01 Jan 2023

Development Of A MATLAB-based Structural Analysis Toolbox For Sensor Placement In A Multi-domain Physical System

Arnold A. Fernandes

Kartikeya J.P. Veeramraju

Jonathan W. Kimball
Missouri University of Science and Technology, kimballjw@mst.edu

IEEE Control Systems Letters paper presented at
2023 American Control Conference (ACC)
San Diego, CA, USA. May 31 - June 2, 2023

Development of a MATLAB-based Structural Analysis Toolbox for Sensor Placement in a Multi-domain Physical System†

Arnold A. Fernandes*, Kartikeya J.P. Veeramraju, and Jonathan W. Kimball

Abstract— **This paper discusses a Bond Graph (BG) Structural Analysis Toolbox developed in MATLAB® (MATSAT) that performs causal analysis on the BG and assists the user in the sensor selection process for a multi-domain physical system. MATSAT contains modules for performing the Sequential Causality Assignment Procedure (SCAP) and Causal Path Search (CaPS). The modules can be combined to check for structural properties such as structural observability (SO) for any sensor set. The working of MATSAT is shown for standard systems. Verification of SCAP, CaPS, and necessary and sufficient SO conditions is shown.**

I. INTRODUCTION

The Bond Graph (BG) modeling approach (sometimes qualified as a power bond graph) provides a unified way of studying a multi-domain physical system. Contrary to the traditional approach in which a researcher works on an individual component in a particular physical domain, using a BG allows the researcher to convert physical system models to a power-based graphical construct. These power-based graphical constructs from different domains can be merged together and can be made as detailed as desired. BG allows the researcher to study the dynamics of the overall system. This could be particularly useful in analyzing the impact of distributed actuation as well as the sensing of system state via sensor placements in large-scale Cyber-Physical Systems (CPS). Qualitative analysis of BGs can also provide insight into the robustness of the system under parameter variation i.e. does the system maintain Structural Observability (SO) and Structural Controllability (SC) for the given sensor and actuator sets respectively. This could provide insight to the researcher to modify the components of the physical system or place more actuators/sensors. The current work strives to provide a BG toolbox to analyze sensor placement in a multi-domain linear physical system. Specifically, the toolbox determines the SO of the system under different sensor configurations, thus allowing the system designer to select a sensor set.

The necessary and sufficient conditions for SO/SC given the junction structure matrix S of the BG of a linear system were first derived for SISO systems in [1] and extended to MIMO systems in [2]. SO and SC were then shown to be dependent on BG storage elements taking preferred integral/derivative causality in [3]. This form of SO and SC is the basis of structural analysis of linear systems modeled using BG [4]. It has also been used for process supervision [5]. The current work looks at developing a software tool in MATLAB® to facilitate structural analysis, sensor placement, and provide a stepping stone to process supervision of a large-scale CPS. To the best knowledge of the authors, no toolbox provides suggestions on sensor sets that can render a system structurally observable.

A CPS generally consists of multiple physical domains interacting, to provide one or more services. System designers and operators must ensure proper operation of the CPS. Logical predicates known as invariants can be constructed for a sub-system that should hold true to ensure a proper operation of the system. Unified invariants then are invariants that should hold true across multiple domains [6]. Unified invariants are difficult to generate as they require having the entire system model described in one mathematical language while also being analyzed by experts in each domain. This is where the BG paradigm can help. MATSAT is a step towards automating unified invariant generation. However, the current work only looks at analyzing the SO of the system state which is a necessary condition for determining Structural Monitorability (SM).

Previously in [7] a microgrid, a type of CPS, was simulated and validated using BGs. However, only the quantitative aspects of BGs were considered. The present work provides instead the framework for considering the qualitative aspects, which removes the need for computations and parameter knowledge, thereby making it robust.

