Available online at www.sciencedirect.com
ScienceDirect
Simulation Modelling Practice and Theory 17 (2009) 175–196
SIMULATION MODELLING PRACTICE AND THEORY
www.elsevier.com/locate/simpat

Structural simplification of modular bond-graph models based on junction inactivity

Tulga Ersal *, Hosam K. Fathy, Jeffrey L. Stein

Department of Mechanical Engineering, The University of Michigan, Ann Arbor, MI 48109, United States

Received 17 February 2007; received in revised form 2 November 2007; accepted 16 February 2008
Available online 2 March 2008

Abstract

The modular modeling paradigm facilitates the efficient building, verification and handling of complex system models by assembling them from general-purpose component models. A drawback of this paradigm, however, is that the assembled system models may have excessively complex structures for certain purposes due to the amount of detail in the component models that has been introduced to promote modularity. For example, a multibody system can be modeled using generic rigid-body models with 6 degrees-of-freedom (DoF) to represent the components of the system, but then constraints have to be added to the model to match the DoF of the system. This work presents a structural simplification technique that can detect such unnecessary complexities in a modular bond-graph system model and eliminate them without compromising accuracy. To this end, the activity concept in the literature is extended to define "inactivity" for junction elements, and simplification is obtained by detecting and eliminating inactive junction elements and by propagating the implications. It is shown by example that this simple idea can result in models that are conceptually and computationally more efficient than the original modular models. The realization-preserving and input-dependent characteristics of this approach are highlighted.

Keywords: Modular modeling; Bond graph; Structural simplification; Junction inactivity

1. Introduction

One possible approach to modeling dynamic systems is to develop modular models for the components first and then to assemble the system model by combining the component models in accordance with the system topology [18]. Such a modular approach is well established in the multibody dynamics area, for example, where the system model is obtained by augmenting 6-DoF rigid-body models with constraints. Commercial software based on this idea is available (e.g., ADAMS [25]).

* Corresponding author. Tel.: +1 734 763 7388.
E-mail address: tersal@umich.edu (T. Ersal).

1569-190X/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.simpat.2008.02.010

Nomenclature

v_{Xi}, ω_{Xi}	ith component of the translational/angular velocity of point X, $X \in \{P, CM, 1, 2\}$, where P and CM represent the connection point and the center of mass, respectively
r_i	ith component of the position vector $\vec{r}$ of the connection point
$\mathbf{A}^{XY}$	coordinate transformation matrix from coordinate frame Y to X, X, $Y \in \{B, I, C, 1, 2\}$, where B, I, and C stand for body, inertial, and constraint frames, respectively
$\mathbf{A}_{ij}^{XY}$	ith-row jth-column element of $\mathbf{A}^{XY}$
$\mathbf{I}_{4\times4}$	4×4 identity matrix

The modular approach has many well-established advantages. These include independent creation and reuse of submodels, hierarchical model structure, ease of adjustment of model complexity, ease of model verification, and ease of handling large systems.

A drawback of this approach, however, is that when general-purpose component models are assembled into a system model, the resulting model can be excessively complex for a given application [8]. The component models need to be created for a broad range of applications and, therefore, need to include a lot of detail relevant for that scope. In the case of modular modeling of multibody systems, for example, a generic component model for a rigid body may consider all possible motions in space and include all 6 DoF. However, when component models are assembled into a particular system model, some of that detail may become irrelevant/unimportant in that particular context. Returning to the multibody example, the model for a particular system can be obtained by augmenting component models of rigid bodies with relevant constraints, but then the number of DoF of the system is less than the sum of the number of DoF of the unconstrained components. Therefore, the system model includes an excessive amount of complexity, and it can be desirable to eliminate this excessive complexity. The purpose of this paper is to describe a model simplification technique designed to detect such excessive complexity and remove it without affecting the accuracy of the model.

One motivation for eliminating excessive complexity could be that a simpler model could prove more insightful by showing only what is of relevance to the problem at hand. Also, a simpler model would typically have fewer parameters and states, which reduces the number of parameter values and initial conditions that need to be identified. Note that an initial estimate of parameters is necessary for the full model, but a costly accurate identification can be delayed until the simplified model is obtained. If the model is going to be involved in a control design problem, a simpler model could also simplify the control problem [30]. Finally, a simpler model is generally more computationally efficient, which makes the model more suitable for iterative processes, such as optimization [3], sensitivity analysis [15,20], Monte Carlo simulation [5], system identification [31], etc., or for real-time simulation [17]. It is acknowledged, however, that simplicity does not necessarily always imply computational efficiency [25].

Reducing the complexity in models is an important challenge. Thus, the literature presents many approaches to address this challenge, which can be classified as partitioning, reduction, and simplification. Note that this classification is neither strict, nor formal. The boundaries between these classes are not impenetrable, and methods can belong to more than one category depending on the point of view. Nevertheless, the authors find this classification useful for presentation purposes and adopt it therefore herein.

Partitioning is an approach where weak two-way couplings in a model are replaced by one-way connections to create driving and driven submodels. For example, models can be partitioned into slow and fast dynamics [32], high- and low-frequency oscillation modes [24], or heavily- and lightly-damped dynamics [24]. As another example, Rideout et al. proposed a systematic way to identify and break weak couplings [26]. Reduction, on the other hand, refers to finding a lower-order model that captures only the most essential dynamics, and thus approximates the dominant behavior of the original model. Examples include balanced truncation [23], Hankel-norm approximation [10], and proper orthogonal decomposition [11]. Finally, simplification refers to finding a more succinct realization without compromising accuracy. For example, the well-known ideas of pole-zero cancellation [4], Kalman's minimal realization [13], or explicit elimination of Lagrange multipliers [28,35] can be referred to as simplification techniques. Thus, in this context simplification refers to removing

what is irrelevant, while partitioning and reduction refer to neglecting some existing, but less important phenomena.

To illustrate the difference between partitioning, reduction, and simplification, consider an automobile that is modeled with 6 DoF, but moves only in the pitch plane. Then, assuming that the translational dynamics affect the pitch dynamics, but not vice versa, decoupling the two would be deemed as partitioning. If the pitch dynamics are considered unimportant for the system dynamics and removed from the model completely, that is considered reduction. Finally, recognizing that the vehicle motion is in a plane and, thus, eliminating yaw, roll, and lateral dynamics to end up with a 3 DoF model would be simplification. This paper focuses exclusively on simplification.

As a way of expressing the models in a modular modeling environment, bond graphs prove to be suitable due to their power-based graphical nature and their lending themselves to modularity [14]. As an example, a bond-graph-based modular modeling paradigm has been shown to be suitable for modeling reconfigurable machine tool servo drives [8]. There are other possible representations to create a modular environment, such as block diagrams [18] or Modelica [9], but bond graphs are preferred in this work due their convenience. Nevertheless, it is noteworthy that although the method presented in this paper is particularly amenable to bond graphs, it is applicable to other representations as well.

The literature presents many mathematical tools, such as pole-zero cancellation [4], Kalman's minimal realization [13], or explicit elimination of Lagrange multipliers [28,35] as mentioned above, that can be used for model simplification, but these techniques apply at the equation level. Thus, even though the equations derived from a bond graph could be simplified with such techniques, the bond graph itself would still contain the excessive complexities, obscuring insight, and requiring the repetition of the equation-level simplification each time the equations are derived from the bond graph.

There are some well-established rules for bond-graph-level simplification [14], such as eliminating power-through junctions, merging adjacent junctions of same type, eliminating a null effort (flow) source connected to a 1-junction (0-junction), lumping dependent elements, or some structural equivalencies. However, these rules by themselves are not enough to eliminate a Lagrange multiplier from a bond graph, for example. Hence, they still leave room for more simplification.

A more advanced model simplification procedure in the bond-graph domain is proposed by Rinderle and Subramaniam [27], consisting of four main steps: (1) eliminating null sources, (2) eliminating transformers with zero modulus, (3) eliminating constraining junction structures, and (4) reducing the number of dependent inertia elements through parameter lumping based on the Lagrangian. Although very useful in some cases, this technique is not very effective when the transformers are modulated, or the constraints are enforced through Lagrange multipliers, for example, so that there are no constraining junction structures in the model.

