pragmatic about the new ideas related to digital twins.

The aspiration for digital twins is being set very high but digital twins cannot somehow overcome the fundamental challenges and limitations related to modelling that we have discussed above. As a result, the concept

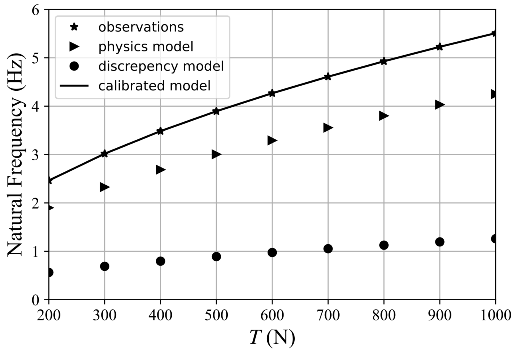


Figure 22. Mass tension wire system example. Showing the model discrepancy between the computer model and observed system when the incorrect model is used.

of digital twins is open to a very wide range of interpretations and in some cases it is being overhyped, causing frustration and scepticism at times. Therefore, it is important to understand the philosophical context which underpins the concept, something that has recently been proposed by Wagg et al. (2024).

A question that is often asked is: What is the difference between a model and a digital twin? There are several answers, but one that is often given is to make the following distinctions between three concepts (see for example Kritzinger et al. (2018));

1. Digital model — no automatic link between virtual and physical parts
2. Digital shadow — virtual part automatically updates to match the state of the physical part
3. Digital twin — two-way interaction between the virtual or physical parts

Another way to answer the question relates to the idea of object-property inheritance. For example, a digital twin can contain a model as one of its components (e.g, the model is the object), and as a result the digital twin will directly inherit (at least) some of the properties of that model. In other words, a digital twin is something more than a model, but can be used to perform functions that have been previously carried out using models. An example of this type of inheritance was shown in Fig. 19.

5.1 Digital Twin Output Functions

We now briefly described the idea of creating digital twin output functions that can be used as one way to mathematically represent a digital twin (Edington et al., 2023).

The first task is to identify a finite set of $N_{\mathbf{z}}$ quantities of interest (QoI), that can be observed from the physical twin, and these will form a vector $\mathbf{z} \in \mathbb{R}^{N_{\mathbf{z}} \times 1}$. The selection of QoIs will depend on the specific application, but would typically be measured physical quantities such as displacement, velocity, force, strain, voltage or temperature.

In order to observe the time evolution of the digital twin, the n^{th} QoI, $z_i^{(n)} \in \mathbf{z}$, (for $n = 1, 2, 3 \ldots N_{\mathbf{z}}$) element of $\mathbf{z}$ is sampled at time step i from the physical twin, and there is a corresponding discrete time series, $t_i \in [t_{start}, t_{end}]$, with a fixed time-step of Δt. The digital twin will use a specified combination of physics-based, data-based, and/or hybrid models to compute an approximation to the state of the physical twin QoIs at a particular time instant, t_i, and the output of the digital twin will be given by

$$\mathbf{y} = \boldsymbol{\eta}(\mathcal{M}, \mathcal{D}, \mathcal{T}, \boldsymbol{\chi}, t_i) \qquad \text{or} \qquad \begin{bmatrix} y_i^{(1)} \\ y_i^{(2)} \\ \vdots \\ y_i^{(n)} \\ \vdots \\ y_i^{(N_{\mathbf{z}})} \end{bmatrix} = \begin{bmatrix} \eta_i^{(1)}(\mathcal{M}, \mathcal{D}, \mathcal{T}, \boldsymbol{\chi}, t_i) \\ \eta_i^{(2)}(\mathcal{M}, \mathcal{D}, \mathcal{T}, \boldsymbol{\chi}, t_i) \\ \vdots \\ \eta_i^{(n)}(\mathcal{M}, \mathcal{D}, \mathcal{T}, \boldsymbol{\chi}, t_i) \\ \vdots \\ \eta_i^{(N_{\mathbf{z}})}(\mathcal{M}, \mathcal{D}, \mathcal{T}, \boldsymbol{\chi}, t_i) \end{bmatrix} \tag{22}$$

where $y_i^{(n)} \in \mathbf{y}$ is the n^{th} scalar output, and $\eta_i^{(n)} \in \boldsymbol{\eta}$ is the corresponding *digital twin output function*. Each digital twin output function is assumed to be a function of one or more of the N_p models $M_p \in \mathcal{M}$, where $\mathcal{M}$ is the model library containing all physics-based, data-based and hybrid (e.g. grey-box) models. Data-based and grey-box models are dependent on data sets that are contained in the library of data sets for the digital twin denoted $\mathcal{D}$. The time-based parameters $\{t_{start}, t_{end}, \Delta t\}$ are contained in the time-base library, $\mathcal{T}$, and all the hyper-parameters for the digital twin (defined below) are contained in the vector $\boldsymbol{\chi}$.

The outputs of each of the p *dynamic models*, M_p, are represented by

$$\hat{y}_{i,p} = \hat{\eta}_{i,p}(\mathbf{x}_i, t_i; \boldsymbol{\theta}_p, \mathbf{u}_p, D_p, \boldsymbol{\psi}_p) \qquad \text{for } p = 1, 2, 3 \ldots N_p \tag{23}$$

where $\hat{y}_{i,p}$ is the p^{th} scalar model output at time t_i, and $\hat{\eta}_{i,p}$ is the corresponding *model output function*. Each model output is a function of the state vector $\mathbf{x}$, the physical parameters vector $\boldsymbol{\theta}_p$, the input/control signals vector $\boldsymbol{u}_p$ and time, t_i. In addition, any datasets required by the model are contained in D_p, and if model hyperparameters are needed, they are contained in the vector $\boldsymbol{\psi}_p$. Although parameters and inputs can vary between models, for simplicity here we assume that all models are assumed to have the same state vector, $\mathbf{x}$, although model specific state vectors can be implemented if required. Note that the index notation here relates to the i^{th}observation of physical twin QoI z, and so $\mathbf{x}_i$ defines the i^{th} iteration of the state vector $\{\mathbf{x} : x_k \in \mathbf{x}\}_i$ for $k = 1, 2, 3....N_k$ where N_k is the total number of states.

Depending on the specific details of the application, there could be just one set of N_p model output functions that are used to build *all* $N_{\mathbf{z}}$ digital twin output functions, or there could be one set of N_p model output functions for *each* of the digital twin output functions. Furthermore, it is possible for models in the model library to be coupled together to simulate interactions, and capture emergent behaviours. In this case, we would expect the model outputs of the model *combinations* to be used in the digital twin output functions. In the case of model combinations, we would normally expect the number of model outputs to be less than N_p. Likewise, if there is a model selection process, where models are tested for suitability, and then the best chosen, then the number of model outputs will be less than N_p. As a result, the set model output functions can be written in vector form (using Eq. (23)) as

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_{i,1} \\ \hat{y}_{i,2} \\ \vdots \\ \hat{y}_{i,p} \\ \vdots \\ \hat{y}_{i,N_p} \end{bmatrix} = \begin{bmatrix} \hat{\eta}_{i,1}(\mathbf{x}_i, t_i; \boldsymbol{\theta}_1, \mathbf{u}_1, D_p, \boldsymbol{\psi}_1) \\ \hat{\eta}_{i,2}(\mathbf{x}_i, t_i; \boldsymbol{\theta}_2, \mathbf{u}_2, D_p, \boldsymbol{\psi}_2) \\ \vdots \\ \hat{\eta}_{i,p}(\mathbf{x}_i, t_i; \boldsymbol{\theta}_p, \mathbf{u}_p, D_p, \boldsymbol{\psi}_p) \\ \vdots \\ \hat{\eta}_{i,N_p}(\mathbf{x}_i, t_i; \boldsymbol{\theta}_{N_p}, \mathbf{u}_{N_p}, D_{N_p}, \boldsymbol{\psi}_{N_p}) \end{bmatrix} \tag{24}$$

5.2 The Statistical Model

When data is available, the QoI's from the physical twin and the outputs of the digital twin are related via a *statistical model* (Kennedy and O'Hagan, 2001; Smith, 2013; Arendt et al., 2012). In fact, we chose a statistical model that allows for *calibration* of the digital twin output functions, as will be explained below. Note also that in this formulation we exclude interactions between models. Instead models are augmented together, or selected, using a weighted sum process.

For the i^{th}observation of the n^{th} QoI, z_i^n, of the physical twin, the statistical model for the digital twin (omitting the n superscripts), can be defined as

$$z_i = \zeta_i + e_i = \eta_i(\mathcal{M}, \mathcal{D}, \mathcal{T}, \boldsymbol{\chi}, t_i) + \delta_i + e_i, \tag{25}$$

where the (unobservable) true process of the physical twin is ζ_i, and e_i is the observation (e.g. measurement) error (note the similarity with Eq. (20)). The digital twin output function η_i at the i^{th}observation computed using (22) and (23) is the best estimate of the QoI at time t_i. The inadequacy (or deficiency) of η_i is represented by δ_i. It should be noted that there is an implicit assumption in (25) that the digital twin inadequacy is separable in an additive way, and it is recognised that there may be other more appropriate ways of representing this relationship depending on the specific application. In some interpretations δ_i and e_i are combined into a single term (e.g. Kapteyn et al. (2020)) or additional error terms are included (e.g. Ward et al. (2020)).