BGs are useful for modeling as well as sensor placement. Unlike other modeling methods, in a BG the physical location of a sensor is displayed explicitly. In [8] a heuristic sensor placement method is provided that circumvents the need for deriving Analytical Redundancy Relations (ARRs) for Fault Detection and Isolation (FDI) purposes. The current paper however provides a software tool that automates the sensor placement process as far as SO is concerned. No heuristic method is provided as the tool is designed to be a platform that can test generic sensor placement algorithms. In [9] behavioral, causal, and structural properties of BG are used for checking SM and performing an optimal sensor placement such that the system is monitorable in the presence of faults. The paper [10] shows an optimal sensor placement strategy for fault detection and isolation obtained after converting a BG to a digraph. Here faults are added to each BG element. If on the digraph the faults propagate to a sensor then the fault is isolable, else sensors are added

†This work was supported in part by the National Science Foundation under award 1837472.
Arnold A. Fernandes, Kartikeya J.P. Veeramraju, and Jonathan W. Kimball are with the Department of Electrical and Computer Engineering, Missouri University of Science and Technology, Rolla, MO, USA. {af2vc, kvkhh, kimballjw}@mst.edu
*Corresponding author: Arnold A. Fernandes

TABLE I: Power variables in multiple physical domain [5].

Domain	Effort (e)	Flow (f)
Electrical	Voltage	Current
Mechanical Rotation	Torque	Angular velocity
Mechanical Translation	Force	Velocity
Hydraulics	Pressure	Volume
Thermodynamics	Temperature	Entropy flow

to isolate the fault. In [11] a sensor placement algorithm is showcased that does not need ARR computation and only uses causal path information. A BG is converted to a tripartite graph consisting of elements to be monitored, sensors, and causal paths connecting the two. The notion of degree of observability/multiple redundancy for BGs is introduced here. In a recent paper [12], the BG FDI methods have been integrated with principal component analysis for a six tank system. The aforementioned papers each implement algorithms on a single BG model. With the use of MATSAT proposed in the current paper, algorithms could be tested on a library of BG models and detailed conclusions could then be drawn.

The paper is organized as follows. Section II gives a brief overview of the BG modeling paradigm. Section III details the structural properties of BGs while Section IV introduces the MATSAT and its modules that perform structural analysis on a BG. Section V validates MATSAT and Section VI concludes the paper.

II. Background

A. BG representation of a physical system

A BG looks at the flow of power in physical systems. As all physical systems use power as a currency, it is possible to describe the interactions of different components in a physical domain as well as interactions among different physical domains by the use of a pair of complementary power variables: an across variable, and a through variable. In the BG terminology, the across variable is termed as an effort variable while the through variable is termed as a flow variable. In electrical systems, the flow variable is current, and the effort variable is voltage.

BG classifies elements as: source, sensor, storage, dissipator, transducer, and junction. The Table II shows the element name, BG representation, and computation causality. Elements are also classified as active, passive, and zero power. The sources are active as they deliver power; storage and dissipative elements are passive as they store and consume power respectively; transducers and junctions neither store nor consume power. Each bond shows the direction of power flow by the use of a half arrow $\rightharpoonup$ which can be seen in the BG representations in Table II. The effort and flow variables are denoted on the bond by e, and f respectively. The computational causality is shown by the use of a causal stroke at one of the ends of the half arrow e.g. **X** $\xrightarrow[f]{e}|$ **Y.** The causal stroke indicates that BG element **X** determines the effort of BG element **Y**; the element **Y** determines the flow of element **X**. Mathematically,

$$e_Y := e := e_X, \tag{1}$$

OR

$$f_X := f := f_Y, \tag{2}$$

TABLE II: Element Name, BG representation, and Computational Causality [5].

