Proceedings Paper:
Somers, R.J., Clark, A.G., Walkinshaw, N. et al. (1 more author) (2022) Reliable counterparts : efficiently testing causal relationships in digital twins. In: MODELS '22: Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings. MODELS '22: ACM/IEEE 25th International Conference on Model Driven Engineering Languages and Systems, 23-28 Oct 2022, Montreal, Quebec, Canada. Association for Computing Machinery , pp. 468-472. ISBN 9781450394673

https://doi.org/10.1145/3550356.3561589

Reliable Counterparts: Efficiently Testing Causal Relationships in Digital Twins

Richard J. Somers
University of Sheffield
Sheffield, United Kingdom
rsomers1@sheffield.ac.uk

Andrew G. Clark
University of Sheffield
Sheffield, United Kingdom

Neil Walkinshaw
University of Sheffield
Sheffield, United Kingdom

Robert M. Hierons
University of Sheffield
Sheffield, United Kingdom

ABSTRACT

The lack of testability of digital twins poses several difficulties when developing reliable systems. Intricate models complicate the definition of comprehensive testing criteria, and physical couplings make obtaining test data an arduous task. To alleviate these challenges, we explore the use of causal inference based testing and propose a technique to allow for correct behaviour of digital twins to be captured in causal diagrams, which are then tested with an efficient data set through the use of counterfactuals. We explore a motivating example of a robotic arm to show how this technique can confirm known causal relationships in a system, and even uncover a fault in the system which caused dangerous behaviour. Our technique localised this erroneous behaviour to a single causal relationship between two variables. Having shown this technique works with a case study, we explore its limitations and the challenges when approaching other industrial applications.

KEYWORDS

digital twin, causal inference, testing, fault localisation, cyber-physical system

ACM Reference Format:
Richard J. Somers, Andrew G. Clark, Neil Walkinshaw, and Robert M. Hierons. 2022. Reliable Counterparts: Efficiently Testing Causal Relationships in Digital Twins. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Digital twins are an emerging technology. They use data-driven and physics-based simulation to model, communicate with and provide additional functionality to their coupled physical system [7]. Digital twins rely on models to accurately represent their physical counterpart, accepting real-time data from the coupled physical system and relaying feedback to provide more informed functionality [5, 8]. These systems have recently been adopted into safety critical and human collaborative environments [6, 10], making it essential that they have correct system behaviour.

Digital twins present a complex challenge due to their lack of testability. Coupling to physical systems as well as intricate models poses a problem to test adequacy across both the physical and virtual domains. These difficulties also provide challenges in executing the required tests due to the extensive physical setup, expensive run times, and complex internal structures [17]. These aspects define digital twins as having a lack of testability [18].

Current testing solutions for digital twins exist [21], but test only the physical system using the digital twin. Testing of the digital twin itself is rarely considered, leading to these inherently complex models being less reliable in safety critical environments [17].

We propose a method that uses causal inference based testing to solve these challenges, building on previous methods [1, 15]. We address the limitations of these methods' requirements to have understandable and accessible source code by capturing domain expertise through causal graphs. As well as this, we use counterfactual test cases, which reduce the amount of test data required, making this technique better for use with physically coupled systems.

2 MOTIVATING EXAMPLE

There has been an increase in the adoption of digital twin based systems in smart manufacturing and safety-critical environments [6, 10]. It is, therefore, important that we have systematic methods for validating such systems. Incorrect or untested digital twin behaviours can lead to expensive faults in modular robotics as well as dangerous behaviour in collaborative human robot environments [6]. Current testing techniques struggle due to the coupling of digital and physical spaces causing cumbersome testing requirements, making obtaining test data for these systems difficult [17].

This section outlines the case study for this paper and highlights the problems that arise when applying traditional testing techniques to digital twins. We first outline the digital twin we will be focusing on during this paper, and then explore its difficulties in testing.