When z_i data is available, the statistical model (25) can be used in the context of a time-evolving digital twin to relate (compare) the QoIs from the physical twin with the digital twin output functions whilst also accounting for the most important errors and uncertainties present in the problem. Specifically these errors and uncertainties are:

- observation (e.g. measurement) error is represented by e_i
- model form error (e.g epistemic uncertainties in the models) are assumed to be captured by δ_i
- parameter uncertainties can be included in the physical parameter vectors $\boldsymbol{\theta}_p$
- numerical errors (sometimes treated as a separate term e.g. Ward et al. (2020)) are assumed to be captured by δ_i

As the ζ_i quantities cannot be known directly we will omit them from the subsequent analysis, and then Eq. (25) can be written in a vector form as

$$\begin{bmatrix} z_i^{(1)} \\ z_i^{(2)} \\ \vdots \\ z_i^{(n)} \\ \vdots \\ z_i^{(N_\mathbf{z})} \end{bmatrix} = \begin{bmatrix} \eta_i^{(1)}(\mathcal{M}, \mathcal{D}, \mathcal{T}, \boldsymbol{\chi}, t_i) \\ \eta_i^{(2)}(\mathcal{M}, \mathcal{D}, \mathcal{T}, \boldsymbol{\chi}, t_i) \\ \vdots \\ \eta_i^{(n)}(\mathcal{M}, \mathcal{D}, \mathcal{T}, \boldsymbol{\chi}, t_i) \\ \vdots \\ \eta_i^{(N_\mathbf{z})}(\mathcal{M}, \mathcal{D}, \mathcal{T}, \boldsymbol{\chi}, t_i) \end{bmatrix} + \begin{bmatrix} \delta_i^{(1)} \\ \delta_i^{(2)} \\ \vdots \\ \delta_i^{(n)} \\ \vdots \\ \delta_i^{(N_\mathbf{z})} \end{bmatrix} = \begin{bmatrix} e_i^{(1)} \\ e_i^{(2)} \\ \vdots \\ e_i^{(n)} \\ \vdots \\ e_i^{(N_\mathbf{z})} \end{bmatrix} \tag{26}$$

or

$$\mathbf{z}_i = \boldsymbol{\eta}_i(\mathcal{M}, \mathcal{D}, \mathcal{T}, \boldsymbol{\chi}, t_i) + \boldsymbol{\delta}_i + \mathbf{e}_i \tag{27}$$

where $\boldsymbol{\delta}_i$ is the model inadequacy (or deficiency) vector and $\mathbf{e}_i$ observation (e.g. measurement) error vector.

The digital twin output functions defined in Eq. (22) can be comprised of more than one computational model (and here it is assumed there is a combination of physics- and/or data-based model components with output functions defined in Eq. (23)). To combine these multiple components into one digital twin output function, a series of $\rho_{s,p} \in \boldsymbol{\chi}$ weighting functions are introduced for each of the N_p model output functions and the s^{th} data segment d_s. Eq. (25) can therefore be modified to show the combination of P physics- and D data-based models, for the $\rho_{s,p}$ weightings (one for each of the $P + D = N_p$ models, and with $p = 1, 2, 3..., N_p$) such that the combined model outputs then produce one digital twin output function for the n^{th} QoI, as given in Eq. (22). This weighted combination of model output functions has the form (omitting the n superscripts) of

$$\begin{aligned} z_i &= \rho_{s,1}\hat{\eta}_{i,1} + \rho_{s,2}\hat{\eta}_{i,2} + ... + \rho_{s,p}\hat{\eta}_{i,p} + ... + \rho_{s,N_p}\hat{\eta}_{i,N_p} + \delta_i + e_i \\ &= \eta_i(\mathcal{M}, \mathcal{D}, \mathcal{T}, \boldsymbol{\chi}, t_i) + \delta_i + e_i \end{aligned} \tag{28}$$

where $\hat{\eta}_{i,p}$, are the model output functions given in (23). The additive relationship $\eta_i = \rho_{s,1}\hat{\eta}_{i,1} + .. + \rho_{s,N_p}\hat{\eta}_{i,N_p}$ relates the model output functions to the digital twin output function, and is assumed to hold for each of the $N_{\mathbf{z}}$ QoIs.

Notice that the formulation given in (28) allows several possibilities depending on how the weighting functions are chosen. For example, models from the model library can be selected (or deselected) using the weights. If just a single model is selected, then the model calibration methods described by Kennedy and O'Hagan (2001); Ward et al. (2020) could potentially be applied. If multiple weightings are used (and a post-processing setting is available), then a range of ensemble types methods may also become applicable, depending on the precise context being used (Zhou, 2019). The overall scenario is shown schematically in Fig. 23.

5.3 Example: Cascading Tanks System

We now consider an example of a cascading water tanks system, as shown in Fig. 24 — see Edington et al. (2023) for full details. The tank system works as follows. A control input, u_1, controls the pumping of water from the reservoir into Tank 1. Then water from Tank 1 flows into Tank 2 below it, and finally back into the reservoir. Note that it is possible for the tanks to overflow under larger inputs.

There some physics based models that can be applied to the tanks system. For example, we take physics-based model 1 to be the following ordi-

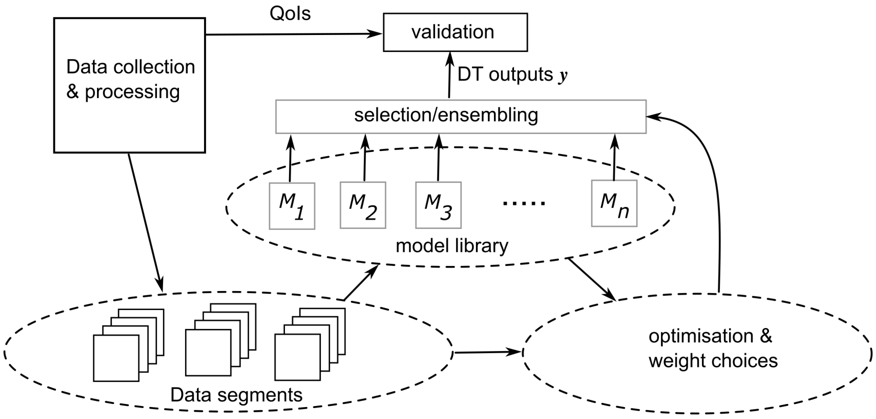


Figure 23. Schematic diagram of the digital twin output function concept, where models can be ensembled or selected to create the output.

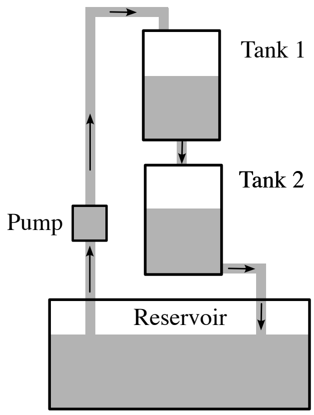


Figure 24. Schematic diagram of the cascading tanks system, as described by Schoukens et al. (2016).

nary differential equation model:

$$M_1 = \begin{cases} \dot{x}_1(t) & = -k_1\sqrt{x_1(t)} + k_4 u(t) + w_1(t) \\ \dot{x}_2(t) & = k_2\sqrt{x_1(t)} - k_3\sqrt{x_2(t)} + w_2(t) \end{cases} \tag{29}$$

Here, the model output is $y_2 = \hat{\eta}_{i,1}^{(2)} = x_2(t)$, $x_1(t)$ and $x_2(t)$ are the water levels of the upper and lower tanks, $u(t) = u_1$ is the system input, $\boldsymbol{\theta}_1 =$

$[k_1\ k_2\ k_3\ k_4]^T$ and $w_1(t)$ and $w_2(t)$ are (optional) noise terms.

A slightly more sophisticated physics-based model (we call it model 2) includes an overflow model so that when $x_2 > 10$, Tank 1 will overflow into the reservoir, and the model becomes:

$$M_2 = \begin{cases} \dot{x}_1(t) & = -k_1\sqrt{x_1(t)} + k_4 u(t) + w_1(t) \\ \dot{x}_2(t) & = \begin{cases} k_2\sqrt{x_1(t)} - k_3\sqrt{x_2(t)} + w_2(t) & x_1(t) \leq 10 \\ k_2\sqrt{x_1(t)} - k_3\sqrt{x_2(t)} + k_5 u(t) + w_3(t) & x_1(t) > 10 \end{cases} \end{cases} \tag{30}$$

As before, the model output is taken to be $y_2 = \hat{\eta}_{i,2}^{(2)} = x_2(t)$. In this model k_5 is an additional parameter and $w_3(t)$ an additional noise term. Therefore $\boldsymbol{\theta}_2 = [k_1\ k_2\ k_3\ k_4\ k_5]^T$ for this model in this case. To account for uncertainty in the physics model parameters, an approximate Bayesian computation (ABC) algorithm with an accept-reject mechanism was employed.