There are also some bond-graph level tools that have been developed for reduction and partitioning purposes, but can serve, to some degree, for simplification purposes as well. In particular, a metric called "activity" for measuring power flow among bond-graph elements has been proposed by Louca et al. and used to create a model reduction algorithm called MORA [21]. If MORA is used to remove zero-activity elements only, it serves as a simplification tool. Nevertheless, a significant amount of complexity may still remain in the model, because MORA concentrates only on the energy elements and not on the junction structure. In addition, the partitioning algorithm proposed by Rideout et al. also leverages the activity concept [26]. Weak coupling points can serve for simplification purposes if the driven partition consists only of an inactive junction structure and elements that can be removed along with the junction structure, but this is not always the case. Thus, this method is also not very effective for simplification.

This paper presents a realization-preserving algorithm to achieve a structural simplification at bond-graph level. To this end, a unified treatment of junction elements is introduced. In particular, the activity metric [21] is extended to junction elements to identify the inactive ones and eliminate them to simplify models. This approach can be considered as a generalization of the well-known idea that 1-junctions with zero flow and 0-junctions with zero effort can be eliminated from a bond graph. Thus, the contribution of this paper is the generalization of the zero-flow and zero-effort metrics into a unified metric, inactivity; a procedure to identify and remove inactive junction structures; and a detailed discussion of the realization-preserving and input-dependent property of this approach.

The remainder of this paper is organized as follows: Section 2 defines inactivity of a junction, lays out the procedure to simplify models when inactive junctions are present, and demonstrates the simplification and its benefits on a simple example. Section 3 discusses the procedure in more detail, highlighting the advantages and disadvantages of its realization-preserving and input-dependent characteristics. Conclusions are drawn in Section 4.

2. Model simplification based on junction inactivity

2.1. Inactivity of junction elements

The activity metric developed by Louca et al. is a measure of power flow in a model for a given input [21]. Activity of an energy element (i.e., generalized inertia (I), capacitance (C) or resistance (R)) is formally defined as

$$A = \int_{t_1}^{t_2} |e \cdot f| \mathrm{d}t \tag{1}$$

where A, e and f are the activity, generalized effort and flow of the element, respectively. Based on the hypothesis that elements with low activity contribute less to the system dynamics, the activity metric is used as the basis for a systematic model reduction technique called "Model Order Reduction Algorithm (MORA)" [21]. In MORA, the activity metric is defined and used for assessing the single port energy elements (I, C and R) only.

If the junction structure is to be considered for simplification, a metric for junctions is needed, as well. In this work a junction element, 1- or 0-junction, is called "inactive," if all the bonds that are connected to the junction element have a negligible activity, i.e.,

$$A_i = \int_{t_1}^{t_2} |e_i f_i| \mathrm{d}t \leqslant \varepsilon, \quad i = 1, 2, \ldots, n \tag{2}$$

where A_i, e_i and f_i are the activity, effort and flow of bond i connected to the junction element, respectively; ε is a small number, which will be referred to hereafter as the inactivity threshold; and n is the number of bonds connected to the junction element. The inactivity threshold is set equal to the numerical zero as determined by the numerical integration tolerance. If an analytical solution is available, the threshold can be set equal to zero.

The hypothesis is that inactive junction structures can be removed from the bond graph without compromising accuracy, thus simplifying the model.

Two points are important to note here: first, the inactive junction concept can be considered as the generalization of the idea that 1-junctions with zero flow and 0-junctions with zero effort can be eliminated from a bond graph without sacrificing the accuracy of the model. This is because a 1-junction (0-junction) will be inactive not only if its flow (effort) is zero, but also if all the efforts (flows) are zero. For example, if a 1-junction represents a characteristic non-zero velocity component along which no force does any work, then that 1-junction is going to be inactive despite the non-zero flow. This point will be clearer in the physical example in Section 2.3.

Second, the elimination of an inactive junction does not necessarily correspond to removing every null-power bond from the model. In fact, the latter may lead to computational problems. To illustrate this, consider a particle of mass m that is constrained to move along an arbitrary path (Fig. 1a). The particle experiences a gravitational force in the $-$y direction, and a viscous friction, b, as it moves along the path. Assume that the modeler has chosen to work with two coordinate frames, the inertial (x–y) and constraint (n–t) frames, for their convenience to express the gravitational and constraint forces, respectively. The bond graph of this scenario is given in Fig. 1b, where the pseudo-flow source (PSf) [8] represents the Lagrange multiplier enforcing the constraint, and modulated transformers take care of the coordinate transformation between the inertial and constraint frames. Specifically, $[\,v_n \quad v_t\,]^{\mathrm{T}} = \mathbf{A}^{CI}[\,v_x \quad v_y\,]^{\mathrm{T}}$, where $\mathbf{A}^{CI}$ is the transforma-

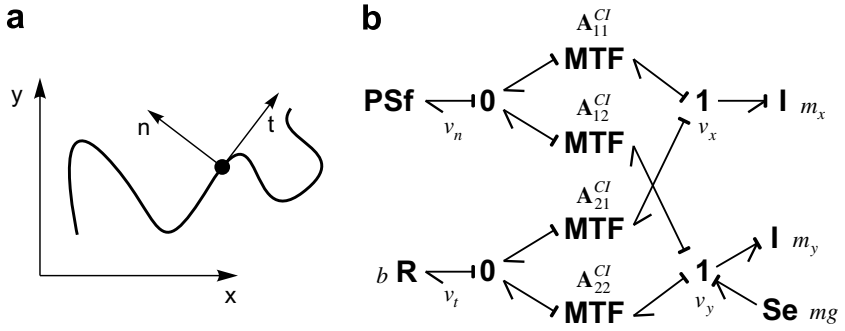

Fig. 1. (a) Particle constrained to an arbitrary path, (b) system bond graph with Lagrange multiplier (PSf).

tion matrix from the inertial frame to the constraint frame. Note that the elements of $\mathbf{A}^{CI}$ can become arbitrarily small, since the path is arbitrary. In other words, even though $\mathbf{A}^{CI}$ will always be invertible, its individual elements may be not. Therefore, all the modulated transformers in Fig. 1b should have a fixed flow-in–flow-out causality to avoid singularities. In that case, the only possible causal assignment is as shown in Fig. 1b. Note that the bond between the PSf and 0-junction elements will have null power due to the zero normal component of the particle velocity. However, it cannot be removed from the bond graph, because that would create a causal conflict due to the fixed causal assignment of the MTF elements. The proposed metric would identify the 0-junction to which the PSf element is connected to as active and would not consider it for simplification, thereby keeping a null-power bond in the model.

2.2. The procedure

Using the inactivity metric defined in the previous subsection for a junction element, a procedure to simplify bond-graph models is proposed below.

Since activity depends on the chosen excitation and time-window, it is critical to choose them properly for the task at hand. Selecting proper excitations and the time-window are considered as prerequisite steps. Section 3 will discuss the importance of proper excitations more in detail.

After the prerequisite steps are completed, the main steps of the simplification procedure can be described as follows:

1. Detection of inactive junction elements: a simulation run is performed and the activity values are recorded. The inactivity threshold ε is selected, typically on the order of magnitude of the numerical integration tolerance, and the inactive junctions are identified using (2).
2. Preserving modulating signals: some junction structures may be inactive, yet important for generating a modulating signal necessary for the rest of the bond graph. To preserve the modulating signal, the inactive bond-graph junction structure that generates the modulating signal should be converted into a block diagram, instead of being removed completely.
3. Elimination of inactive junctions: the remaining inactive junctions are removed from the model, along with the elements and submodel ports that become detached after the removal of the inactive junction elements.

After the last step, if desired, the well-known bond-graph simplification techniques can be applied, such as removal of power through 1- or 0-junctions, removal of unity gain transformers, merging adjacent junctions of the same type, or lumping dependent inertias onto independent ones.

This procedure is given as a flowchart in Fig. 2 and is illustrated with an example in Section 2.3.