2.1 Digital Twin - Robotic Arm

We analysed a robotic arm model as part of a digital twin framework [5] as a case study. This digital twin framework was provided by the Sheffield Advanced Manufacturing Research Centre and is used to monitor and assure the safety of real human robot collaborative environments. It allows for execution of digital twin models,

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

coupling to physical systems and sharing data between different domains and other digital twins. The robot arm chosen for this study was the UR10 from Universal Robotics [16] as it is widely used in industry and would require a complex model for the behaviour of all 6 joints. This robotic arm allowed for tools, held by the robot's end effector, to be positioned anywhere in 3d space, at any angle, requiring precise movements to achieve this.

The model of the UR10 provided in the digital twin framework is complex, with different aspects of the model running on separate services to accommodate its intricate behaviour. A position in 3d space can be given to the digital twin, which moves the arm, in virtual space, so that the tool of the end effector is positioned precisely where requested. We aim to test the model of this digital twin throughout the remainder of this paper.

2.2 Testing Difficulties

It is essential for safety that we ensure reliably correct behaviour in a human robotic collaborative framework. We outline how the unintelligible models and lack of controlled data causes testing difficulties for the motivating example, with regards to the lack of testability in digital twins outlined in Section 1.

Capturing the desired behaviour of digital twins requires a model-based representation. Current techniques rely on analysing the source code of systems to find faults [1, 2, 15]. These approaches are not suited to digital twins as they can contain complex models, possibly running across multiples services and sometimes even in the cloud with completely inaccessible source code. This can be seen in our motivating example as the source code for the arm's path planning algorithm is inaccessible. Analysing, or even obtaining, the source code for such models becomes an impossible task, requiring a different approach to represent digital twin behaviour.

Digital twins make the gathering and controlling of data required for testing very challenging. Traditional testing approaches run the model in a variety of configurations and observe the output. Digital twins are not this simple as both the physical and virtual domains must be accounted for when testing. Extensive set up and physical execution can be required for specific test cases which take into account physical stochasticity [17]. Our motivating example shares these difficulties due to its physical coupling. Therefore, we cannot rely on large data sets which contain the exact data required for desired test cases.

3 CAUSAL INFERENCE AND TESTING

Digital twins bring a number of difficulties that make the application of traditional testing techniques challenging. In this section, we explore Causal Inference and how it can help alleviate some of the issues addressed in Section 2.2. We then continue by outlining a causal testing procedure for use in the next sections.

Causal Inference (CI) is a family of methodologies that measure causal effects to make claims about causal relationships [11], such as investigating whether changing a particular line of code causes a fault [15]. CI techniques, such as Causal Directed Acyclic Graphs (DAGs) and counterfactuals, allow us to address the difficulties of testing digital twins by making strong claims on causality, without access to large, specific data sets.

3.1 Modelling unintelligible behaviour

CI can be used to test strong claims about causal relationships, in a way that conventional statistics cannot [13], because it is supplied with domain knowledge in the form of a causal DAG. DAGs are models that provide an intuitive graphical method for expressing the causal assumptions necessary to solve CI problems [12]. Informally, a causal DAG $G = (V, E)$ is a directed acyclic graph comprising a set of random variables V (the nodes) and directed edges E that capture causality, such that an edge $X \rightarrow Y$ denotes the assumption that changing X in isolation should cause Y to change.

In the context of digital twins, this presents an opportunity for the developer to represent their expectations of what correct relationships should be. This provides a simple, intuitive basis for capturing the complex interactions between the variables and states of a digital twin across multiple domains.

3.2 Handling small, uncontrolled data sets

A strength of CI is that DAGs can be reconciled with existing data sets to reason about causal relationships between variables. CI uses counterfactuals to reason about data which is not present in the data set [14], presenting a way in which specific data need not be obtained and allowing for smaller, less controlled data sets to be used. Such a technique allows for causal outcomes to be inferred without the need for specific executions or large data sets.

For testing digital twins, this means that test data for specific scenarios need not be present within the data set. This technique can be used to help alleviate the problem of running specific test cases as the required scenarios can be reasoned about without precise data being present, allowing existing datasets to be used.