In addition to this, a third model (M_3) based on data was used. This was a nonlinear autoregressive exogenous model (NARX) neural network model which is a nonlinear autoregressive model which has exogenous inputs. In this case the NARX model is given as

$$\begin{aligned} M_3 = \{x_2(t_i) = F(x_2(t_i - 1), x_2(t_i - 2), ... \\ , x_2(t_i - n_x); u_1(t_i), u_1(t_i - 1), ..., u_1(t_i - n_u)) \end{aligned} \tag{31}$$

In this case the model output is $y_2 = \hat{\eta}_{i,3}^{(2)} = x_2(t)$, $x_2(t_i)$ is the level of Tank 2, $u_1(t_i)$ is the input and n_x is the maximum output time lag and n_u the maximum input time lag. A 3-layer NN with 5 hidden nodes was used for each segment's NARX model — see Edington et al. (2023) for full details.

Consider when the physics-based model is chosen to be M_2 and the data-based model is M_3, so the statistical model for $n = 2$ is;

$$z_i^{(2)} = \rho_{s,2}\hat{\eta}_{i,2}^{(2)} + \rho_{s,3}\hat{\eta}_{i,3}^{(2)} + \delta_i + e_i = \eta_i^{(2)} + \delta_i + e_i \tag{32}$$

where δ_i represents the unknown combined model inadequacies, and $\eta_i^{(2)} = \rho_{s,2}\hat{\eta}_{i,2}^{(2)} + \rho_{s,3}\hat{\eta}_{i,3}^{(2)}$ is the combined digital twin output function.

Notice that the subscript, p for ρ_p and $\hat{\eta}_{i,p}$ relates to which model in the model library is being used. For this example there are three models in the library, $M_1, M_2, M_3 \in \mathcal{M}$. So another way to interpret the ensemble is that the weightings are being used to select two out of the three available models, and therefore in this case $\rho_{s,1} = 0$. Note also that $P = 1$, $D = 1$, and so $N_p = 2$ in this example.

The error-based weightings were chosen based on normalised mean-squared error (NMSE). For the s^{th} data segment we chose to minimise the error (combined uncertainties) in the weighted sum

$$\sum_{p=1}^{N_p} \rho_{s,p} \sqrt{C_{s,p}} \tag{33}$$

subject to the constraints that

$$\sum_{p=1}^{N_p} \rho_{s,p} = 1 \quad \text{and} \quad \rho_{s,p} \in [0, 1] \tag{34}$$

A sample result for the cascading tank digital twin system is shown in Fig. 25.

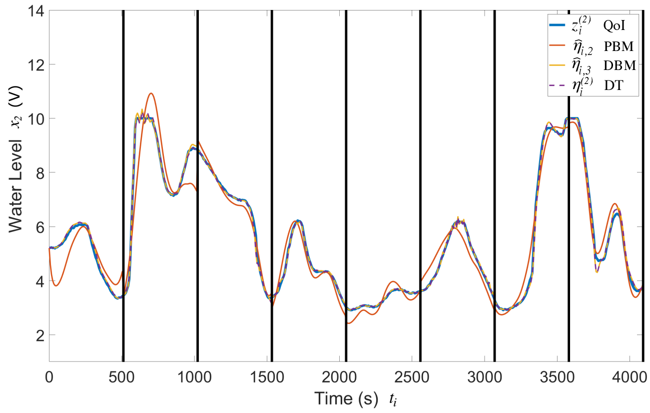


Figure 25. Comparison of the second tank water levels, x_2, computed using the physics-based model (PBM) M_2 (shown by $\hat{\eta}_{i,2}$) & data-based model (DBM) M_3 (shown by $\hat{\eta}_{i,3}$) and digital twin output function for $n = 2$ (denoted $\eta_i^{(2)}$ and DT) compared with the measured water level QoI $z_i^{(2)}$. Note that following Schoukens et al. (2016) the units of x_2 are in volts (V) from the measurement technique used in gathering the data. See Edington et al. (2023) for full details.

5.4 Digital Twin Operational Platforms

"Ready access to versatile and powerful software enables the engineer to do more and think less,"

— Mete A. Sozen, 2002 Distinguished Lecture at the EERI Annual Meeting.

Digital twins are typically realised via a software and hardware *operational platform* (Bonney et al., 2022). Sozen's observation about software enabling engineers to "do more and think less" is relevant not just from the point of view of over-reliance on software, but (although it couldn't be known back in 2002) the comment is also relevant to new developments in AI (e.g. large language models, see Teubner et al. (2023)) which offer the possibility of a non-human AI doing the thinking for us via so-called "cognitive surrogates" (Leslie, 2021). One of the aspirations for future digital twins is that they could include these types of advanced AI functionalities.

In fact, most digital twin software developments up until now have been driven by proprietary software vendors, (Minerva et al., 2020; Somers et al., 2022) resulting primarily in closed-source products. However, there are some open source initiatives as well, as discussed in Bonney et al. (2022), although these are less mature. Hardware aspects are also being discussed, including IoT integration — see for example Platenius-Mohr et al. (2020). Digital twins are composed from many things, but one of the most important is that they are deployment platforms for AI methods. For example, there has been a significant amount of machine learning and data-based modelling in digital twins, and a great deal of interest in ontologies and knowledge models, which we will discuss in Section 5.6.

5.5 Digital Twin Case Study

As a case study of how digital twins are being realised, we consider the example of an aircraft asset management digital twin shown in Fig. 26. In this example, the physical twin is a decommissioned former RAF Hawk T1A aircraft. The purpose of the digital twin was to demonstrate how a ground vibration test (GVT) might be automated and integrated into an asset-management digital twin for the Hawk. The aspiration for the future is that the data would be gathered using non-contact sensors, such as laser vibrometers, as the aircraft enters of leaves the hangar. However, as the aircraft has no engine, and the non-contact sensors were not available to us, accelerometers were attached to the aircraft surface to gather data during the tests, and full details of the data collection method is given in Haywood-Alexander et al. (2024).

A ground vibration test (GVT) is a well established process for aerospace engineering. It gives engineering information about the dynamic behaviour of the aircraft which can be used to (amongst other things) (i) understand how the aircraft might perform in-flight, and, (ii) assess the health state

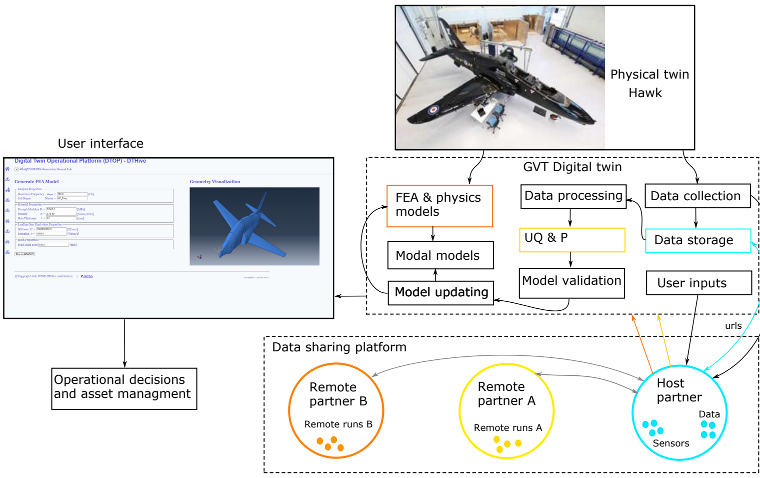


Figure 26. Example of a digital twin for a Hawk T1A aircraft showing a schematic representation of the digital twin operational platform. The physical twin in this example is the (decommissioned) former RAF Hawk T1A aircraft (top box). The objective of the digital twin was to demonstrate how a ground vibration test (GVT) might be automated and integrated into an asset-management digital twin for the Hawk. Discussion of the process for data collection is given in Haywood-Alexander et al. (2024). In a digital twin context, the data collection would ideally be automated, or semi-automated using non-contact sensors. The data collected from the physical twin is processed by the GVT digital twin (middle right box), and then stored in a data-sharing platform (lower right box). The data sharing platform in this case was provided by a commercial partner IOTICS. The data sharing platform enabled geographically remote partners to access and share data. The User Interface (UI) was provided via a web application (middle left), and enabled users to interact with the digital twin (bottom left). UQ & P refers to Uncertainty quantification and propagation analysis.

of the aircraft over a period of time. The GVT is a natural candidate for digital twinning, as it already involves the combination of physics-based models and data. In this case, the physics-based models were finite elements and modal models, and the data sets were acceleration measurements.

The process for combining the data- and physics-based models is shown in the centre-right box of Fig. 26. The data collected from the physical twin is processed by the GVT digital twin, and then used to both validate and update the physics-based models. After processing the data is stored in a data-sharing platform, shown as the lower box in Fig. 26. In this work, the data sharing platform in this case was provided by a commercial partner IOTICS. The data sharing platform enabled two geographically remote partners to access and share data including processing data remotely and uploading results back to the data sharing platform.

The User Interface (UI) was provided via a web application, and enabled users to interact with the digital twin, using an approach described in detail in Bonney et al. (2022), and is shown on the left-hand side of Fig. 26. The operational platform shown in Fig. 26 can be considered as three "layers" — Fig. 27. In this formulation, the foundation layer is the IoT layer, where physical twins and other devices are connected to the network using bespoke hardware. This also includes local DAQ and control hardware, sensors and actuators. The interface layer provides access to users via a web server that coordinates and schedules the required tasks within the workflow. This layer uses the network to connect the user to the data provided by the IoT layer and cloud-based services. The services provided in the cloud computing layer would typically include data storage, high-power-computing (HPC), and any other remote computing facilities required as part of utilising the digital twin.