Element Name	BG Representation	Computational Causality
Source effort	**Se:e** $\xrightarrow[f_1]{e_1}|$	$\begin{cases} e_1(t) := e(t) \\ f(t) \text{ is arbitrary} \end{cases}$
Source flow	**Sf:f** $|\xrightarrow[f_1]{e_1}$	$\begin{cases} f_1(t) := f(t) \\ e(t) \text{ is arbitrary} \end{cases}$
Detector effort	$\xrightarrow[f_1=0]{e_1}|$ **De:e**	$\begin{cases} e(t) := e_1(t) \\ f(t) := 0 \end{cases}$
Detector flow	$|\xrightarrow[f_1]{e_1=0}$ **Df:f**	$\begin{cases} f(t) := f_1(t) \\ e(t) := 0 \end{cases}$
Dissipator	$\xrightarrow[f_1]{e_1}|$ **R:R**	$f_1(t) := \frac{1}{R} \cdot e_1(t)$
	$|\xrightarrow[f_1]{e_1}$ **R:R**	$e_1(t) := R \cdot f_1(t)$
Storage C (int.)	$|\xrightarrow[f_1]{e_1}$ **C:C**	$e_1(t) := \frac{1}{C} \cdot \int f_1(\tau) d\tau$
Storage C (der.)	$\xrightarrow[f_1]{e_1}|$ **C:C**	$f_1(t) := C \cdot \frac{\Delta e_1}{\Delta t}$
Storage I (int.)	$\xrightarrow[f_1]{e_1}|$ **I:I**	$f_1(t) := \frac{1}{I} \cdot \int e_1(\tau) d\tau$
Storage I (der.)	$|\xrightarrow[f_1]{e_1}$ **I:I**	$e_1(t) := I \cdot \frac{\Delta f_1}{\Delta t}$
Transducer TF	$|\xrightarrow[f_1]{e_1}$ **TF:r** $|\xrightarrow[f_2]{e_2}$	$\begin{cases} f_2(t) := r \cdot f_1(t) \\ e_1(t) := r \cdot e_2(t) \end{cases}$
Transducer TF	$\xrightarrow[f_1]{e_1}|$ **TF:r** $\xrightarrow[f_2]{e_2}|$	$\begin{cases} f_1(t) := r \cdot f_2(t) \\ e_2(t) := r \cdot e_1(t) \end{cases}$
Transducer GY	$\xrightarrow[f_1]{e_1}|$ **GY:r** $|\xrightarrow[f_2]{e_2}$	$\begin{cases} f_2(t) := r \cdot e_1(t) \\ f_1(t) := r \cdot e_2(t) \end{cases}$
Transducer GY	$|\xrightarrow[f_1]{e_1}$ **GY:r** $\xrightarrow[f_2]{e_2}|$	$\begin{cases} e_2(t) := r \cdot f_1(t) \\ e_1(t) := r \cdot f_2(t) \end{cases}$
Junction 1	$\xrightarrow[f_1]{e_1}|$ **1:1$_1$** $|\xrightarrow[f_2]{e_2}$; bond (e_3, f_3) upward from **1:1$_1$**	$\begin{cases} e_3(t) := e_1(t) + e_2(t) \\ f_1(t) := f_3(t) \\ f_2(t) := f_3(t) \end{cases}$
Junction 0	$|\xrightarrow[f_1]{e_1}$ **0:0$_1$** $|\xrightarrow[f_2]{e_2}$; bond (e_3, f_3) upward from **0:0$_1$**	$\begin{cases} f_2(t) := f_1(t) + f_3(t) \\ e_1(t) := e_2(t) \\ e_3(t) := e_2(t) \end{cases}$

where $:=$ is the assignment symbol that assigns the effort value at **X** to **Y**. The above equations should not be confused

with constitutive equations, which only consider the equal (=) symbol and not the assignment symbol. Computational causality allows for modeling and simulating the physical system in a computer. Computational causality for all linear BG elements are given in Table II.

B. *Causality assignment*

The BG storage elements can take integral causality as well as derivative causality. The detailed explanation for integral and derivative causality can be found in [3]. Storage elements are required to take a preferred integral or a preferred derivative causality to satisfy certain structural conditions (see Theorem 1). The causality assignment of a bond graph is performed by the Sequential Causality Assignment Procedure (SCAP), detailed in [13].

MATSAT contains a module that performs SCAP. When SCAP is performed with preferred derivative causality for the storage elements, it will be denoted by SCAP-D; when performed for preferred integral causality, it will be denoted as SCAP-I.

III. Structural Properties

After SCAP-I or SCAP-D is performed, certain BG structural properties are revealed. These properties are listed below. For in-depth detail refer [3].