3.3 Causal Testing

As shown above, CI makes it possible to answer causal questions from passively collected data that contains various forms of bias, such as confounding (informally, when the cause and effect share a common cause), instead of running costly experiments. In a software testing context, this has the potential to draw causal conclusions from existing test data and, therefore, predict the outcome of test cases that cannot be run [3]. Such an approach would help improve the testability of software that has a vast input space and is expensive to run, such as digital twins.

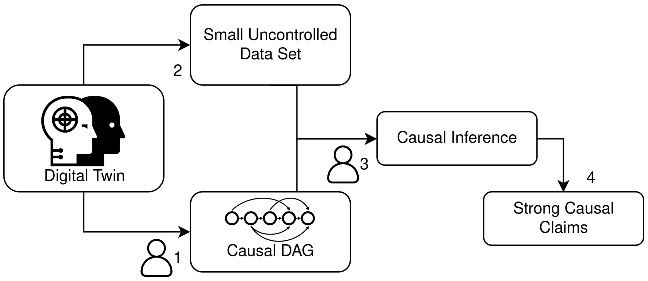

Figure 1: Using causal inference to test a digital twin. Stages 1-4 are outlined. People in the diagram represent the stages in which domain expertise is used

We outline the steps for causal testing, shown in Figure 1:

(1) Generating a causal DAG: domain expertise capture the intended behaviour of the digital twin in a causal DAG. Relationships within this DAG allow for causal claims to be made about the system. Complex behaviours can be outlined using this without understanding the exact internal structure of the system. DAGs are created manually by domain experts as there is no way to do this automatically.
(2) Defining a data set: Existing data from a digital twin can be used with regards to the variables present in the causal DAG. Due to the limitations of data collection outlined in Section 2.2, a smaller data set which does not contain the exact scenarios to be tested is adequate.
(3) Causal Inference: The generated DAG is used, alongside the obtained data set, to perform causal inference. Causal test cases are defined that intervene on a variable to measure the effect on another variable. The average treatment effect [9] as well as the confidence intervals of this intervention can be observed. Informally, given $X \rightarrow Y$, we use the causal DAG and data set to infer how an intervention to the value of X would affect the value of Y.
(4) Causal claims: The data collected from causal inference can be used to determine whether there is evidence that a causal relationship holds in the digital twin (the digital twin is behaving as expected) or if it does not hold (the digital twin is behaving unexpectedly).

4 ROBOT ARM CASE STUDY

This section follows the outlined approach in Figure 1. We use a causal DAG to represent the motivating example outlined in Section 2.1. We then generate a data-set from the digital twin and use this to test causal relationships in the model. We provide two example scenarios: a scenario where we are able to confirm a known causal relationship and one where a fault is localised to another specific causal relationship. Although the digital twin has 6 joints, we simplify this case study to only observe 2 consecutive joints.

4.1 Generating a Causal Diagram

A causal DAG was used to capture the assumptions of correct behaviour of the robotic arm (Figure 1 step 1), allowing for a representation of the arms motion. Figure 2 presents the initial state of the joints (*joint_1_start_rot*, *joint_2_start_rot*), the target given to the robot arm (*effector_target*), the desired final rotation of the joints (*joint_1_end_rot*, *joint_2_end_rot*) and the rotation of each joint half way through their motion (*joint_1_mid_rot*, *joint_2_mid_rot*). The joint rotations half way through the arm's motion were used to simply represent the path of each joint. This simplification does not capture the entire path and will be discussed in Section 5.

The causal relationships between variables in the causal DAG represent the intended behaviour of variables with respect to one another. This can be seen by the desired end rotation of the joints being causally affected by the given target position. We also assume that the path of a joint is dependent on its start and end positions as they define the beginning and end of that path.