Future digital-twins are expected to contain a vast amount of information, much of which will be processed through visualisation techniques to be displayed in UIs. In addition, it is anticipated that augmented/virtual reality UIs (or augmented/virtual inspection see e.g. Moreu et al. (2017)) will become possible. Ultimately, the aspiration is that these type of operational platforms for digital twins will provide a reliable connection between the virtual and physical domains.

5.6 Digital Twin Knowledge Models and Ontologies

In order to support functions such as decision support, and scenario planning, digital twins need to store and represent knowledge about the physical twin, and depending on the context, the users, business processes, and multiple other possible related activities. The requirement for a knowledge

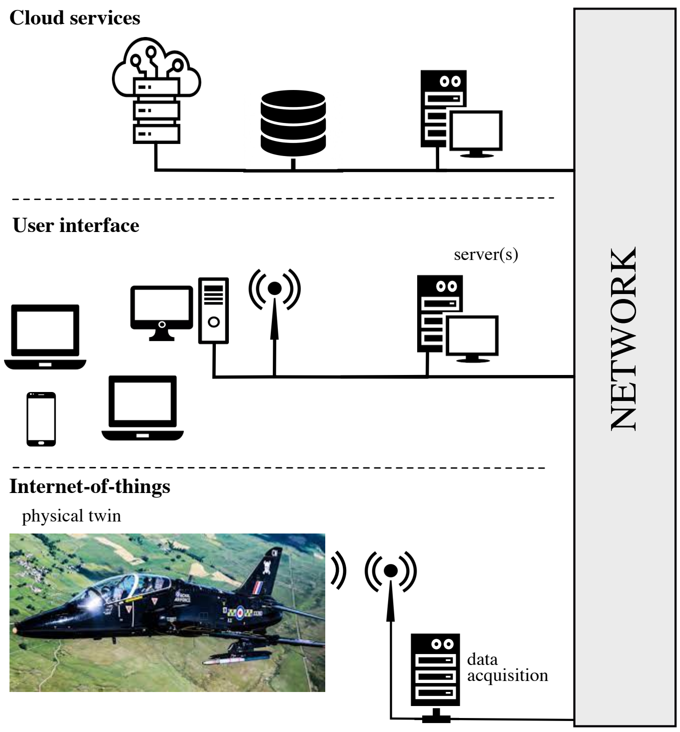


Figure 27. Overall 'layer structure' for a digital twin operational platform. Note the wireless communication with the aircraft is conceptual only. In the testing so-far (see Haywood-Alexander et al. (2024)) the aircraft has not flown and the communications with sensors is via wired connections.

management within a digital twin has been considered by many previous authors, and is sometimes referred to as an information management system or framework (West, 2011; Hetherington and West, 2020).

One particular area that has received considerable attention for developing an information management system is ontologies, and in particular the use of knowledge graphs (Hogan et al., 2021). Knowledge graphs are a graphical way of representing ontologies. The idea of an ontology, and their representation as a knowledge graph, is based in philosophical models for categorisations of knowledge. Knowledge graphs have recently been developed as a tool for computer science applications, such as web searching and customer profiling (Hogan et al., 2021).

In the context of a digital twin, knowledge graphs can be used to represent the semantic relations between data, information and other knowledge within the digital twin. The knowledge graph is made up of a series of entities (nodes) and relationships (edges) that form connections between the entities. Once an initial knowledge graph has been "seeded" (e.g. started) then additional entities and relationships can be added as the digital twin evolves in time. This leads to a *dynamic knowledge graph*, that can be used to create a *digital thread* (Kraft, 2015; West and Blackburn, 2017; Singh and Willcox, 2018). Within a digital twin, knowledge graphs can be used to provide many functions, including:

- Structured representation of domain knowledge and information
- Semantic context for user interaction
- Information retrieval & queries
- Visualisation of complex inter-relationships between the digital objects in the digital twin
- Support for decision-making
- Improved interoperability
- Incorporation of large language models (LLM) and other AI tools

An example schematic layout for a digital twin including an information management system is depicted in Fig. 28. The lower part of Fig. 28, shows (schematically) the relationship between data acquired from the physical twin, and then a data segmented process. The idea of data segmenting was introduced in the example shown in Section 5.3. In the case shown in Fig. 28 data segmentation is combined with the GVT digital twin example introduced in Fig. 26. Note that this is for schematic illustration only, and there will typically be other processes in the digital twin that are not shown.

In Fig. 28, data from the physical twin is recorded using local sensors and data acquisition hardware (sometimes called "edge" hardware). There is also the facility to carry out local (e.g. "edge") data processing and local control.

In the scenario shown in Fig. 28, data is received, segmented and processed continually, data points are labelled $i = 1, 2, 3...$ whereas data segments are $j = 1, 2, 3.....$ The segments build up a continuous catalogue (or library) of data and meta data that is stored in the information management system (IMS). Within each segment, the data is used to validate and (if needed) update the physics-based models.

The information management system acts as a repository for all the data and information associated with the digital twin. A knowledge graph is built based on all the information in the IMS. Once established, the knowledge graph is dynamically updated as each new piece of information is received.

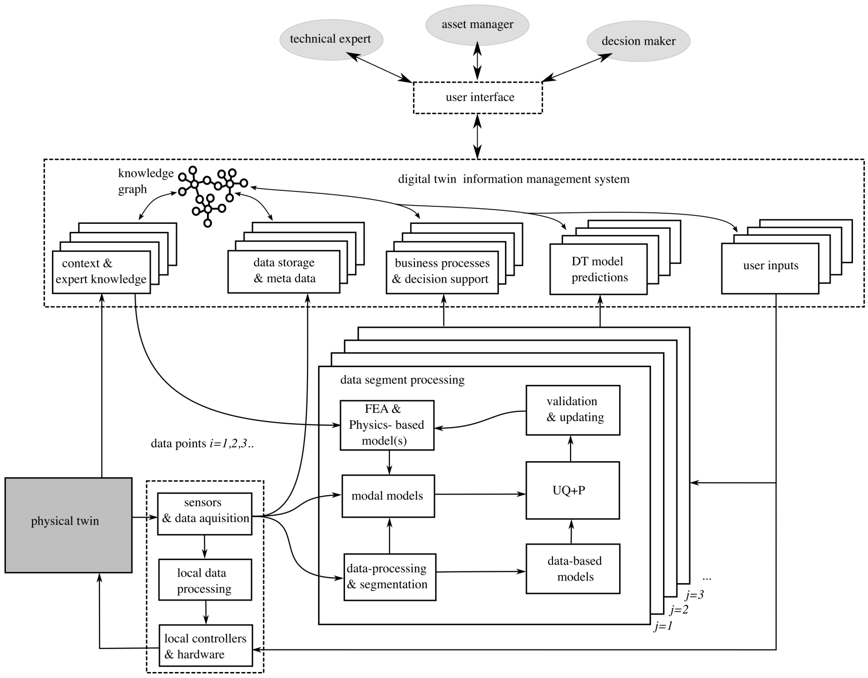


Figure 28. An example schematic layout for a digital twin including an information management system (IMS). Here the user interface (top) provides a means by which the users can interact with the IMS. The IMS acts as a repository for all the data and information associated with the digital twin. The knowledge graph is dynamically updated as each new piece of information is received. In the scenario shown in the figure, data is received, segmented and processed continually. Users are able to interact with the IMS and also give inputs to manage the processing of data, and control of the physical twin. This figure is for schematic illustration only, not all processes are shown.

Users are able to interact with the IMS and also give inputs to manage the processing of data, and control of the physical twin via the user interface as was shown previously, for example in Fig. 26.

6 Conclusions

In this Chapter we have discussed reductionism in science and engineering, particularly in the context of dynamics and control of engineering systems. In particular, we showed an example of control for the nonlinear planar-vertical-take-off-and-landing system. The discussion on control continued with a detailed account of passive control methods. In particular, the idea of adding passive devices (redesign) to reduce resonances was discussed. Two important passive control devices were discussed. The first was the tuned-mass-damper, and the second was the inerter. The inerter is a very new innovation, and we briefly described the development of linear & nonlinear devices for vibration isolation and absorption applications.

Next we carried out a selected review of the philosophy of modelling. As part of the review, the role of knowledge in model making was discussed. We also considered how complexity manifests itself in engineering systems. Related topics of systems engineering, uncertainty analysis and artificial intelligence were also briefly discussed in the context of digital twins. We argued that utility, trust and insight are the three key properties of models that would ideally be extended to digital twins.

Digital twins will rely on the assembly of digital objects. To illustrate this idea, an example of the assembly of models, using a hybrid testing methodology was explained. This is an important technique, that can be used to try and recreate emergent behaviours. Here we also included a discussion on what can be expected in a holistic digital twin. We also discussed the importance of data and showed a Bayesian calibration example to illustrate this point.