2.3. Example: bead on a stick

To illustrate how the proposed algorithm works, consider a bead that can slide smoothly on a stick, which is swung in a vertical plane with constant angular velocity ω. Fig. 3 illustrates the described system and shows

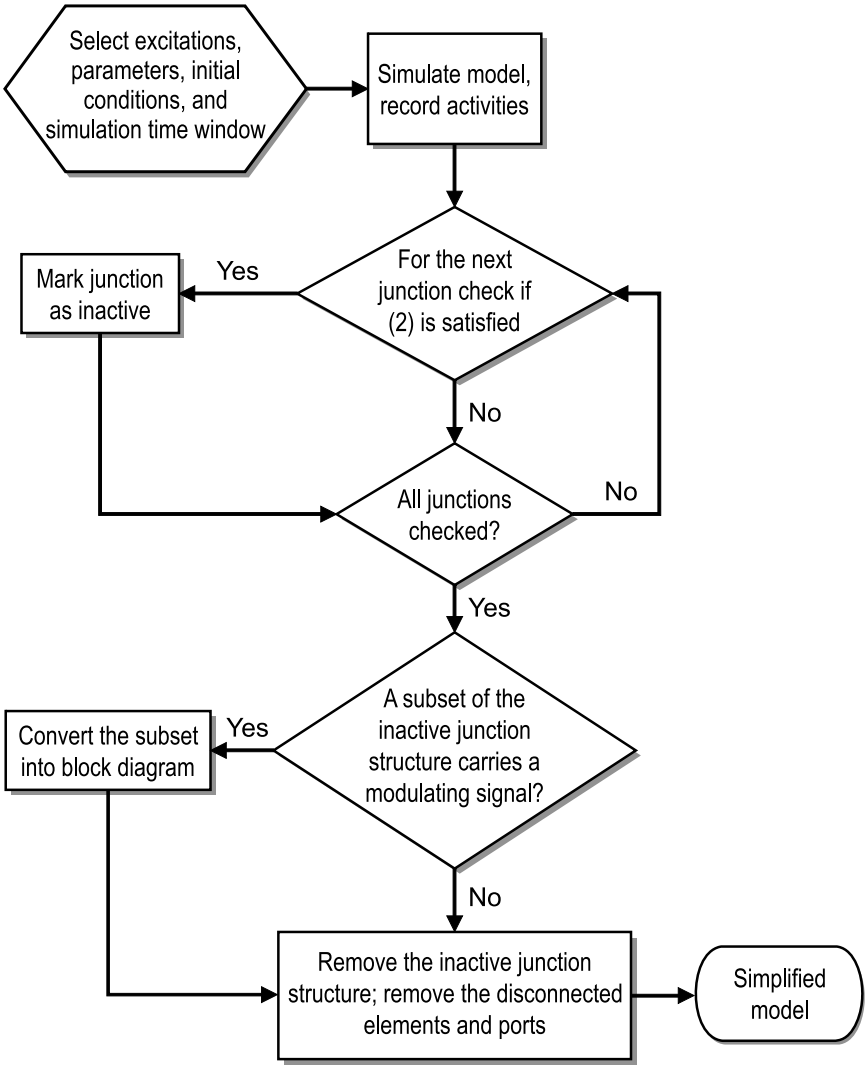

Fig. 2. The flowchart of the simplification process.

its modular bond graph, which is composed of 3D rigid-body and joint models. For the details of the modules please see the Appendix. One may argue that this system is too simple to justify the modular modeling approach, but since the purpose here is to demonstrate and discuss the proposed simplification algorithm, this and all the other example systems in this paper are intentionally kept simple.

Note that the modular model (Fig. 3b), although very easy to create, is quite complex considering the given 1-DoF system: the model has two 6-DoF rigid-body models representing the stick and bead, and two joint models representing the connections between the bead and stick, and the stick and ground. As a result, there are many dimensions, in which the system cannot move, but which are included in the model anyway because of the modular approach adopted in creating it. For the bond-graph this implies that there are many unnecessary elements in the model. Thus, there is an opportunity for simplification created by the modular approach to modeling.

Simplification is then carried out as follows: a simulation run using the 20-sim software [1] is performed to record the activity values, and the original model is simplified based on an inactivity threshold of 10^{-5}, which is also equal to the simulation tolerance. The resulting simplified model is given in Fig. 4. As seen in Fig. 4, the proposed algorithm significantly reduces the complexity of the bead and joint models, and completely removes the stick and ground models. A detailed explanation of the simplification and the physical motivation behind it is given next.

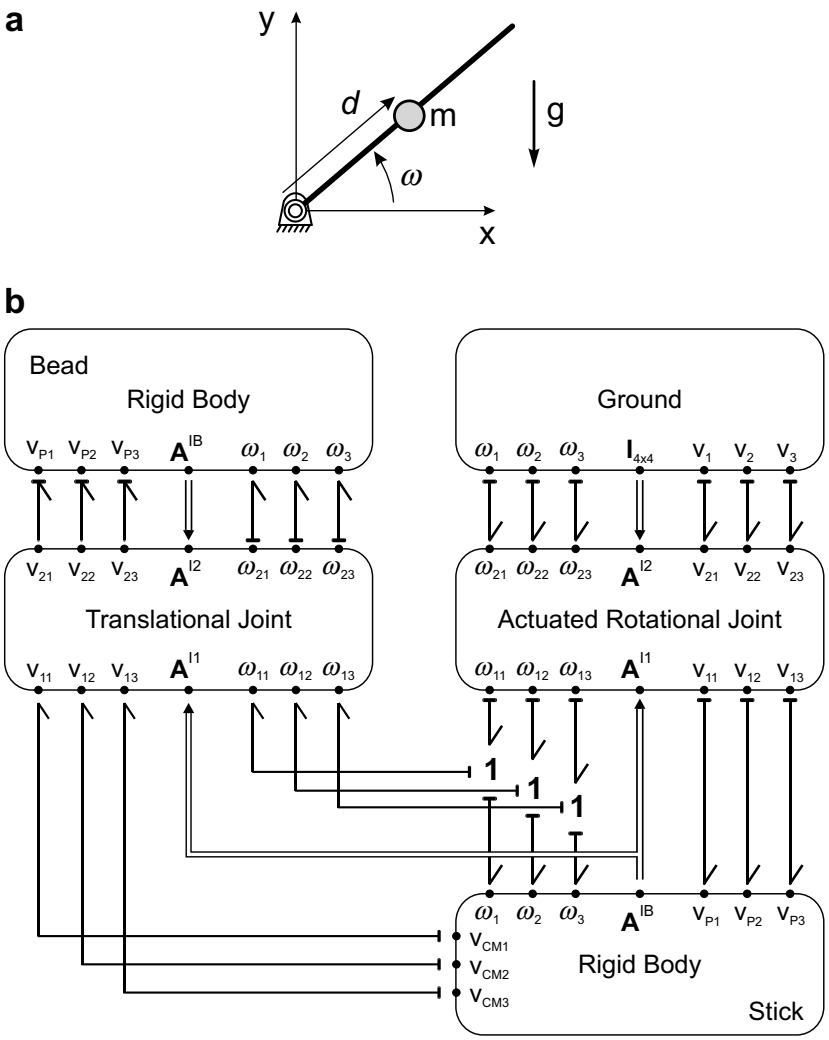

Fig. 3. (a) The bead and stick system and (b) its modular bond-graph model.

First, consider the bead model. Fig. 5 shows the original rigid-body module representing the bead and its simplification in detail. In Fig. 5 the inactive junction-structure to be removed is shown in grey, along with the elements to be removed based on the implication of the inactive junction-structure. The junction structure that is inactive, but important for the generation of a modulating signal, is shown with outlined characters and the corresponding bonds are denoted by dash-dotted lines.

The phenomena that are removed due to inactivity or preserved despite inactivity, and the physical motivations behind those removals and preservations are as follows: Rotational dynamics are eliminated completely, because the stick and, therefore, the bead connected to it are rotating with a constant angular velocity. The cross-product $\vec{\omega} \times \vec{r}$, i.e., the velocity of the connection point of the bead due to rotation, is removed, because the connection point coincides with the center of mass of the bead, i.e., $\vec{r} = 0$. The translational dynamics in the z-direction is removed, because the system is planar. Due to the same reason, there is also some simplification in the coordinate transformation from bead frame to the inertial frame. Finally, the preserved inactive structure is generating the signal $\dot{d}$, i.e., the velocity of the bead relative to the stick. The inactivity of this particular structure is due to the smoothness of the motion. Nevertheless, the signal $\dot{d}$ is important, as its integral, d, i.e., the position of the bead relative to stick, is one of the states of the system. The associated force variable, however, is not necessary, because there is no force that acts along d. Hence, this particular junction structure is converted into a block diagram in Fig. 4, which is the signal path from the 1-junctions in the bead model to the integrator in the translational-joint model. This eliminates the corresponding force variable from the model.