To test the digital twin, we establish that there is an interdependency between the two joints under test. The arm's motion should be informed by the rotation of all joints throughout the motion. This ensures the behaviour is safe in a human robotic collaborative environment. Therefore, the path of a joint should depend on the initial and final states of other joints. This can be seen in Figure 2 by the start and desired end rotations of joint 1 affecting the rotation mid way through movement of joint 2 and vice versa. If each joint was not aware of each other joint, possibly dangerous and unexpected behaviour could occur.

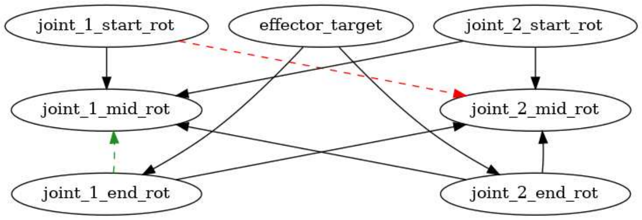

Figure 2: Generated causal DAG of the motivating example. Two causal relationships are highlighted for Scenario A (green) and Scenario B (red)

Interdependency, however, can cause issues in the creation of a causal DAG as it could have produced cycles in the causal relationships, which are not permitted. Causal Directed Cyclic Graphs (DCG) [22] as well as equation-based techniques [4] have been explored in the literature for representing more complex behaviours, such as cycles, as an alternative to causal DAGs and are further explored in Section 5.

4.2 Obtaining a Data-set

A major advantage of using CI is that it is possible to test from existing data sets without running the digital twin for specific test cases (Figure 1 step 2). The data required for the CI data set could be found by examining the causal DAG presented in Figure 2. We were able to run the digital twin model 105 times to produce a data-set which contained the required variables at the required time-steps for a variety of random start states and target positions. Although this digital twin was not connected to a physical asset, the model was only run a relatively small number of times to replicate the difficulties of generating data-sets from digital twin systems explored in Section 2.2. This small data-set meant that the exact data required for each test case would likely not be present, therefore requiring us to deal with counterfactuals. How these test cases are constructed is explored in Section 4.3.

4.3 Generating Test Cases

Causal test cases present evidence of a causal relationship by inferring the effect of a given intervention (Figure 1 step 3). This process is further explained in part 3 of Section 3.3. Multiple test cases can be executed with different intervention values, each providing evidence toward a causal relationship between the cause and effect variables. This evidence can either support known causal relationships, inferring that this relationship in the model is acting as expected, or oppose expected causal relationships, localising errors in how the system should be behaving to that specific relationship.

Each joint of the arm stored a rotation in radians between $-\pi$ and π. The range -2.8 to 2.8 was partitioned into intervals of 0.4

Table 1: Results of causal testing showing the Average Treatment Effect (ATE) and 95% Confidence Intervals (95% CIs) for both Scenario A and Scenario B. Evidence of causal relationships are highlighted in bold.

Int	ATE A	95% CIs A	ATE B	95% CIs B
-2.8	1.45	**0.98, 1.92**	0.14	-0.34, 0.62
-2.4	1.24	**0.84, 1.65**	0.12	-0.29, 0.53
-2.0	1.03	**0.70, 1.37**	0.10	-0.24, 0.44
-1.6	0.83	**0.56, 1.10**	0.08	-0.20, 0.35
-1.2	0.62	**0.42, 0.82**	0.06	-0.15, 0.26
-0.8	0.41	**0.28, 0.55**	0.04	-0.10, 0.18
-0.4	0.21	**0.13, 0.27**	0.02	-0.05, 0.09
0.4	-0.21	**-0.27, -0.14**	-0.02	-0.09, 0.05
0.8	-0.41	**-0.55, -0.28**	-0.04	-0.18, 0.10
1.2	-0.62	**-0.82, -0.42**	-0.06	-0.26, 0.15
1.6	-0.83	**-1.10, -0.56**	-0.08	-0.35, 0.20
2.0	-1.03	**-1.37, -0.70**	-0.10	-0.44, 0.24
2.4	-1.24	**-1.65, -0.84**	-0.12	-0.53, 0.29
2.8	-1.45	**-1.92, -0.98**	-0.14	-0.62, 0.34

and these values were taken as intervention values, as seen in Table 1. From this, we generated test cases which explored how deviating from a rotation of 0.0 radians on a joint would affect its own and other joints' planned motion.