The concept of holistic, data-driven applications naturally leads to digital twins. In this Chapter we have focused on dynamics applications, and so we showed examples of how this could work — for example by using digital twin output functions to ensemble multiple model outputs together whilst also modelling uncertainties. In order to implement a digital twin, an operational platform is required. We discussed the software and hardware realisations of digital twins, and showed an example of how they could be used as an operational platform. Lastly we discussed digital twin knowledge models and ontologies, and how this might help shape digital twins in the future.

In terms of future directions for research in this area, there are several topics that could be pursued including:

1. Methods for dynamic assembly: How to couple together multiple, often heterogeneous digital objects
2. Taking actions using control: Can be done using a physics-based approach (nonlinear control) or a data-based ethos (reinforcement learning) or perhaps some combination of both?
3. The role of knowledge in developing digital twins: Ontologies, knowledge graphs and large language models for enhanced functionality of the digital twin
4. Holistic systems methods with more integration of AI methods, such as the use of intelligent agents
5. Emergent behaviour: This is key to more complex behaviours like self-organisation, synchronisation etc.
6. Uncertainty and trust in digital twins: Managing expectations of what could be possible from interactions, and also validating digital twin outputs to build user confidence
7. Philosophical principles for digital twins, that enable new mathematical methods to be developed
8. Digital twin output functions: Ensemble modelling can be used to combine multiple components — can this be extended to interaction?
9. A neuro-symbolic functionality that balances learning and reasoning within the digital twin to support decision making and greater automation

Bibliography

ISO 19650-5:2020 Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) — Information management using building information modelling — Part 5: Security-minded approach to information management, 2020. URL `https://www.iso.org/standard/74206.html`.

ISO 23247-1:2021 Automation systems and integration — Digital twin framework for manufacturing — Part 1: Overview and general principles, 2021. URL `https://www.iso.org/standard/75066.html`.

K. J. Åström and B. Wittenmark. *Adaptive Control*. Addison Wesley: Menlo Park, CA, 1995.

S. Abar, G. K. Theodoropoulos, P. Lemarinier, and G. M. O'Hare. Agent based modelling and simulation tools: A review of the state-of-art software. *Computer Science Review*, 24:13–33, 2017.

K. M. Adams and T. J. Meyers. Perspective 1 of the SoSE methodology: framing the system under study. *International Journal of System of Systems Engineering*, 2(2-3):163–192, 2011.

J. Akroyd, S. Mosbach, A. Bhave, and M. Kraft. Universal digital twin—a dynamic knowledge graph. *Data-Centric Engineering*, 2, 2021.

P. D. Arendt, D. W. Apley, and W. Chen. Quantification of model uncertainty: Calibration, model discrepancy, and identifiability. *Journal of Mechanical Design*, 134(100908):1–12, 2012.

V. I. Arnold. *Geometrical Methods in the Theory of Ordinary Differential Equations*. Springer, 1988.

W. R. Ashby. *An introduction to cybernetics*. Chapman & Hall, 1956.

T. Aven. On the meaning of a black swan in a risk context. *Safety science*, 57:44–51, 2013.

A. Baker. Complexity, networks, and non-uniqueness. *Foundations of Science*, 18:687–705, 2013.

C. F. Beards. *Vibration analysis and control system dynamics*. Ellis Horwood: Chichester, 1981.

M. A. Bedau and P. Humphreys. *Emergence: Contemporary readings in philosophy and science*. MIT press, 2008.

G. Berman and F. Izrailev. The Fermi-Pasta-Ulam problem: fifty years of progress. *Chaos*, 15(1):15104, 2005.

C. M. Bishop. *Pattern Recognition and Machine Learning by Christopher M. Bishop*. Springer Science+ Business Media, LLC, 2006.

R. D. Blevins. *Formulas for natural frequency and mode shape*. Van Nostrand Reinhold: New York, 1979.

D. I. Blockley. *The nature of structural design and safety*. Ellis Horwood, Chichester, 1980.

C. Boje, A. Guerriero, S. Kubicki, and Y. Rezgui. Towards a semantic construction digital twin: Directions for future research. *Automation in construction*, 114:103179, 2020.

M. S. Bonney, M. de Angelis, M. D. Borgo, L. Andrade, S. Beregi, N. Jamia, and D. J. Wagg. Development of a digital twin operational platform using python flask. *Data Centric Engineering*, 3(E1):1–14, 2022. doi: 10.1017/dce.2022.1.

M. S. Bonney, M. de Angelis, M. Dal Borgo, and D. J. Wagg. Contextualisation of information in digital twin processes. *Mechanical Systems and Signal Processing*, 184:109657, 2023.

D. M. Botín-Sanabria, A.-S. Mihaita, R. E. Peimbert-García, M. A. Ramírez-Moreno, R. A. Ramírez-Mendoza, and J. d. J. Lozoya-Santos. Digital twin technology challenges and applications: A comprehensive review. *Remote Sensing*, 14(6):1335, 2022.

G. E. Box. Choice of response surface design and alphabetic optimality. In *Proceedings of the... Conference on the Design of Experiments in Army Research, Development and Testing*, volume 28, page 237, 1982.

J. E. Brock. A note on the damped vibration absorber. *Trans. ASME, J. Appl. Mech.*, 13(4):A–284, 1946.

J. Brynjarsdóttir and A. O'Hagan. Learning about physical parameters: The importance of model discrepancy. *Inverse Problems*, 30(11):114007, 2014.

L. Bucciarelli. *Engineering philosophy.* DUP Satellite; an imprint of Delft University Press, 2003.

J. E. Carrion. *Model-based strategies for real-time hybrid testing.* University of Illinois at Urbana-Champaign Press, 2007.

J. E. Carrion, B. Spencer, and B. M. Phillips. Real-time hybrid simulation for structural control performance assessment. *Earthquake Engineering and Engineering Vibration*, 8:481–492, 2009.

F. Casciati, G. Magonette, and F. Marazzi. *Semiactive devices and applications in vibration mitigation.* Wiley: Chichester, 2006.

P. Checkland. *Systems thinking, systems practice.* Chichester: John Wiley, 1999.

C. Chen and J. M. Ricles. Improving the inverse compensation method for real-time hybrid simulation through a dual compensation scheme. *Earthquake Engineering & Structural Dynamics*, 38(10):1237–1255, 2009.

M. Chen, S. Mao, and Y. Liu. Big data: A survey. *Mobile networks and applications*, 19(2):171–209, 2014.

T. Chouard. The Go files: AI computer clinches victory against Go champion. *Nature*, News March 12, 2016. doi: 10.1038/nature.2016.19553.

R. Christenson, Y. Z. Lin, A. Emmons, and B. Bass. Large-scale experimental verification of semiactive control through real-time hybrid simulation. *Journal of Structural Engineering*, 134(4):522–534, 2008.

O. Chukhno, N. Chukhno, G. Araniti, C. Campolo, A. Iera, and A. Molinaro. Optimal placement of social digital twins in edge iot networks. *Sensors*, 20(21):6181, 2020.

C. Cimino, E. Negri, and L. Fumagalli. Review of digital twin applications in manufacturing. *Computers in Industry*, 113:103130, 2019.

T. Dauxois. Fermi, pasta, ulam, and a mysterious lady. *Physics today*, 61 (1):55–57, 2008.

P. Deastra, D. J. Wagg, N. D. Sims, and R. S. Mills. Experimental shake table validation of damping behaviour in inerter-based dampers. *Bulletin of Earthquake Engineering*, 21(3):1389–1409, 2023.

J. P. Den Hartog. *Mechanical Vibrations.* McGraw-Hill: New York, 1934.

R. Descartes. *The Philosophical Writings of Descartes (Translated by J. Cottingham).* Cambridge University Press, 1985.

R. A. Desjardins and W. E. Hooper. Antiresonant rotor isolation for vibration reduction. *Journal of the American Helicopter Society*, 25(3):46–55, 1980.

A. Dingli and D. Farrugia. *Neuro-Symbolic AI: Design transparent and trustworthy systems that understand the world as you do.* Packt, 2023.

H. Dogan, N. D. Sims, and D. J. Wagg. Implementation of inerter-based dynamic vibration absorber for chatter suppression. *Journal of Manufacturing Science and Engineering*, 145(8):084502, 2023.

L. Edington, N. Dervilis, A. B. Abdessalem, and D. Wagg. A time-evolving digital twin tool for engineering dynamics applications. *Mechanical Systems and Signal Processing*, 188:109971, 2023.

M. C. Edson, P. B. Henning, and S. Sankaran. *A guide to systems research: Philosophy, processes and practice*, volume 10. Springer, 2016.

A. Einstein. On the method of theoretical physics. *Philosophy of science*, 1 (2):163–169, 1934.

M. R. Enders and N. Hoßbach. Dimensions of digital twin applications - a literature review. In *Proceedings of Twenty-fifth Americas Conference on Information Systems*, 2019.

I. Errandonea, S. Beltrán, and S. Arrizabalaga. Digital twin for maintenance: A literature review. *Computers in Industry*, 123:103316, 2020.

N. Fernández, C. Maldonado, and C. Gershenson. Information measures of complexity, emergence, self-organization, homeostasis, and autopoiesis. In *Guided self-organization: Inception*, pages 19–51. Springer, 2014.