Property 1. *The order of a BG is the number of storage elements that take integral causality when SCAP-I is performed.*

Property 2. *For a linear system, the rank is the number of storage elements that take derivative causality when SCAP-D is performed.*

Definition 1. *A causal path is a sequence of power bonds having the causal strokes at the same end while going from one element to another with causal stroke being reversed across a two-port gyrator.*

The properties and definition mentioned above are inherently utilized to check a systems structural observability, which decides if a given sensor configuration allows a systems states to be observable as is specified by the following theorem.

Theorem 1. *A linear time invariant MIMO system with matrices A and C is structurally observable if and only if the following two conditions are satisfied [2]–[4].*

1) *Attainability/reachability/necessary condition: Given that SCAP-I is performed on a BG, every storage element in integral causality must have at least one causal path to a sensor.*
2) *Sufficient condition: Energy stores taking integral causality when SCAP-I is performed should take a derivative causality when SCAP-D is performed. In the possibility that this condition is not met directly, it can be achieved by replacing some sensors in appropriate places by their dual.*

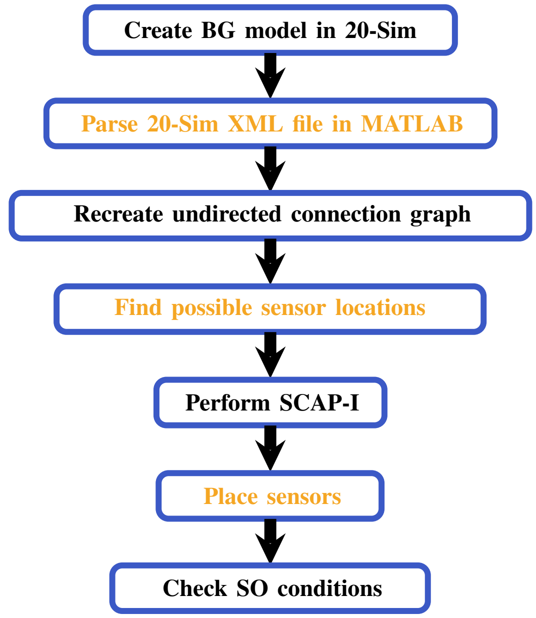

Fig. 1: Use of MATSAT to evaluate SO.

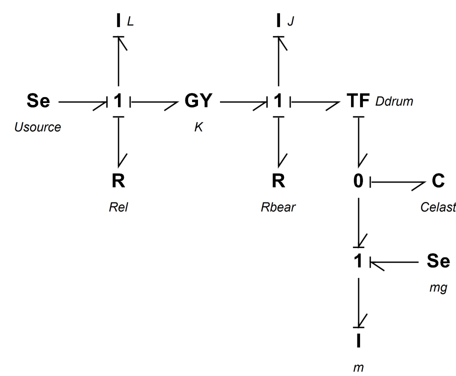

Fig. 2: BG of the hoist model in 20-Sim [13].

IV. Usage of MATSAT

The current functionality of MATSAT performs automated analysis of SO. Figure 1 highlights all the modules used. To begin with, MATSAT parses through a 20-Sim .emx file which contains XML code. Next, information gathered is utilized to create the connection graph among various BG elements in a MATLAB environment. Possible locations at which effort or flow sensors can be attached are then found out. A causal assignment is then performed at each of the bonds using the SCAP-I. Sensors are then placed combinatorially and SO conditions are checked in the end.

A. *Text parser*

20-Sim provides a GUI to easily create BGs but it does not contain tools to perform structural analysis on the BGs nor does it provide tools to check for optimal sensor placement. MATSAT extends the usability of 20-Sim. The in-built text parser takes 20-Sim XML code and generates a connection graph. A BG of the hoist model [13] is considered throughout

the paper to illustrate the working of MATSAT. This model is represented in 20-Sim as shown in Figure 2. The XML file is then read by the text parser module which gives the undirected graph presented in Figure 4a.

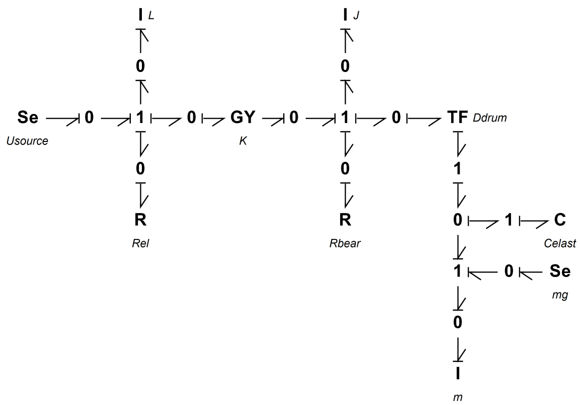

Fig. 3: Expanded BG of the hoist model showing possible sensor location junctions.