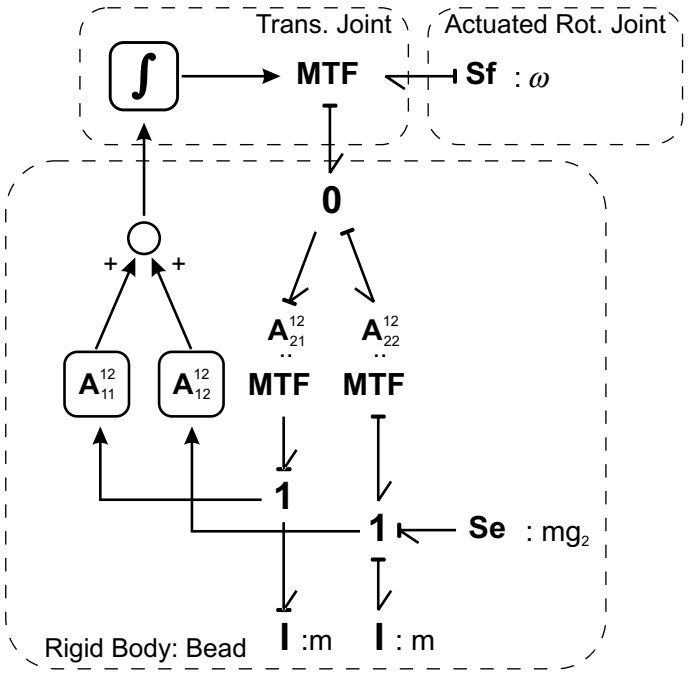

Fig. 4. The simplified bead and stick model.

Second, consider the stick and ground models. Note how the algorithm recognizes the fact that the dynamics of the stick are irrelevant due to the constant angular velocity, and eliminates the rigid-body model representing the stick completely from the model. This makes the ground model redundant as well.

Finally, with 4 DoF removed from the bead model, and the stick and ground models eliminated completely, the constraints in the joint models become unnecessary and are therefore removed, which significantly simplifies the joint models as well.

Note that all these physical observations are incorporated into the model automatically by the simplification algorithm. Therefore, the simplification provides additional insight into the system, making the model more conceptually efficient. Furthermore, accuracy is preserved (within the numerical tolerance) during the simplification as seen in Fig. 6, where the relative position of the bead is chosen as the output. These results have been obtained using the following parameter values and initial conditions: $m = 1$ kg, $\omega = 5$ rad/s, $g = 9.81$ m/s^2, $\theta(0) = 0$, $d(0) = 0.3$ m.

The simplification of the bond graph leads not only to conceptual efficiency, but also to computational efficiency. With a reduced number of simulation equations, as evident from Table 1, the simplified model takes nearly 80% less time than the original model for 1000 iterations of simulation while predicting the same behavior as the original model. The rationale behind repeating the simulation 1000 times is to reduce the stochastic variations in simulation time due to the other processes concurrently running on the computer. The quantities in Table 1 are obtained with the 20-sim software [1].

3. Discussion of characteristics of inactive-junction-based simplification

The proposed algorithm has two important characteristics: being realization-preserving and input-dependent. It is realization-preserving, because the simplified bond graph is in essence a subset of the initial bond graph. This property is important to preserve the physical meaning of the original model, but has further implications as will be discussed further in this section. The proposed algorithm is input-dependent, because the power flow in the model, and therefore the activity analysis, depends on the inputs. Here the term "input" collectively refers to excitation, parameters, and initial conditions. This can be considered as both an advantage and disadvantage as will be demonstrated later in this section.

This section illustrates the importance of these two characteristics using examples. A more formal discussion of these characteristics is omitted for brevity, and is subject of other papers [6,7] and future work.

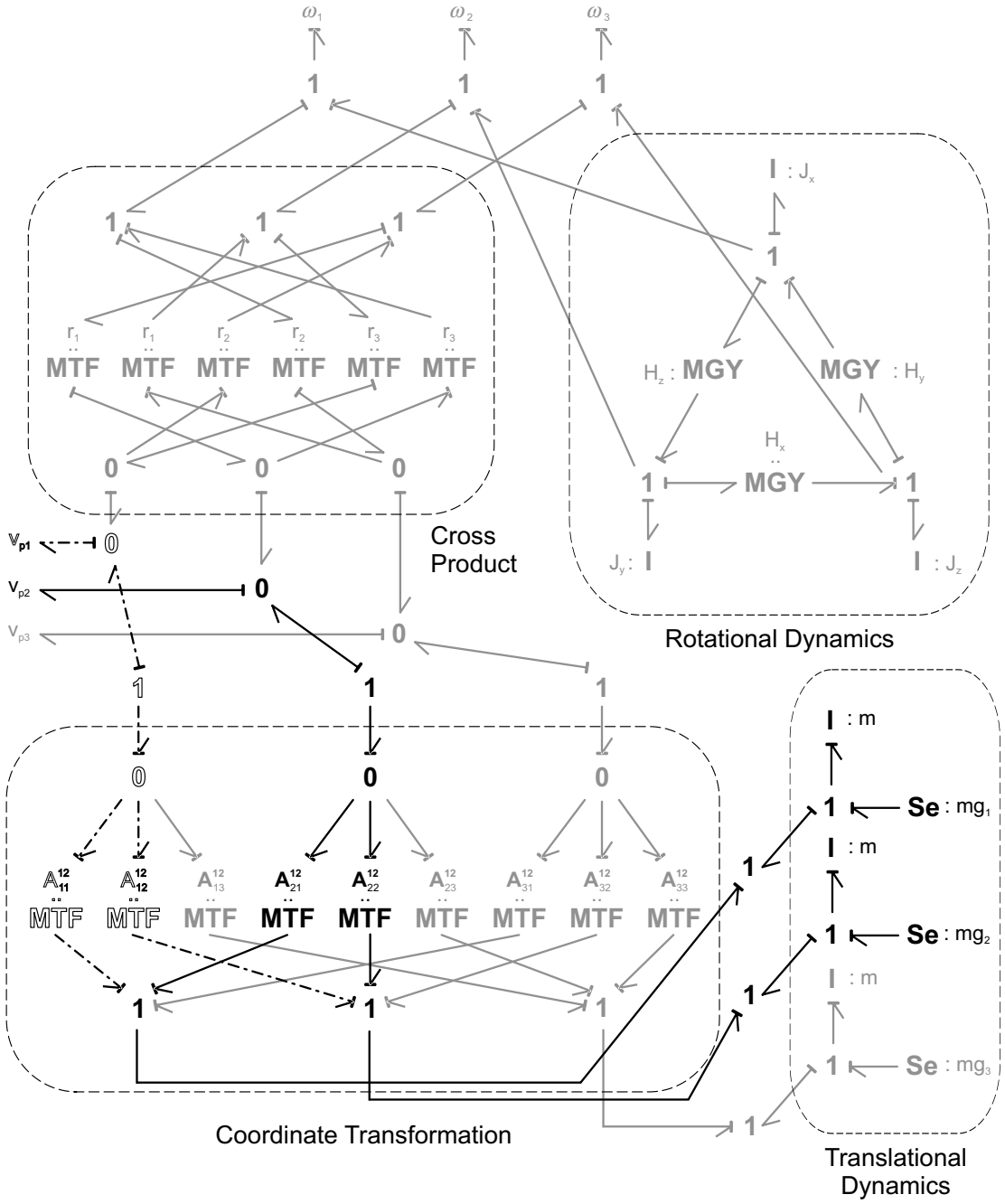

Fig. 5. Bead model with inactive junction structure and its implications marked.

3.1. Preservation of realization

The proposed algorithm simplifies a given bond-graph model by detecting and eliminating the elements that do not contribute to the system dynamics. As such, the simplified model is a subset of the initial bond-graph. Therefore, the realization of the original model is preserved. This realization-preserving property is important to preserve the physical meaning of the original model. As a result of this property, the realization of the original model significantly affects the outcome of the algorithm, because some realizations can be more conducive to simplification than others. This is demonstrated by two mechanical system examples below. In particular, the first example highlights the effect of the orientation of the coordinate frames, and the second one highlights the effect of using absolute versus relative coordinates.