W. G. Flannelly. Dynamic antiresonant vibration isolator, May 30 1967. US Patent 3,322,379.

W. C. Flower. Understanding hydraulic mounts for improved vehicle noise, vibration and ride qualities. *SAE Technical Paper*, 1:850975, 1985.

A. L. Fradkov, I. M. Miroshnik, and V. O. Nikiforov. *Nonlinear and adaptive control of complex systems.* Kluwer: Dordrecht, 1999.

H. Frahm. Device for damping vibrations of bodies, 1909.

R. Frei and G. D. M. Serugendo. The future of complexity engineering. *Central European Journal of Engineering*, 2(2):164–188, 2012.

X. Gao, N. Castaneda, and S. J. Dyke. Real time hybrid simulation: from dynamic system, motion control to experimental error. *Earthquake engineering & structural dynamics*, 42(6):815–832, 2013.

A. Garcez and L. C. Lamb. Neurosymbolic AI: The 3rd wave. *Artificial Intelligence Review*, 56(11):12387–12406, 2023.

P. Gardner, M. Dal Borgo, V. Ruffini, A. J. Hughes, Y. Zhu, and D. J. Wagg. Towards the development of an operational digital twin. *Vibration*, 3(3): 235–265, 2020.

P. J. Gawthrop, S. A. Neild, A. Gonzalez-Buelga, and D. J. Wagg. Causality in real-time dynamic substructure testing. *Mechatronics*, 19(7):1105–1115, 2009.

A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Dunson. *Bayesian data analysis*, volume 3. CRC press, 2014.

C. Gershenson. The implications of interactions for science and philosophy. *Foundations of Science*, 18:781–790, 2013.

M. Girolami. *A first course in machine learning*. Chapman and Hall/CRC, 2011.

E. H. Glaessgen and D. Stargel. The digital twin paradigm for future NASA and US Air Force vehicles. In *53rd Struct. Dyn. Mater. Conf. Special Session: Digital Twin, Honolulu, HI, US*, pages 1–14, 2012.

P. Glendinning. *Stability, instability and chaos*. Cambridge University Press, 1994.

A. Gonzalez-Buelga, D. J. Wagg, M. I. Wallace, S. A. Neild, and J. H. G. Macdonald. Testing an autoparametric pendulum using a hybrid numerical experimental method. In *Proceedings of the 1st International Conference on Advances in Experimental Structural Engineering*, volume 1, pages 417–424, 2005.

A. Gonzalez-Buelga, D. J. Wagg, and S. A. Neild. Parametric variation of a coupled pendulum-oscillator system using real-time dynamic substructuring. *Structural Control and Health Monitoring*, 14(7):991–1012, 2007.

G. C. Goodwin, S. F. Graebe, and M. E. Salgado. *Control System Design*. Pearson, 2000.

L. Graesser and W. L. Keng. *Foundations of deep reinforcement learning*. Addison-Wesley Professional, 2019.

J. Gray. *Henri Poincaré: A Scientific Biography*. Princeton University Press, 2012.

M. Grieves. *Product Lifecycle Management: Driving the next generation of lean thinking*. McGraw-Hill Professional, 2005.

M. Grieves and J. Vickers. Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. In *Transdisciplinary Perspectives on Complex Systems*, pages 85–113. Springer, 2017.

J. Guckenheimer and P. Holmes. *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*. Springer-Verlag: New York, 1983.

M. Gutierrez Soto and H. Adeli. Tuned mass dampers. *Arch. Comput. Methods Eng.*, 20:419–431, 2013.

M. Haenlein and A. Kaplan. A brief history of artificial intelligence: On the past, present, and future of artificial intelligence. *California management review*, 61(4):5–14, 2019.

N. Hall. *The new scientist guide to chaos*. (Penguin), 1992.

S. Hawking and L. Mlodinow. *The grand design*. Random House Digital, Inc., 2010.

M. Haywood-Alexander, R. S. Mills, M. D. Champneys, M. R. Jones, M. S. Bonney, D. Wagg, and T. J. Rogers. Full-scale modal testing of a Hawk T1A aircraft for benchmarking vibration-based methods. *Journal of Sound and Vibration*, page 118295, 2024.

B. He and K.-J. Bai. Digital twin-based sustainable intelligent manufacturing: A review. *Advances in Manufacturing*, 9(1):1–21, 2021.

D. Heber and M. Groll. Towards a digital twin: How the blockchain can foster e/e-traceability in consideration of model-based systems engineering. In *DS 87-3 Proceedings of the 21st International Conference on Engineering Design (ICED 17) Vol 3: Product, Services and Systems Design, Vancouver, Canada, 21-25.08. 2017*, 2017.

J. Hetherington and M. West. The pathway towards an information management framework-a 'commons' for digital built britain. cdbb white paper. 2020.

F. Heylighen, P. Cilliers, and C. G. Garcia. Complexity and philosophy. In *Complexity, Science and Society*, pages 117–134. Radcliffe Publishing, 2007.

F. S. Hillier and G. J. Lieberman. *Introduction to operations research*. McGraw hill Companies, Inc., 2001.

M. W. Hirsch and S. Smale. *Differential equations, dynamical systems , and linear algebra*. Academic Press, 1974.

S. R. Hirshorn, L. D. Voss, and L. K. Bromley. *Nasa systems engineering handbook*. NASA, 2017.

H. Hoagland. Harlow shapley—some recollections. *Publications of the Astronomical Society of the Pacific*, 77(459):422–430, 1965.

A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, et al. Knowledge graphs. *ACM Computing Surveys (Csur)*, 54(4):1–37, 2021.

O. T. Holland. Taxonomy for the modeling and simulation of emergent behavior systems. In *Proceedings of the 2007 spring simulation multiconference-Volume 2*, pages 28–35, 2007.

T. Holland. *Foundations for the modeling and simulation of emergent behavior systems*. CRS Press/Taylor and Francis, Boca Raton, FL, USA, 2018.

W. Hoppitt and K. N. Laland. *Social learning: an introduction to mechanisms, methods, and models*. Princeton University Press, 2013.

Z. Huang, Y. Shen, J. Li, M. Fey, and C. Brecher. A survey on AI-driven digital twins in Industry 4.0: Smart manufacturing and advanced robotics. *Sensors*, 21(19):6340, 2021.

R. A. Ibrahim. Recent advances in nonlinear passive vibration isolators. *Journal of Sound and Vibration*, 314(3–5):371–452, 2008.

K. Ikago, K. Saito, and N. Inoue. Seismic control of single-degree-of-freedom structure using tuned viscous mass damper. *Earthquake Engineering & Structural Dynamics*, 41(3):453–474, 2012.

D. J. Inman. *Vibration with control*. Wiley: Chichester, 2006.

A. Isidori. *Nonlinear Control Systems*. Springer, 1995.

M. Jafari, A. Kavousi-Fard, T. Chen, and M. Karimi. A review on digital twin technology in smart grid, transportation system and smart city: Challenges and future. *IEEE Access*, 2023.

M. Jans-Singh, K. Leeming, R. Choudhary, and M. Girolami. Digital twin of an urban-integrated hydroponic farm. *Data-Centric Engineering*, 1, 2020.

H. J. Jensen. *Complexity science: the study of emergence*. Cambridge University Press, 2022.

F. Jiang, L. Ma, T. Broyd, and K. Chen. Digital twin and its implementations in the civil engineering sector. *Automation in Construction*, 130: 103838, 2021.

L. Jinzhi, Y. Zhaorui, Z. Xiaochen, W. Jian, and K. Dimitris. Exploring the concept of cognitive digital twin from model-based systems engineering perspective. *The International Journal of Advanced Manufacturing Technology*, 121(9-10):5835–5854, 2022.

D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks. Characterising the digital twin: A systematic literature review. *CIRP Journal of Manufacturing Science and Technology*, 2020.

R. Jones. An analytical and model test research study on the kaman dynamic antiresonant vibration isolator (DAVI). Technical report, Kaman Aerospace Corp Bloomfield CT, 1968.

M. G. Kapteyn, D. J. Knezevic, and K. Willcox. Toward predictive digital twins via component-based reduced-order models and interpretable machine learning. In *AIAA Scitech 2020 Forum*, page 0418, 2020.

M. G. Kapteyn, J. V. R. Pretorius, and K. E. Willcox. A probabilistic graphical model foundation for enabling predictive digital twins at scale. *Nature*, 1:337–347, 2021.

S. A. Kauffman. *Investigations*. Oxford University Press, 2000.

S. Kawamata. Development of a vibration control system of structures by means of mass pumps. Technical report, Institute of Industrial Science. Tokyo, Japan: University of Tokyo, 1973.

M. C. Kennedy and A. O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001.

H. K. Khalil. *Nonlinear Systems*. Macmillan: New York, 1992.

B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. Al Sallab, S. Yogamani, and P. Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6): 4909–4926, 2021.

P. Kitcher. Science in a democratic society. In *Scientific realism and democratic society*, pages 95–112. Brill, 2011.

I. Kourakis. *Structural systems and tuned mass dampers of super-tall buildings: case study of Taipei 101.* PhD thesis, Massachusetts Institute of Technology, 2007.

E. M. Kraft. HPCMP CREATE TM-AV and the Air Force digital thread. *AIAA SciTech*, 2015.

C. Krishnamoorthy and S. Rajeev. *Artificial intelligence and expert systems for engineers.* CRC press, 2018.

W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn. Digital twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine*, 51(11):1016–1022, 2018.