B. Sensor location

The sensor location module checks the connection graph and adds in extra **1** and **0** nodes so that it is possible to measure each and every effort and flow on the graph e.g. if an element's flow needs to be measured, a **1** junction is added to the connection graph. Figure 3 shows possible sensor location junctions on a BG while 4b is the output given by MATSAT.

C. SCAP-I

The SCAP-I module performs SCAP, with storage elements getting a preferred integral causality. The working of the SCAP-I module can be seen in Figure 4c. The effort out causality is indicated by an arrow head on the digraph.

D. Sensor placement

This module places a sensor on the BG. It places an effort sensor at a **0** junction and/or a flow sensor at a **1** junction. The sensors are placed with fixed causalities. The effort sensor/detector **De** has an effort out causality while the flow sensor **Df** has a flow out causality with complimentary effort and flow variables respectively taking a zero value as no power is consumed by the sensors.

E. Causal Path Search (CaPS)

Once the sensor is placed, the Causal Path Search (CaPS) algorithm starts a causal path search from a sensor node and tries to reach a storage node. The search begins from flow sensors on an Effort Causal Graph (ECG). A Depth-First-Search (DFS) is performed until all terminal node on the ECG are reached. Then the search is carried on a Flow Causal Graph (FCG) by performing a DFS from the ECG terminal nodes or effort sensors until terminal nodes on the FCG are reached. This search is repeated until all the storage elements are reached or until a manual termination condition occurs. For the causal hoist model, a sensor is placed at node 25 and the ECG is obtained as shown in Figure 5a, and the FCG is in Figure 5b.

F. SCAP-D

The SCAP-D module performs SCAP while ensuring storage elements get a preferred derivative causality. The Figure 5c shows the working of the SCAP-D module after a sensor has been placed. Note that unlike SCAP-I, SCAP-D is generally performed after a sensor has been placed.

G. Structural Observability

The SO module makes use of the SCAP-I, CaPS, and SCAP-D modules. It first makes a note of the storage elements taking integral causality under SCAP-I i.e the order of the system. Next, the existence of a causal path from sensors to storage elements is assessed. Finally, it performs SCAP-D on the BG and checks if the storage elements that initially took preferred integral causality under SCAP-I, now take a preferred differential causality under SCAP-D i.e. the rank and the order of a system are the same.

V. Verification of MATSAT

In this section, causality tests will be first performed followed by SO tests for single sensor placements to check the working of MATSAT.

A. Causality tests

Causality tests are performed to validate the causal assignments after SCAP-I has been applied by MATSAT to junction elements. Due to space limitations, tests under SCAP-D are not shown. Figure 6 shows causally assigned BGs designed to test the causal assignments at **0** and **1** junctions. The BGs are created in 20-Sim and the generated XML files are parsed by the MATSAT Text parser module. The SCAP-I module performs causality assignment and generates the causal BGs in Figure 7. Subfigure 6a shows a **1** junction connected to two effort sources and one dissipator. The effort sources have fixed effort-out causality which fixes the causality of the third port of the **1** junction to be an effort-out causality. The same result is obtained by MATSAT in Subfigure 7a. The rest of the subfigures can be similarly compared and are provided for completeness.

B. SO tests for single sensor placement

To verify that MATSAT correctly classifies a system as SO, it is crucial to check if it analyzes both necessary and sufficient conditions correctly. Hence in the following subsections both the necessary and sufficient conditions are tested individually.