We outline two scenarios to apply CI-based testing. Each tests a different causal relationship, highlighted in Figure 2. We then use the data obtained from CI to determine whether these causal relationships hold and if the digital twin is behaving correctly.

Scenario A - Confirming that joint_1_end_rot affects joint_1_mid_rot. This scenario examines a single relationship within the causal DAG generated for the motivating example. The specified target end point, and therefore the calculated end position of a joint, should always have a causal effect on the planned route for the same joint. In this case, the middle rotation of a joint will be causally affected by its end rotation. This relationship can be seen highlighted in green in Figure 2.

Scenario B - Confirming that joint_1_start_rot affects joint_2_mid_rot. This scenario explores a different relationship in the causal DAG for the motivating example. Each joint should take into account the movement of each other joint when planning the arm's movement to ensure a safe and informed movement. Therefore, a causal relationship between the start position of one joint and the planned route of another should be present for correct behaviour. This relationship can be seen highlighted in red in Figure 2.

4.4 Scenario Results

Table 1 displays the results of both Scenario A and B for each of the interventions (Int) described in Section 4.3. For each scenario, an average treatment effect (ATE) [9] as well as the 95% confidence intervals (95% CIs) were generated using causal inference based on the generated causal diagram and the obtained digital twin data-set. The ATE provides the average amount by which the effect variable is changed due to the intervention on the cause variable. The 95% confidence intervals represents the range in which the observed variable will likely lie for the given intervention. To provide evidence of a causal relationship, the 95% confidence intervals must not overlap the value zero (Figure 1 step 4).

For Scenario A, the ATE value of the middle rotation of the joint decreases as the intervention value of the end rotation of the joint is increased. This is also mirrored by the 95% confidence intervals which also decrease. These intervals do not overlap the value zero for any intervention and therefore all provide evidence of a causal relationship for the end rotation of a joint causing an effect on the middle rotation of a joint.

The ATE and 95% confidence intervals in Scenario B follow a similar trend, decreasing as the intervention to the initial rotation is increased. However, the 95% confidence intervals all contain the value zero and therefore do not provide evidence of a causal relationship. Informally, this means that joint 2's path is *not* calculated based on the initial position of joint 1. For a system in a safety critical setting, this misrepresentation of the actual system could lead to a situation where the digital twin deems certain movements to be safe that would be hazardous in the corresponding physical setting. Our technique was able to localise this fault to a single causal relationship, providing a starting point for developers to fix this behaviour in a complex, typically untestable model.

5 DISCUSSION

Having gathered results for our two scenarios, we use this section to discuss the effectiveness of CI based testing techniques for digital twins. We also discuss difficulties uncovered during the process of testing the motivating example.

Our results showed that we were able to identify correct behaviour and even uncover erroneous behaviour in a safety critical digital twin. These scenarios allowed for specific causal relationships to be tested and, therefore, enabled us to isolate a fault down to a single causal relationship. This technique allows developers to accurately find where erroneous behaviour needs to be fixed, reducing time searching through intricate digital twin models.

Our technique was able to overcome the challenges to testability posed by digital twins, successfully building upon previous causal inference based testing techniques [1, 2, 15], further explored in Section 6. We used an efficient data set with no specific test cases to perform testing. This allowed testing to not require expensive run times due to physical couplings, but instead could use existing, unspecific run time data. Our small data set was able to determine both correct and incorrect behaviour within the digital twin.

Digital twins do, however, present difficulties when testing with CI. Accurately representing complex digital twin behaviours, such as data driven components [19] and adaptive behaviour [7], require extensive domain knowledge. In our example, a simplification was made to represent the path of a joint as a single variable. In practice, this is not representative of the full path and could have lead to causal relationships within its execution being overlooked. Our motivating example also only tested direct causal relationships between variables. Testing more complex behaviour could require the use of causal mediation analysis [13], allowing for paths of multiple causal relationships to be observed. Outlining future works to accommodate these complex behaviours is outlined in Section 7.