M. Krstić, I. Kanellakopoulos, and P. Kokotović. *Nonlinear and adaptive control design.* John Wiley, 1995.

J. Kruschke. *Doing Bayesian data analysis: A tutorial with R, JAGS, and Stan.* Academic Press, 2014.

W. M. Kuhnert, P. J. P. Gonçalves, D. F. Ledezma-Ramirez, and M. J. Brennan. Inerter-like devices used for vibration isolation: A historical perspective. *Journal of the Franklin Institute*, 2020.

Y. N. Kyrychko, K. B. Blyuss, A. Gonzalez-Buelga, S. J. Hogan, and D. J. Wagg. Real-time dynamic substructuring in a coupled oscillator-pendulum system. *Proceedings of the Royal Society A*, 462(2068):1271–1294, 2006.

R. B. Lanza. Does your project risk management system do the job? *Information Strategy: The executive's journal*, 17(1):6–12, 2000.

P.-S. Laplace. *A philosophical essay on probabilities (Translated by F Truscott and F Emory).* Courier Corporation, 2012.

A. Lasota and M. C. Mackey. *Chaos, Fractals and Noise; Stochastic Aspects of Dynamics.* Springer-Verlag: New York, 1994.

I. F. Lazar, S. A. Neild, and D. J. Wagg. Using an inerter-based device for structural vibration suppression. *Earthquake Engineering & Structural Dynamics*, 43(8):1129–1147, 2014. DOI: 10.1002/eqe.2390.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553): 436–444, 2015.

D. Leslie. The Arc of the Data Scientific Universe. *Harvard Data Science Review*, 3(1), jan 29 2021. https://hdsr.mitpress.mit.edu/pub/ln5hu6vq.

M. Li, P. Vitányi, et al. *An introduction to Kolmogorov complexity and its applications*, volume 3. Springer, 2008.

C. Liu, L. Chen, H. P. Lee, Y. Yang, and X. Zhang. A review of the inerter and inerter-based vibration isolation: theory, devices, and applications. *Journal of the Franklin Institute*, 2022.

K. Liu and J. Liu. The damped dynamic vibration absorbers: revisited and new result. *Journal of sound and vibration*, 284(3-5):1181–1189, 2005.

M. Liu, S. Fang, H. Dong, and C. Xu. Review of digital twin about concepts, technologies, and industrial applications. *Journal of Manufacturing Systems*, 2020. ISSN 0278-6125.

S. Liu, J. Bao, and P. Zheng. A review of digital twin-driven machining: From digitization to intellectualization. *Journal of Manufacturing Systems*, 67:361–378, 2023.

C. Lo, C. Chen, and R. Y. Zhong. A review of digital twin in product design and development. *Advanced Engineering Informatics*, 48:101297, 2021.

R. Ma, K. Bi, and H. Hao. Inerter-based structural vibration control: A state-of-the-art review. *Engineering Structures*, 243:112655, 2021.

A. M. Madni, C. C. Madni, and S. D. Lucero. Leveraging digital twin technology in model-based systems engineering. *Systems*, 7(1):7, 2019.

G. Marcus. Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631*, 2018.

G. Marcus. The next decade in AI: four steps towards robust artificial intelligence. *arXiv preprint arXiv:2002.06177*, 2020.

L. Marian and A. Giaralis. Optimal design of a novel tuned mass-damper-inerter (TMDI) passive vibration control configuration for stochastically support-excited structural systems. *Probabilistic Engineering Mechanics*, 38:156–164., 2014.

M. W. Martin and R. Schinzinger. *Engineering ethics*. McGraw-Hill, 2008.

P. McCorduck. *Machines who think: A personal inquiry into the history and prospects of artificial intelligence*. CRC Press, 2004.

D. J. Mead. *Passive Vibration Control*. John Wiley and Sons, 1999.

T. Y. Melesse, V. Di Pasquale, and S. Riemma. Digital twin models in industrial operations: A systematic literature review. *Procedia Manufacturing*, 42:267–272, 2020.

J. Michael, J. Pfeiffer, B. Rumpe, and A. Wortmann. Integration challenges for digital twin systems-of-systems. In *Proceedings of the 10th IEEE/ACM International Workshop on Software Engineering for Systems-of-Systems and Software Ecosystems*, pages 9–12, 2022.

A. M. Miller, R. Alvarez, and N. Hartman. Towards an extended model-based definition for the digital twin. *Computer-Aided Design and Applications*, 15(6):880–891, 2018.

R. Minerva, G. M. Lee, and N. Crespi. Digital twin in the IoT context: a survey on technical features, scenarios, and architectural models. *Proceedings of the IEEE*, 108(10):1785–1824, 2020.

C. Mitcham. The importance of philosophy to engineering. *Teorema: Revista internacional de filosofía*, pages 27–47, 1998.

M. Mitchell. *Complexity: A guided tour*. Oxford university press, 2009.

F. C. Moon. *Chaotic vibrations: An introduction for applied scientists and engineers*. John Wiley: New York, 1987.

F. Moreu, B. Bleck, S. Vemuganti, D. Rogers, and D. Mascarenas. Augmented reality tools for enhanced structural inspection. In *Structural Health Monitoring 2017*, 2017.

E. Negri, L. Fumagalli, and M. Macchi. A review of the roles of digital twin in CPS-based production systems. *Procedia Manufacturing*, 11:939–948, 2017.

I. Newton. *Philosophiae naturalis principia mathematica.* London: Reg. Soc. Præss, 1686.

T. N. Nguyen. Toward human digital twins for cybersecurity simulations on the metaverse: Ontological and network science approach. *JMIRx Med*, 3(2):e33502, 2022.

S. A. Niederer, M. S. Sacks, M. Girolami, and K. Willcox. Scaling digital twins from the artisanal to the industrial. *Nature Computational Science*, 1(5):313–320, 2021.

H. Nijmeijer and A. van der Schaft. *Nonlinear Dynamical Control Systems.* Springer-Verlag: New York, 1990.

N. J. Nilsson. *The quest for artificial intelligence.* Cambridge University Press, 2009.

W. L. Oberkampf and C. J. Roy. *Verification and validation in scientific computing.* Cambridge University Press, 2010.

L. A. Okashah and P. M. Goldwater. Unknown unknowns: modeling unanticipated events. In *Proceedings of Winter Simulation Conference*, pages 689–694. IEEE, 1994.

A. Okumura. The gyro-mass inerter japan patent koukai. h09-177875, 1997.

T. Olsson and J. Axelsson. Systems-of-systems and digital twins: A survey and analysis of the current knowledge. In *2023 18th Annual System of Systems Engineering Conference (SoSe)*, pages 1–6. IEEE, 2023.

R. Pathria and P. D. Beale. *edition 3. Statistical mechanics.* Amsterdam. Elsevier, 2011.

M. Platenius-Mohr, S. Malakuti, S. Grüner, J. Schmitt, and T. Goldschmidt. File-and API-based interoperability of digital twins by model transformation: An IIoT case study using asset administration shell. *Future Generation Computer Systems*, 113:94–105, 2020.

A. Preumont and K. Seto. *Active Control of Structures.* WileyBlackwell, 2008.

W. Purcell and T. Neubauer. Digital twins in agriculture: A state-of-the-art review. *Smart Agricultural Technology*, page 100094, 2022.

J. Ríos, J. C. Hernández, M. Oliva, and F. Mas. Product avatar as digital counterpart of a physical individual product: Literature review and implications in an aircraft. In *ISPE CE*, pages 657–666, 2015.

R. Rosen, G. von Wichert, G. Lo, and K. D. Bettenhausen. About the importance of autonomy and digital twins for the future of manufacturing. *IFAC-PapersOnLine*, 48(3):567–572, 2015.

C. Rovelli. *Seven brief lessons on physics*. Riverhead Books, 2016.

S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, 2010.

R. Sacks, I. Brilakis, E. Pikas, H. S. Xie, and M. Girolami. Construction with digital twin information systems. *Data-Centric Engineering*, 1, 2020.

M. Saitoh. On the performance of gyro-mass devices for displacement mitigation in base isolation systems. *Structural Control and Health Monitoring*, 19(2):246–259, 2012.

S. Sastry. *Nonlinear systems:Analysis, stability and control*. Springer-Verlag: New York, 1999.

T. Savage, J. Akroyd, S. Mosbach, M. Hillman, F. Sielker, and M. Kraft. Universal digital twin–the impact of heat pumps on social inequality. *Advances in Applied Energy*, 5:100079, 2022.

K. J. Schlager. Systems engineering-key to modern development. *IRE transactions on engineering management*, 1(3):64–66, 1956.

M. Schluse, M. Priggemeyer, L. Atorf, and J. Rossmann. Experimentable digital twins—streamlining simulation-based systems engineering for industry 4.0. *IEEE Transactions on Industrial Informatics*, 14(4):1722–1731, 2018.

M. Schoukens, P. Mattson, T. Wigren, and J.-P. Noel. Cascaded tanks benchmark combining soft and hard nonlinearities. In *Workshop on nonlinear system identification benchmarks*, pages 20–23, 2016.