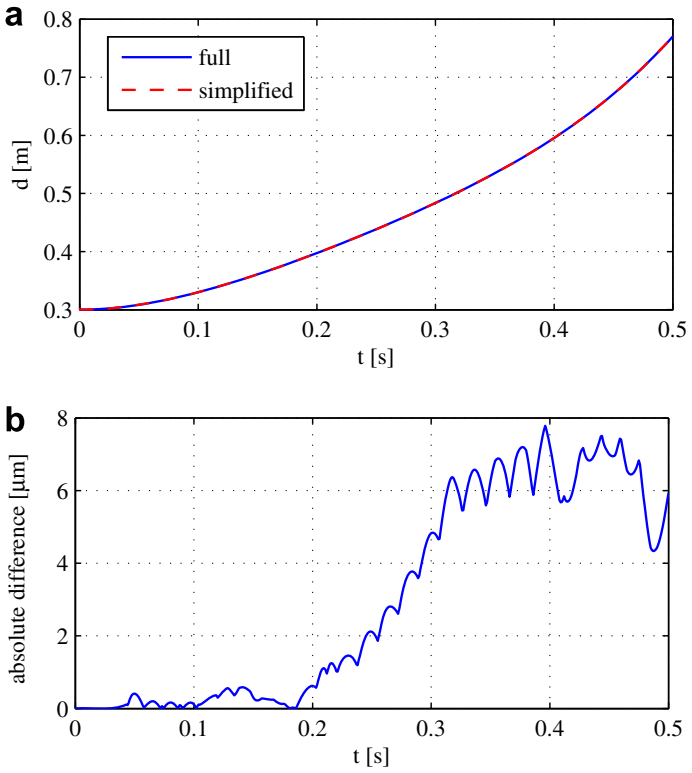

Fig. 6. (a) Outputs of the full and simplified models and (b) the absolute difference between the two outputs.

Table 1
Increase in efficiency due to simplification

	Original model	Simplified model
Number of equations	557	37
Number of variables	698	53
Number of independent states	20	4
Number of dependent states	9	1
Number of constraints	2	0
CPU time for 1000 runs	17.33 s	3.64 s

Example 1: orientation of the coordinate frames in multibody systems

In the mechanical domain the orientation of the coordinate systems is an important factor in simplification, because some coordinate systems can be more conducive to simplification than others. If the coordinate systems of the original model are not properly aligned with the motions and constraints, then performing a coordinate transformation first can yield more dramatic simplification results.

As an example, consider the simple pendulum shown in Fig. 7 with two possible choices of coordinates for this system, where the first one is rather arbitrary and the second one is aligned with the rotation axis. The original modular model for both choices of coordinates is given in Fig. 8. Due to poor alignment, the arbitrary frame Fig. 7a does not yield a significant simplification, as shown in Fig. 9.

The structure of the bond-graph model in Fig. 9 can be briefly explained as follows. The Rigid Body block represents the pendulum as a rigid body. Within this block, the Rotational Dynamics block is the implementation of Euler's equations for a rotating body. Its implementation is what is known as the Eulerian Junction Structure in bond-graph literature [14], and can be seen in the Appendix in detail (Fig. 22). Similarly, the Translational Dynamics block is the implementation of Newton's equations. The translational dynamics are expressed in the inertial frame, whereas the rotational dynamics are expressed in a coordinate frame fixed

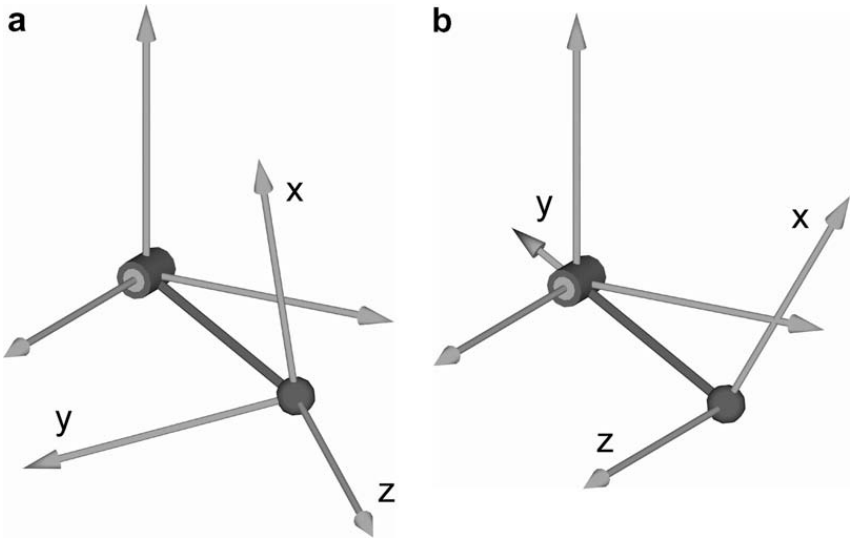

Fig. 7. Two alternatives for the pendulum coordinate frame: (a) arbitrary and (b) aligned with motion.

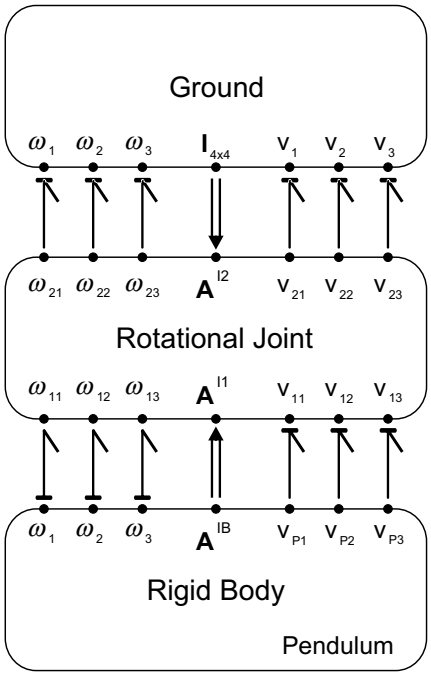

Fig. 8. The modular pendulum model.

to the pendulum. The Coordinate Transformation block takes care of the transformation between the two coordinate frames using the output of the Quaternion block as the modulating signal for the transformers. Given the angular velocity vector, the Quaternion block calculates the orientation of the pendulum in the form of the transformation matrix from the pendulum frame into the inertial frame. Quaternions are preferred in this modular environment to avoid singularities. The Cross Product block calculates the velocity of the hinge point due to the rotation of the pendulum. When added together with the velocity of the center of mass using the 0-junction, this gives the absolute velocity of the hinge point, which happens to be zero in this case. Furthermore, the constraint that the pendulum swings in a plane is imposed through the Rotational Joint block, which involves another coordinate transformation into the constraint frame, and two PSf elements implementing Lagrange multipliers and restraining the rotation about the two axis perpendicular to the axis of rotation.