6 RELATED WORK

Baah et al. [1, 2] propose a graphical based approach to fault diagnosis. An approach of extracting a program dependency graph and using its edges to statistically determine faults [2] was developed and then extended to use these graphs, in addition to causal inference, to perform fault localisation on source code [1]. This technique requires a causal diagram to be developed from a program dependence graph of the source code, allowing causal relationships to be tested within it. However, access to the source code to generate causal graphs is not always possible with digital twins as their models can be too complex for this technique and also extend across both physical and digital spaces.

These methods have been expanded by Podgurski et al. [15], allowing for faults to be identified using causal inference to intervene on specific lines of code. This approach, however, still requires access to the source code and would not be suitable for the complex, multi-domain models found in digital twins. Our method addresses the limitations and difficulties of these approaches through capturing domain expertise in causal DAGs, allowing the testing of more systems using causal inference without accessing their source code.

Wang et al. [21] designed a digital twin which provides fault diagnosis for its physical system. It models a rotor blade and determines when the blade of the physical system has become unbalanced. We build upon this work by providing a technique which can not only determine faults in a physical system, but also allows for cross domain testing with a greatly reduced observed data set.

At the time of writing, no studies could be found relating to causal inference based testing of digital twins. There are a number of techniques for testing complex and unintelligible systems [20] which explore multiple techniques to test these models, primarily for model inference. We will comparing these approaches to our technique to allow for limitations and enhancements to be uncovered in future works.

7 CONCLUSION AND FUTURE WORKS

In this paper, we explore a method of using causal inference based testing to alleviate the challenges of testing digital twins posed by their lack of testability. We build upon existing methods [1, 2, 15] by using causal DAGs to capture expected causal properties about a digital twin's intended behaviour. This attempts to solve the limitations of existing methods in capturing the complex models found in these systems. We also use counterfactuals to allow for reduced data sets, which need not contain specific test cases, to be used when testing physically coupled systems, reducing the time and expense of acquiring an extensive controlled testing data set.

We explore an industrial digital twin framework to show the validity and limitations of our technique. In doing so, we show how, using causal DAGs, we are able to confirm the presence of a causal relationship and, conversely, able to identify and localise a new fault, which causes dangerous behaviour in a safety critical environment, to a single causal relationship.

In future work, we aim to examine more digital twins to test more intricate behaviour. In doing this, we aim to understand how to ensure coverage of causal relationships within the complex and inaccessible behaviours of a digital twin. Behaviours which include time-based actions, neural networks and adapting to the environment present further challenges to testing digital twins. Ensuring the creation of accurate causal DAGs will allow domain expertise inform this process, improving the testability of digital twins.

REFERENCES

[1] George K. Baah, Andy Podgurski, and Mary Jean Harrold. 2010. Causal Inference for Statistical Fault Localization. In *Proceedings of the 19th International Symposium on Software Testing and Analysis* (Trento, Italy) *(ISSTA '10)*. Association for Computing Machinery, New York, NY, USA, 73–84. https://doi.org/10.1145/1831708.1831717

[2] George K. Baah, Andy Podgurski, and Mary Jean Harrold. 2010. The probabilistic program dependence graph and its application to fault diagnosis. *IEEE Transactions on Software Engineering* 36 (2010), 528–545. Issue 4. https://doi.org/10.1109/TSE.2009.87

[3] Andrew G. Clark, Michael Foster, Benedikt Prifling, Neil Walkinshaw, Robert M. Hierons, Volker Schmidt, and Robert D. Turner. 2022. Testing Causality in Scientific Modelling Software. https://doi.org/10.48550/ARXIV.2209.00357

[4] Constantin Corduneanu. 2002. *Functional Equations with Causal Operators.* CRC Press, London. https://doi.org/10.4324/9780203166376

[5] J A Douthwaite, B Lesage, M Gleirscher, R Calinescu, J M Aitken, R Alexander, and J Law. 2021. *A Modular Digital Twinning Framework for Safety Assurance of Collaborative Robotics.* Technical Report. The University of Sheffield.