1) Test for SO necessary condition: In Table III the SO analysis results for the hoist model are shown. Placing a sensor on node 25 is shown to make the system SO. This can be verified by looking at Figure 5a and Figure 5c. Basically, all storage elements can be reached from node 25 and all storage elements take preferred causality under both SCAP-I and SCAP-D. The Table III also mentions that on placing sensors at node 18, and 24, the system is not SO. This can

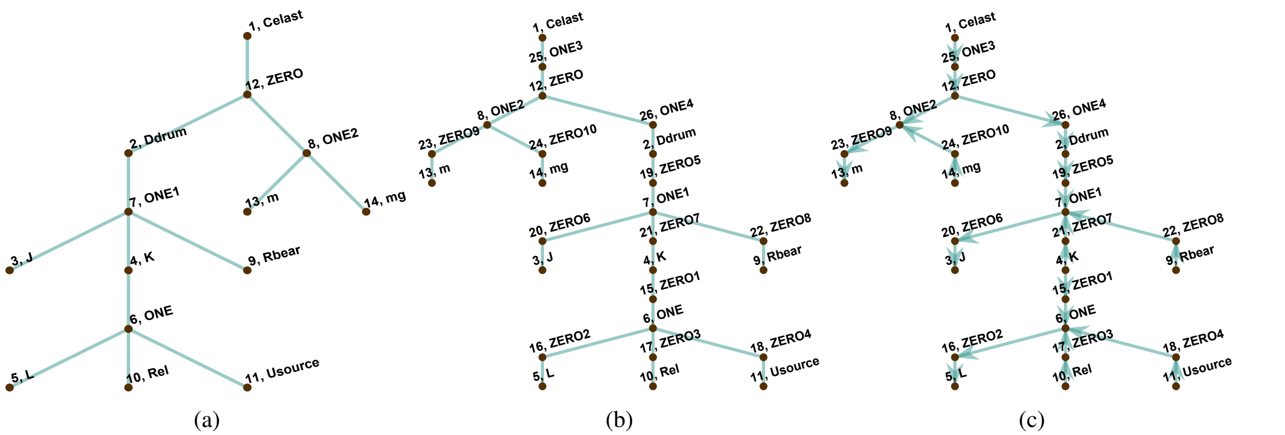

Fig. 4: Graphs obtained by various modules (a) connection graph obtained by the text parser module, (b) expanded graph shows locations for sensor attachment, and (c) Effort Causal Graph (ECG) given by SCAP-I module.

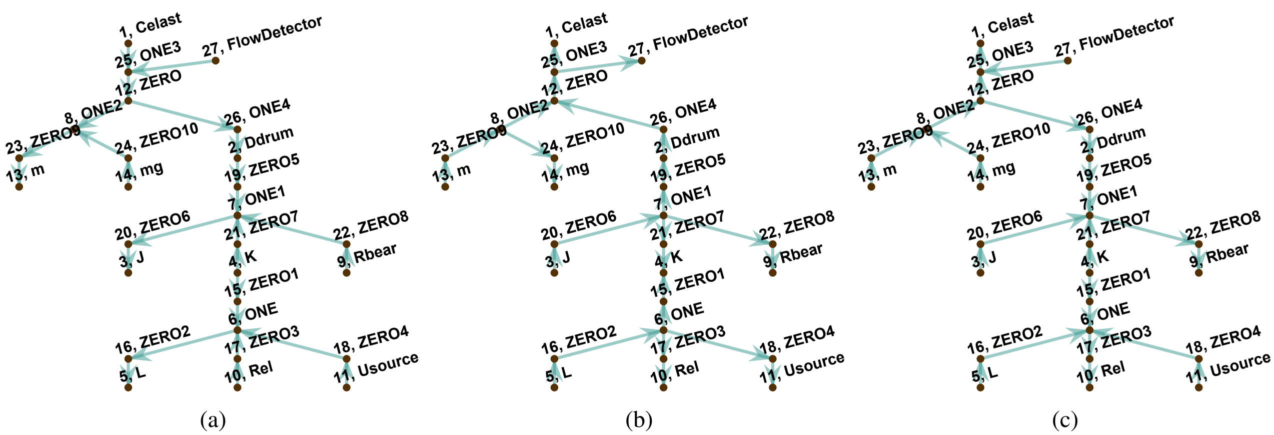

Fig. 5: Graphs obtained after sensor placement (a) ECG after sensor placement (b) Flow Causal Graph (FCG) obtained after flipping edges of the ECG (c) graph obtained after SCAP-D on a connection graph with sensors.