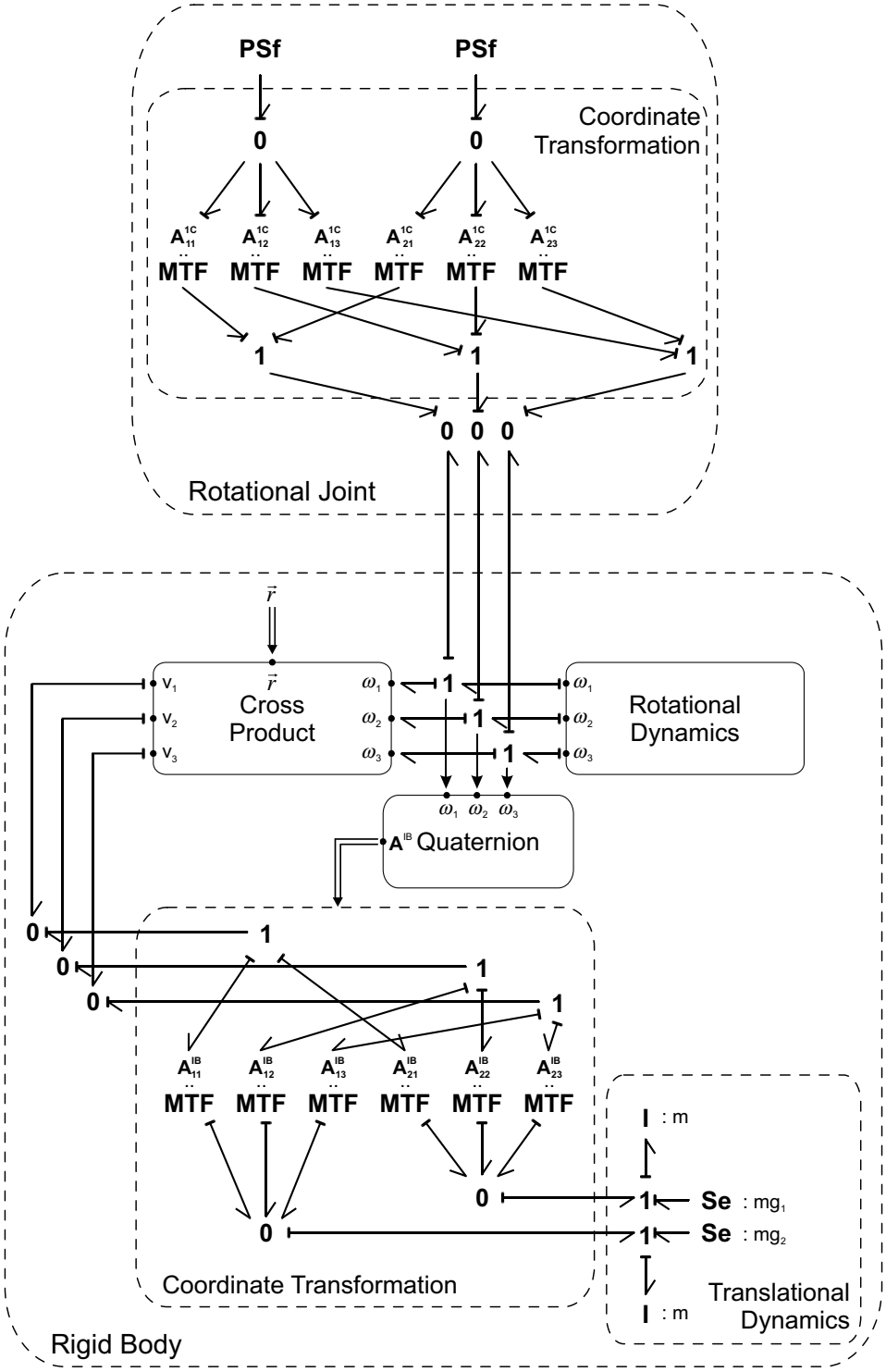

Fig. 9. The simplified pendulum model for the first coordinate frame.

If, however, the pendulum coordinate frame is oriented as in Fig. 7b, the junction structure corresponding to the rotation of the pendulum about the x- and y-axes becomes inactive, because the angular velocity of the pendulum is completely described by its z-component when resolved in this particular coordinate frame. In this case one obtains the simple model shown in Fig. 10.

An automated way to switch from the first frame to the second one could be desirable as a supplement to the proposed algorithm. Such a reorientation of the coordinate frame is a special case of the problem of selecting a suitable set of independent coordinates to solve a set of differential algebraic equations (DAE), which has been

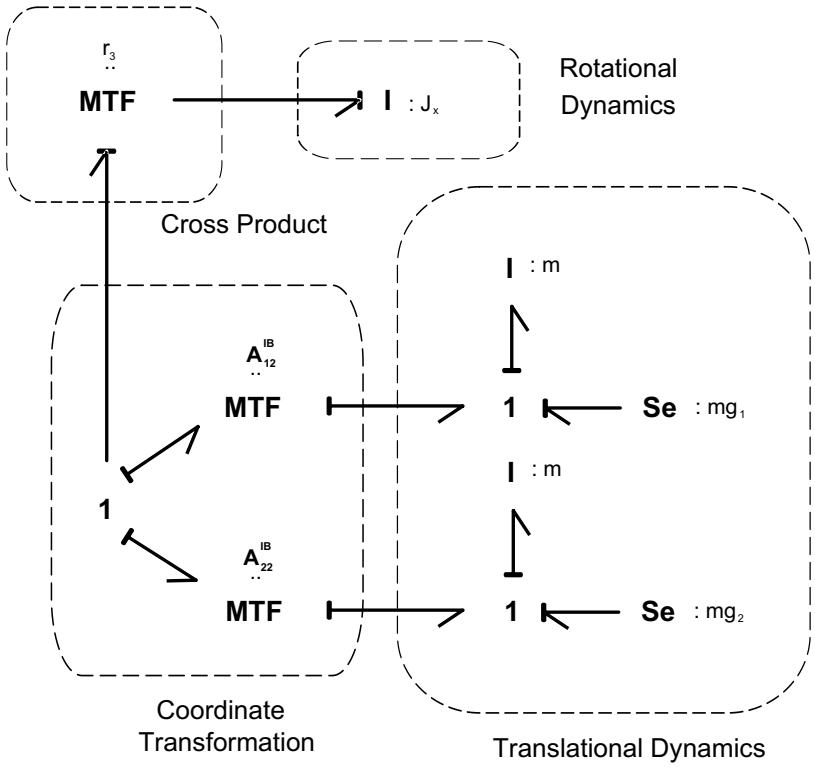

Fig. 10. The simplified pendulum model for the second coordinate frame.

studied widely in literature under the keywords "coordinate partitioning" and "tangent/null space methods". In particular, these keywords refer to the approach, in which, given a set of coordinates and constraints, a set of independent coordinates are sought to avoid integrating dependent coordinates and causing constraint violations. Various approaches exist to automatically select the independent coordinates, including selecting a subset of the original coordinates using the LU decomposition [34], or creating independent linear combinations of the original coordinates by using the zero eigenvalues theorem [33], QR decomposition [16], singular value decomposition [22,29], or Gram–Schmidt orthogonalization [2,12,19]. However, these methods work with algebraic constraint equations, and typically combine all original coordinates into new ones, resulting in a change in realization more dramatic than just reorienting a coordinate frame. Addressing these issues, however, is beyond the scope of this paper. Nevertheless, a procedure is proposed in [6,7] that employs the Karhunen-Loève expansion to detect the existence of and to find the transformation into a better aligned coordinate frame.

Example 2: choosing between absolute and relative coordinates

Reorienting the coordinate frames may not by itself furnish the realization most conducive to simplification. Another realization-related question to consider is whether to use absolute or relative coordinates. This example shows how the two different choices yield different simplification results.

Consider a ball, which can move smoothly on the oblique surface of a wedge, as shown in Fig. 11a. A horizontal force acts on the wedge, so that the system starts moving from rest. Fig. 11b shows the corresponding modular bond graph.

The bond graph as given in Fig. 11b uses absolute coordinates to express the dynamics, i.e., all the inertia elements are connected to 1-junctions that represent absolute velocities, whether resolved in the inertial frame or a body coordinate frame. In this case, no matter how the coordinate frames are oriented, the simplification procedure will always end up with two translational degrees of freedom for the ball along with a constraint, because the ball moves in two dimensions with respect to the ground.