C. Semeraro, M. Lezoche, H. Panetto, and M. Dassisti. Digital twin paradigm: A systematic literature review. *Computers in Industry*, 130: 103469, 2021.

S. M. Sepasgozar, A. A. Khan, K. Smith, J. G. Romero, X. Shen, S. Shirowzhan, H. Li, and F. Tahmasebinia. Bim and digital twin for developing convergence technologies as future of digital construction. *Buildings*, 13(2):441, 2023.

E. Shahat, C. T. Hyun, and C. Yeom. City digital twin potentials: A review and research agenda. *Sustainability*, 13(6):3386, 2021.

A. D. Shaw, S. A. Neild, and D. J. Wagg. Dynamic analysis of high static low dynamic stiffness vibration isolation mounts. *Journal of Sound and Vibration*, 332:1437–1455, 2013a.

A. D. Shaw, S. A. Neild, D. J. Wagg, P. M. Weaver, and A. Carrella. A nonlinear spring mechanism incorporating a bistable composite plate for vibration isolation. *Journal of Sound and Vibration*, 332(24):6265–6275, 2013b.

D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

M. Singh, R. Srivastava, E. Fuenmayor, V. Kuts, Y. Qiao, N. Murray, and D. Devine. Applications of digital twin across industries: a review. *Applied Sciences*, 12(11):5727, 2022.

S. Singh, E. Shehab, N. Higgins, K. Fowler, D. Reynolds, J. A. Erkoyuncu, and P. Gadd. Data management for developing digital twin ontology model. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, page 0954405420978117, 2020.

V. Singh and K. E. Willcox. Engineering design with digital thread. *AIAA Journal*, 56(11):4515–4528, 2018.

M. Sivaselvan, A. Reinhorn, Z. Liang, and X. Shao. Real-time dynamic hybrid testing of structural systems. *Proc. 13th World Conf. Earthquake Engineering*, 2004.

J.-J. E. Slotine and W. Li. *Applied nonlinear control*. Prentice Hall: Englewood Cliffs, NJ, 1991.

M. C. Smith. Synthesis of mechanical networks: the inerter. *IEEE Transactions on Automatic Control*, 47(10):1648–1662, 2002.

M. C. Smith. The inerter: A retrospective. *Annual Review of Control, Robotics, and Autonomous Systems*, 3:361–391, 2020.

R. C. Smith. *Uncertainty quantification: theory, implementation, and applications*, volume 12. Siam, 2013.

R. J. Somers, J. A. Douthwaite, D. J. Wagg, N. Walkinshaw, and R. M. Hierons. Digital-twin-based testing for cyber–physical systems: A systematic literature review. *Information and Software Technology*, page 107145, 2022.

S. H. Strogatz. *Nonlinear Dynamics and Chaos*. Perseus Books Group, 2019.

B. Tadić. Self-organised criticality and emergent hyperbolic networks: blueprint for complexity in social dynamics. *European Journal of Physics*, 40(2):024002, 2019.

N. Taleb. *The Black Swan:The Impact of the Highly Improbable*. Penguin, London, 2007.

F. Tao, B. Xiao, Q. Qi, J. Cheng, and P. Ji. Digital twin modeling. *Journal of Manufacturing Systems*, 64:372–389, 2022.

T. Teubner, C. M. Flath, C. Weinhardt, W. van der Aalst, and O. Hinz. Welcome to the era of chatgpt et al. the prospects of large language models. *Business & Information Systems Engineering*, 65(2):95–101, 2023.

A. Thelen, X. Zhang, O. Fink, Y. Lu, S. Ghosh, B. D. Youn, M. D. Todd, S. Mahadevan, C. Hu, and Z. Hu. A comprehensive review of digital twin—part 2: roles of uncertainty quantification and optimization, a battery digital twin, and perspectives. *Structural and multidisciplinary optimization*, 66(1):1, 2023.

J. M. T. Thompson and H. B. Stewart. *Nonlinear dynamics and chaos.* John Wiley: Chichester, 2002.

J. J. Thomsen. *Vibrations and stability: Advanced theory, analysis and tools.* Springer, 2003.

B. Titurus. Generalized liquid-based damping device for passive vibration control. *AIAA Journal*, 56(10):4134–4145, 2018.

G. Tononi, M. Boly, M. Massimini, and C. Koch. Integrated information theory: from consciousness to its physical substrate. *Nature Reviews Neuroscience*, 17(7):450–461, 2016.

N. Tsokanas, D. Wagg, and B. Stojadinovic. Robust model predictive control for dynamics compensation in real-time hybrid simulation. *Frontiers in Built Environment*, 6, 2020.

N. Tsokanas, D. Wagg, and B. Stojadinovic. Robust model predictive control for dynamics compensation in real-time hybrid simulation. *Recent Advances and Applications of Hybrid Simulation*, pages 57–73, 2021.

E. J. Tuegel, A. R. Ingraffea, T. G. Eason, and S. M. Spottswood. Reengineering aircraft structural life prediction using a digital twin. *International Journal of Aerospace Engineering*, 2011, 2011.

A. Turing. Computing machinery and intelligence. *Mind*, 59(236):433–60, 1950.

I. Van de Poel and D. E. Goldberg. *Philosophy and engineering: An emerging agenda*, volume 2. Springer Science & Business Media, 2010.

W. G. Vincenti et al. *What engineers know and how they know it*, volume 141. Baltimore: Johns Hopkins University Press, 1990.

G. Vining. George box: Scholar, scientist, and statistician, 2013.

D. Wagg, C. Burr, J. Shepherd, Z. Xuereb Conti, M. Enzer, and S. Niederer. The philosophical foundations of digital twinning. Jan. 2024. doi: 10.31224/3500.

D. J. Wagg. A review of the mechanical inerter: historical context, physical realisations and nonlinear applications. *Nonlinear Dynamics*, 104:13–34, 2021.

D. J. Wagg and S. A. Neild. *Nonlinear vibration with control — For flexible and adaptive structures.* Springer, 2nd edition, 2015.

D. J. Wagg, P. Gardner, R. J. Barthorpe, and K. Worden. On key technologies for realising digital twins for structural dynamics applications. In *Model Validation and Uncertainty Quantification, Volume 3*, pages 267–272. Springer, 2020a.

D. J. Wagg, K. Worden, R. J. Barthorpe, and P. Gardner. Digital Twins: State-of-the-Art and Future Directions for Modeling and Simulation in Engineering Dynamics Applications. *ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg*, 6(3), 05 2020b. ISSN 2332-9017. doi: 10.1115/1.4046739. 030901.

D. D. Walden, G. J. Roedler, K. Forsberg, R. D. Hamelin, and T. M. Shortell. *INCOSE systems engineering handbook*. John Wiley, 2015.

M. M. Waldrop. *Complexity: The emerging science at the edge of order and chaos*. Simon and Schuster, 1993.

M. Wallace, D. Wagg, and S. A. Neild. An adaptive polynomial based forward prediction algorithm for multi-actuator real-time dynamic substructuring. *Proceedings of the Royal Society of London A.*, 461(2064): 3807–3826, 2005a.

M. I. Wallace, A. Gonzalez-Buelga, S. A. Neild, and D. J. Wagg. Control techniques for real-time dynamic substructuring. *IADAT Journal of Advanced Technology on Automation, Control and Instrumentation*, 1(1): 1–4, 2005b.

T. R. Wanasinghe, L. Wroblewski, B. Petersen, R. G. Gosine, L. A. James, O. De Silva, G. K. Mann, and P. J. Warrian. Digital twin for the oil and gas industry: Overview, research trends, opportunities, and challenges. *IEEE Access*, 2020.

R. Ward, R. Choudhary, A. Gregory, and M. Girolami. Continuous calibration of a digital twin: comparison of particle filter and Bayesian calibration approaches. *arXiv preprint arXiv:2011.09810*, 2020.

M. D. Watson. Systems engineering principles and hypotheses. *Insight*, 22 (1):18–28, 2019.

M. D. Watson, B. Mesmer, and P. Farrington. Engineering elegant systems: Postulates, principles, and hypotheses of systems engineering. In *Systems Engineering in Context: Proceedings of the 16th Annual Conference on Systems Engineering Research*, pages 495–513. Springer, 2019.

T. Weissert. *The genesis of simulation in dynamics: pursuing the Fermi-Pasta-Ulam problem*. Springer-Verlag New York, Inc., 1999.

M. West. *Developing high quality data models*. Elsevier, 2011.

T. D. West and M. Blackburn. Is digital thread/digital twin affordable? a systemic assessment of the cost of dod's latest manhattan project. *Procedia computer science*, 114:47–56, 2017.

K. Worden, E. J. Cross, R. J. Barthorpe, D. J. Wagg, and P. Gardner. On digital twins, mirrors, and virtualizations: Frameworks for model verification and validation. *ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg*, 6(3), 2020.

L. Wright and S. Davidson. How to tell the difference between a model and a digital twin. *Advanced Modeling and Simulation in Engineering Sciences*, 7(1):1–13, 2020.

L. Yang, K. Cormican, and M. Yu. Ontology-based systems engineering: A state-of-the-art review. *Computers in Industry*, 111:148–171, 2019.

Z.-H. Zhou. *Ensemble methods: foundations and algorithms*. Chapman and Hall/CRC, 2019.