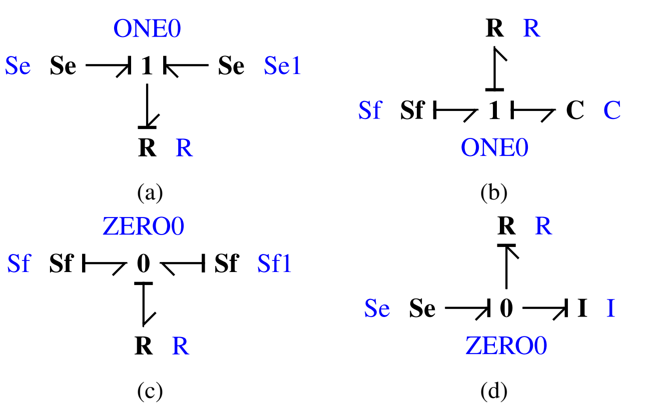

Fig. 6: BGs given to MATSAT to verify Junction **1** and **0** causal restrictions (a) **1** junction with $n-1$ effort-in causality (b) **1** junction with 1 flow-in (effort-out) causality (c) **0** junction with $n-1$ flow-in (effort-out) causality (d) **0** junction with 1 effort-in causality

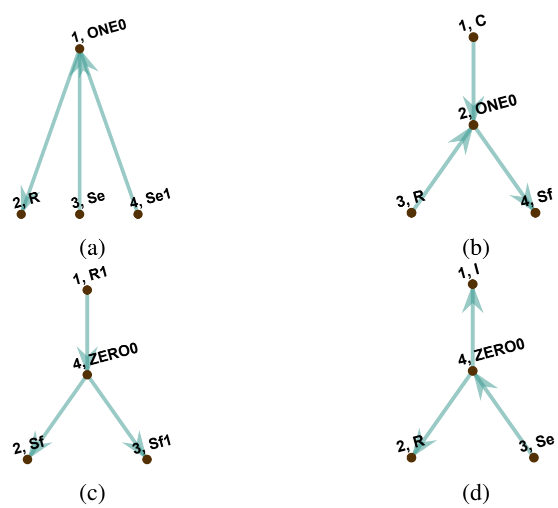

Fig. 7: Causal assignment of **1** and **0** junction performed by MATSAT (a) **1** junction with $n-1$ effort-in (flow-out) causality (b) **1** junction with 1 flow-in (effort-out) causality (c) **0** junction with $n-1$ flow-in causality (d) **0** junction with 1 effort-in causality.

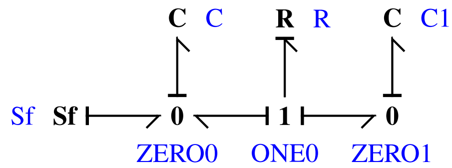

Fig. 8: BG of a two tank example [5].

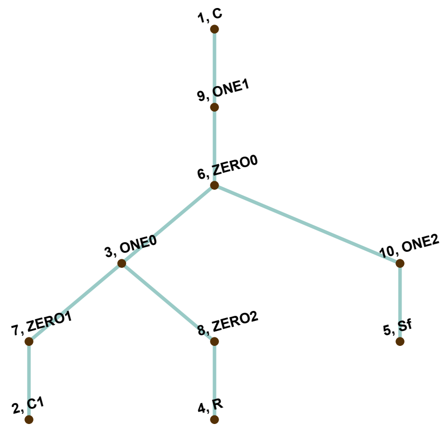

Fig. 9: Graph showing junctions for sensor placement.

be easily verified. Nodes 18 and 24 are connected to ideal effort sources and thus the effort measured at those nodes are not impacted by storage elements. Alternately, no causal paths exist from storage elements to effort sensors at those nodes. Thus the software correctly evaluates the necessary condition required for SO.