It is easy to see, however, that if the generalized coordinates are chosen as X and d (Fig. 11a), it is possible to express the system dynamics in only two second-order differential equations with no constraints. To get the

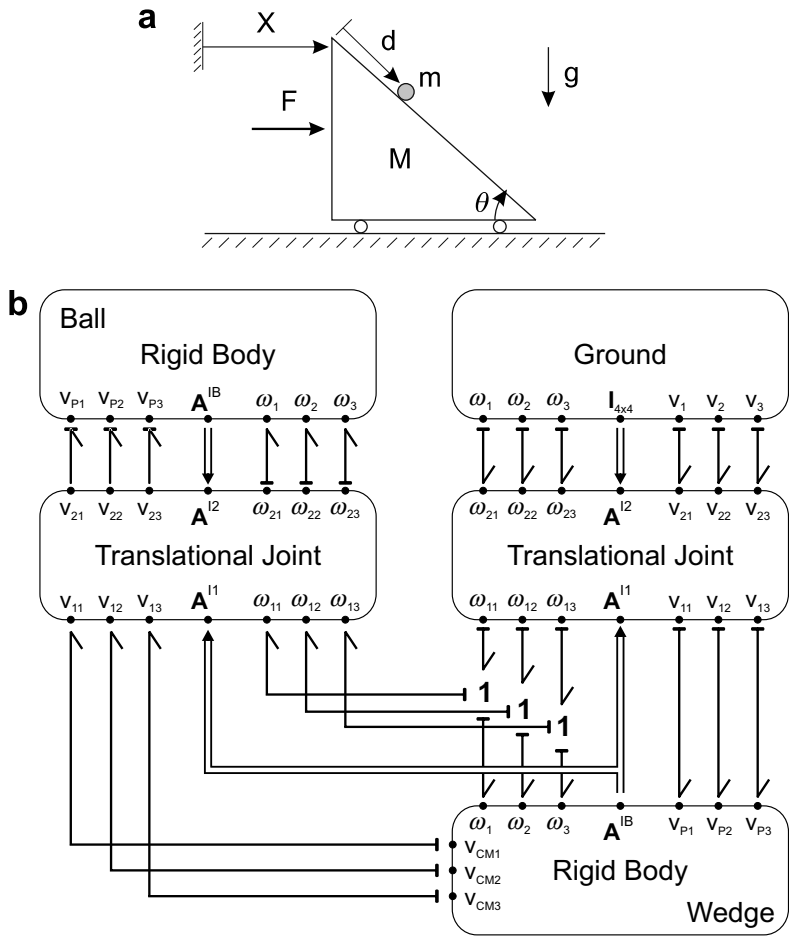

Fig. 11. (a) The ball and wedge system and (b) its modular bond-graph model.

same result with the simplification procedure, the same coordinates must be used in the bond graph, i.e., the translational dynamics of the ball must be expressed using relative coordinates, since the second generalized coordinate is a relative quantity. In bond-graph terms, the inertia elements that are currently connected to the 1-junctions representing the ball's absolute velocity in the inertial coordinate frame must be transferred over to the 1-junctions that stand for the ball's relative velocity with respect to the wedge and reside in the translational joint block. This transformation is straightforward; hence, its details are omitted for brevity.

When the aforementioned modification is made to the system bond graph and the simplification procedure is applied, one obtains the result shown in Fig. 12, where there are only two degrees of freedom and no constraints.

This example shows that if the coordinates used to express the dynamics are independent, as is the case when relative coordinates are used, the proposed algorithm can eliminate all the constraints from the model.

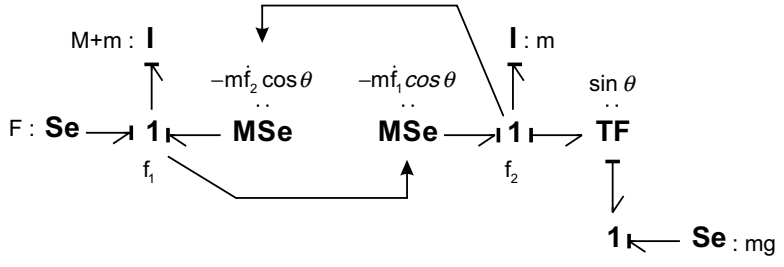

Fig. 12. Simplified ball and wedge model when relative coordinates are used.

Although not essential for the proposed algorithm to work, an automated way to convert a realization in absolute coordinates into a realization in relative coordinates could be another beneficial supplemental tool for the proposed algorithm.

3.2. Dependence on inputs

The proposed algorithm is input-dependent, where the term "input" is used in a more general sense, referring to excitations, parameters, and initial conditions altogether. This subsection demonstrates that this explicit dependence can be both an advantage and disadvantage. It is a disadvantage in the sense that if the inputs are not chosen properly, the algorithm may yield a model that is simplified for a different scenario than the intended one. In this context excitations are considered proper if they correctly represent the scenario of interest, e.g., if they cause non-zero constraint forces for all constraints. This is demonstrated in Example 1. Nevertheless, it is also advantageous to have a tool that can take into account different scenarios and simplify the model accordingly. Example 2 illustrates this.

Example 1: importance of proper excitation

Consider the pendulum example given in Example 1 of Section 3.1 with the arbitrary choice of pendulum coordinate frame (Fig. 7a). The choice of the excitation becomes very critical in this case: if there is no excitation force in the model besides gravity, the simplification algorithm simplifies the original model (Fig. 8) as shown in Fig. 13. Note that there are three independent rotational DoF in the rigid-body model and no constraints,

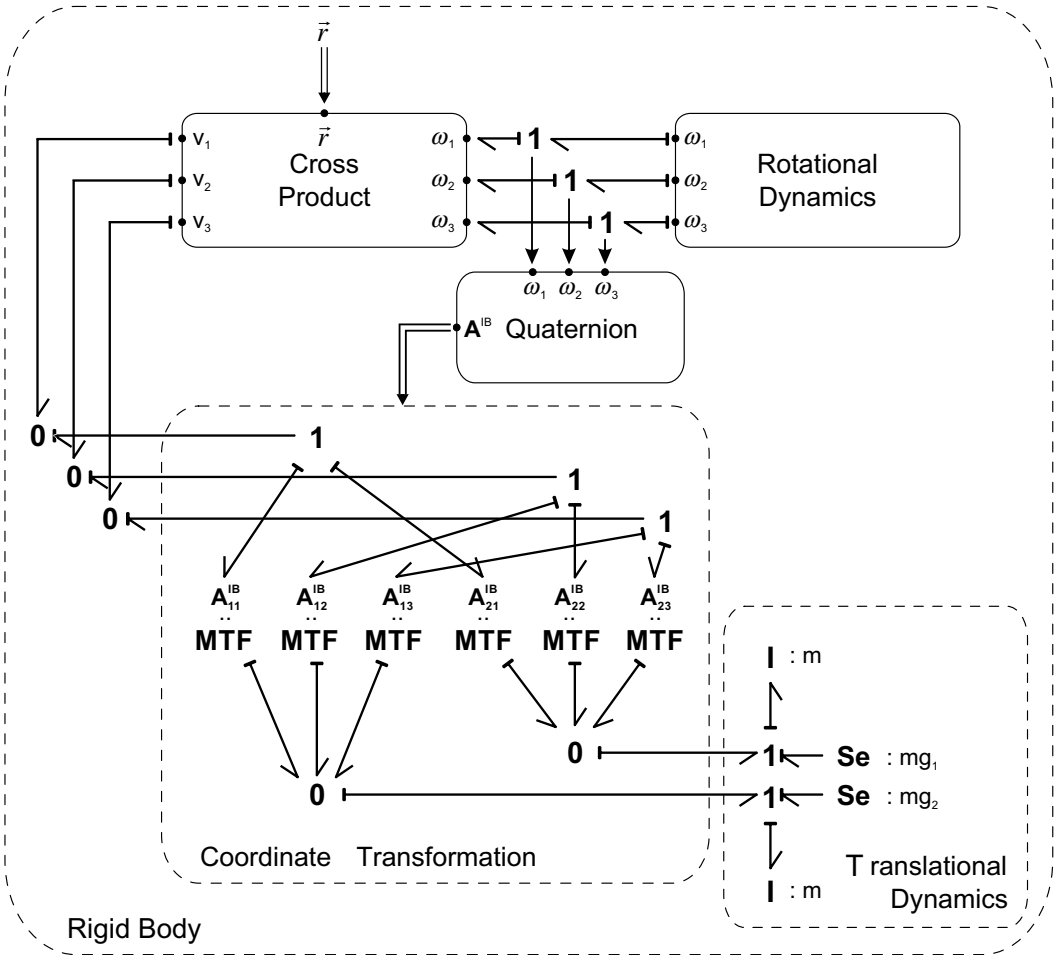

Fig. 13. The simplified pendulum model obtained with a poor excitation.

which means that the number of DoF of the simplified model is three, which is two more than the number of DoF of the original model.

The reason why the number of DoF changes after simplification is the improper excitation. Since there is no force that tries to move the pendulum away from its swing plane, the constraint forces get eliminated from the original model due to the inactivity of junctions to which the constraint forces are connected. As long as the initial conditions satisfy the constraints there is no problem with the simplified model in terms of predicting the pendulum behavior. The problem is, however, that there is no structural mechanism left in the model to enforce the conformity of the initial conditions with the constraints.