[6] Waguih ElMaraghy, Hoda ElMaraghy, Tetsuo Tomiyama, and Laszlo Monostori. 2012. Complexity in engineering design and manufacturing. *CIRP Annals* 61, 2 (2012), 793–814. https://doi.org/10.1016/j.cirp.2012.05.001

[7] Jonathan Eyre, Sam Hyde, Daniel Walker, Seun Ojo, Oliver Hayes, Robin Hartley, Rab Scott, and Jonathan Bray. 2021. *Untangling the requirements of a Digital Twin.* Technical Report. Advanced Manufacturing Research Centre. https://www.amrc.co.uk/pages/digital-twin-report

[8] Aidan Fuller, Zhong Fan, Charles Day, and Chris Barlow. 2020. Digital Twin: Enabling Technologies, Challenges and Open Research. *IEEE Access* 8 (2020), 108952–108971.

[9] Miguel Hernan and James Robins. 2020. *Causal Inference: What If.* Boca Raton: Chapman & Hall/CRC.

[10] Yuling Hsu, Jing-Ming Chiu, and John S. Liu. 2019. Digital Twins for Industry 4.0 and Beyond. In *2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM).* 526–530. https://doi.org/10.1109/IEEM44572.2019.8978614

[11] Luke Keele. 2015. The statistics of causal inference: A view from political methodology. *Political Analysis* (2015), 313–335.

[12] Judea Pearl. 1995. causal graphs for empirical research. *Biometrika* 82, 4 (1995), 669–688.

[13] Judea Pearl et al. 2009. Causal inference in statistics: An overview. *Statistics surveys* 3 (2009), 96–146.

[14] Judea Pearl and Dana Mackenzie. 2018. *The book of why: the new science of cause and effect.* Basic books.

[15] Andy Podgurski and Yiğit Küçük. 2020. CounterFault: Value-Based Fault Localization by Modeling and Predicting Counterfactual Outcomes. In *2020 IEEE International Conference on Software Maintenance and Evolution.* IEEE, 382–393.

[16] Universal Robotics. [n.d.]. UR10E collaborative industrial robot - payload up to 12.5 kg. https://www.universal-robots.com/products/ur10-robot/

[17] Angira Sharma, Edward Elson Kosasih, Jie Zhang, Alexandra Brintrup, and Anisoara Calinescu. 2020. Digital Twins: State of the Art Theory and Practice, Challenges, and Open Research Questions. *CoRR* abs/2011.02833 (2020). arXiv:2011.02833

[18] IEEE ISO Standards. 2017. *Systems and software engineering — Vocabulary: ISO/IEC/IEEE 24765:2017.* Standard.

[19] D. J. Wagg, K. Worden, R. J. Barthorpe, and P. Gardner. 2020. 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 2020). https://doi.org/10.1115/1.4046739 030901.

[20] Neil Walkinshaw. 2018. *Testing Functional Black-Box Programs Without a Specification.* Springer International Publishing, Cham, 101–120. https://doi.org/10.1007/978-3-319-96562-8_4

[21] Jinjiang Wang, Lunkuan Ye, Robert X. Gao, Chen Li, and Laibin Zhang. 2019. Digital Twin for rotating machinery fault diagnosis in smart manufacturing. *International Journal of Production Research* 57, 12 (2019), 3920–3934. https://doi.org/10.1080/00207543.2018.1552032

[22] Qin Zhang. 2015. Dynamic uncertain causality graph for knowledge representation and probabilistic reasoning: Directed cyclic graph and joint probability distribution. *IEEE Transactions on Neural Networks and Learning Systems* 26 (7 2015), 1503–1517. Issue 7. https://doi.org/10.1109/TNNLS.2015.2402162