2) Test for SO sufficient condition: The BG of the hybrid two tank system mentioned in [5] is depicted in Figure 8. The corresponding connection graph after application of the sensor location module of MATSAT is given in Figure 9. It was shown in [5] that the system is not SO when a sensor is attached to ONE0; however the system is SO if a sensor is placed on ZERO1. MATSAT also defines these nodes as node 3 and node 7 respectively in Figure 9. Basically, if the sensor is placed at node 3, the storage C_1 does not get a preferred derivative causality when SCAP-D is performed. However, if the sensor is placed at node 7 all storage elements take preferred derivative causality under SCAP-D. MATSAT arrives at the same conclusion as can be seen in Table IV. Additionally, MATSAT provides suggestions of alternate locations to place/not place sensors. Thus the software correctly evaluates the sufficient condition for SO.

VI. Conclusions

This paper showcases MATSAT, a MATLAB toolbox designed to analyze BG structural properties. The paper also verifies proper working of MATSAT and illustrates the information it provides to a user regarding sensor locations that guarantee SO. Future work will look at sensor placements that guarantee SM. The linearity assumption will be relaxed and systems with algebraic/differential loops will be considered. Further work to incorporate SM is underway after which the code will be available on GitHub.

TABLE III: SO analysis on hoist model (not all placements shown).

Sensor Junction	Sufficiency Flag	Necessity Flag	SO Flag
6	1	1	1
7	1	1	1
25	1	1	1
18	1	0	0
24	1	0	0

TABLE IV: SO analysis of hybrid two tank model.

Sensor Junction	Sufficiency Flag	Necessity Flag	SO Flag
3	0	1	0
9	0	1	0
10	0	0	0
6	1	1	1
7	1	1	1
8	0	1	0

References

[1] C. Sueur and G. Dauphin-Tanguy, “Structural controllability/observability of linear systems represented by bond graphs,” *Journal of the Franklin Institute*, vol. 326, no. 6, pp. 869–883, 1989.
[2] C. Sueur and G. Dauphin-Tanguy, “Bond-graph approach for structural analysis of mimo linear systems,” *Journal of the Franklin Institute*, vol. 328, no. 1, pp. 55–70, 1991.
[3] G. Dauphin-Tanguy, A. Rahmani, and C. Sueur, “Bond graph aided design of controlled systems,” *Simulation Practice and Theory*, vol. 7, no. 5-6, pp. 493–513, 1999.
[4] W. Borutzky, *Bond graph methodology: development and analysis of multidisciplinary dynamic system models*. Springer Science & Business Media, 2009.
[5] A. K. Samantaray and B. O. Bouamama, *Model-based process supervision: a bond graph approach*. Springer, 2008.
[6] T. Paul, *Application of unified invariants for cyber physical systems in smart grids, Ph.D. dissertation*. Missouri University of Science and Technology, 2015.
[7] M. J. White, *Bond graph modeling of critical infrastructures for cyber-physical security implementation, M.S. thesis*. Missouri University of Science and Technology, 2021.
[8] M. Khemliche, B. O. Bouamama, and H. Haffaf, “Optimal sensor placement using bond-graph model for fdi design,” *IFAC Proceedings Volumes*, vol. 37, no. 5, pp. 79–84, 2004.
[9] M. A. Djeziri, B. O. Bouamama, R. Merzouki, and G. Dauphin-Tanguy, “Optimal sensor placement for fault diagnosis,” in *2009 IEEE International Conference on Mechatronics*, pp. 1–6, IEEE, 2009.
[10] S. Alem and D. Benazzouz, “Bond graph to digraph conversion: A sensor placement optimization for fault detection and isolation by a structural approach,” *Sadhana*, vol. 39, no. 5, pp. 1151–1164, 2014.
[11] M. Benallel, H. Haffaf, and A. Meghebbar, “Sensor placement for monitoring systems modelled by bond graph.,” *Electrotehnica, Electronica, Automatica*, vol. 64, no. 1, 2016.
[12] M. Said, H. Lahdhiri, and O. Taouali, “Monitoring nonlinear system using bond graph and pca method,” in *2019 International Conference on Advanced Systems and Emergent Technologies (IC_ASET)*, pp. 28–33, IEEE, 2019.
[13] J. F. Broenink, “Introduction to physical systems modelling with bond graphs,” *SiE whitebook on simulation methodologies*, vol. 31, no. 2, 1999.