The solution to this problem is to augment the original model with proper excitations. If we add to the original model forces that try to move the pendulum out of its swing plane, the simplified rotational joint model becomes as shown in Fig. 9. Although the rigid-body model still has three independent inertia elements, the system has only one DoF due to the two constraint elements kept in the rotational joint. Therefore, the model can prevent inconsistent initial conditions, and also accommodate forces that do not lie in the swing plane.

A more general conclusion that can be drawn from this example is that a set of representative excitations, instead of a single one, may be necessary to describe the scenario of interest. In that case, the simplification has to be repeated for all excitations in the set. The simplified model for that particular scenario is then the union of the simplified models. In this paper, the selection of the set of excitations descriptive enough for a particular scenario is left to the modeler. Development of a tool to aid in this task is beyond the scope of this paper, but appropriate for future research.

Example 2: advantage of input-dependence

The input-dependent characteristic of the proposed algorithm allows for the algorithm to take different scenarios into account. As a result, the algorithm can exploit the different scenarios and yield different simplified models. This example shows that the algorithm can simplify a model that is already simple enough for many different scenarios even further, when a particular input is considered.

Consider a system where two mass-spring systems are concatenated and the mass on the bottom is excited with a harmonic force. This physical system and its bond-graph model are given in Fig. 14.

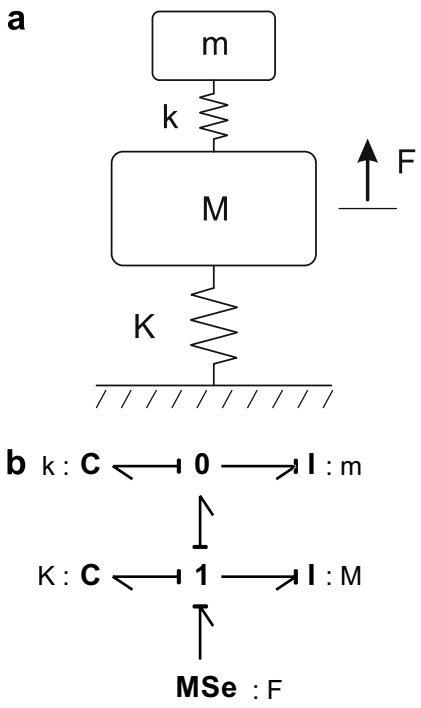

Fig. 14. (a) Two-mass-spring system and (b) its bond-graph model.

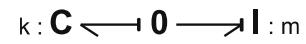

Fig. 15. Simplified two-mass-spring system for the perfectly-tuned vibration absorber scenario.

The bond graph of the system is certainly simple enough, but, as an illustration, consider what happens for a special parameter and initial condition set that results in the top mass-spring system acting like a perfectly-tuned passive vibration absorber. In that case, the bottom mass-spring system would not move, because the force generated by the vibration absorber would be symmetric to the applied force. If one applies the simplification procedure for this particular excitation, parameter and initial condition set, one obtains a simpler model, in which the inert mass-spring system is removed along with the excitation force Fig. 15. In other words, two idle states and one excitation are removed from the model, and the system motion is then only due to the initial conditions of the remaining two states.

The general conclusion drawn for Example 1 also applies here, i.e., the simplification may have to be repeated for different parameters to find a simplified model suitable for the scenario considered. For example, in the case of parameter uncertainty it may be a good practice to perturb the parameters to check if the simplification results are not for a very special case such as a perfectly-tuned vibration absorber. Similarly, if a possible set/range of parameters exists, the simplification should be repeated for a representative set of parameters and the union of the results should be taken as the simplified model. How the representative parameters should be selected is an important issue beyond the scope of this paper, but appropriate for future research.

4. Summary and conclusion

Inactivity of a junction element is defined, and a bond-graph-level structural simplification procedure is proposed that is based on junction inactivity. This approach can be considered as a generalization of simplifying a bond graph by removing zero-flow 1-junctions and zero-effort 0-junctions. The suggested use of the activity metric allows for a unified treatment of junction elements and leads to a unified structural simplification procedure that is easy to implement at bond-graph level. The procedure is illustrated with examples, and its realization-preserving and input-dependent characteristics are highlighted. Even though the examples in this paper were selected from the mechanical domain, the proposed algorithm is applicable to any energetic domain due to the energy-based metric utilized. In that sense, the algorithm is domain independent.

Although the procedure is presented in a bond-graph framework, it is not exclusively for bond graphs and can be implemented in other modeling environments as well, provided that the power variables required for the activity calculation are available. Nevertheless, bond graphs *readily* provide these power variables and *explicitly* show the junction elements of interest, thereby *greatly facilitating* the presentation, implementation, and discussion of the algorithm. Therefore, they are the preferred representation in this work.

The preservation of realization property of the proposed algorithm helps preserve the physical meaning of the model during simplification, but also raises the question about which realizations to use to achieve best results. To some extent, this question has been answered in [6,7], and further investigation of this topic is left as a future work.

The input-dependent characteristic of the proposed algorithm, on the other hand, has been shown to be both a pitfall and an advantage. It is important to select the inputs (excitations, parameters, initial conditions) carefully, as they define the scenario that is being subject to simplification. Ill-defined scenarios may lead to overly simplified results, which is a potential pitfall. However, when the inputs are selected properly, the input-dependence of the algorithm can create different simplified models for different scenarios. From this point of view, this characteristic is considered as an advantage.

If more than one set of inputs is necessary to define the scenario of interest, e.g., in the case of stochastic inputs, parameter uncertainty, a range of possible parameter values, etc., then repetition of the simplification algorithm for all inputs is necessary. In that case, the union of the simplification results should be taken as the simplified model for that particular scenario. Proper selection of inputs is left to the modeler in this paper. However, it may be a challenging task, and a tool to address this challenge could be valuable. This is another potential area for future research.

The results obtained so far with the proposed procedure encourage the integration of this tool with the modular modeling approach. Instead of creating a simple model by hand, which is error-prone and time-consuming, system models can be assembled quickly in a modular way, and then simplified automatically to increase insight and efficiency. This provides an alternative way to obtaining simple models, not just at the equation level, but also at the bond-graph level, while preserving all the benefits of modularity. This in turn can be very valuable from both a conceptual and computational point of view by allowing for rapid development of mathematical models tailored for specific scenarios of interest.

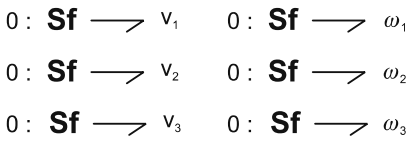

Fig. 16. The ground model.

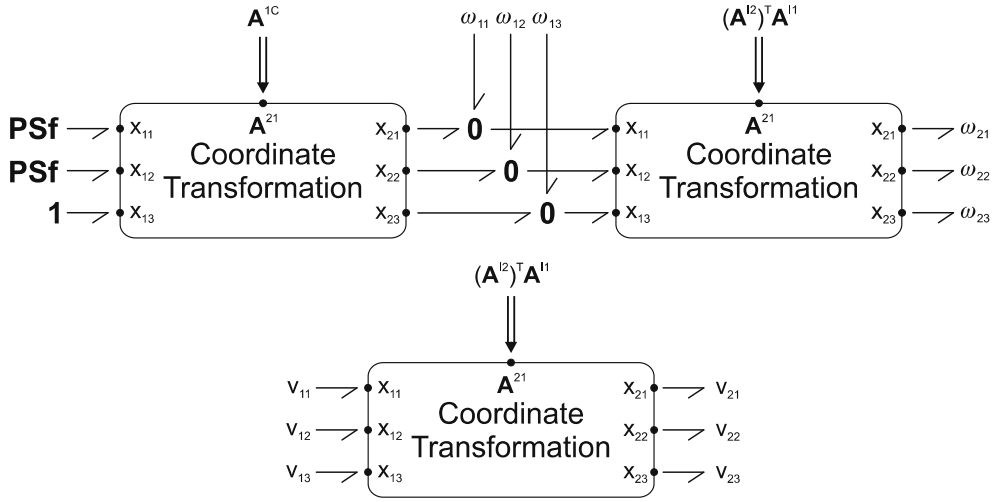

Fig. 17. The rotational joint model.

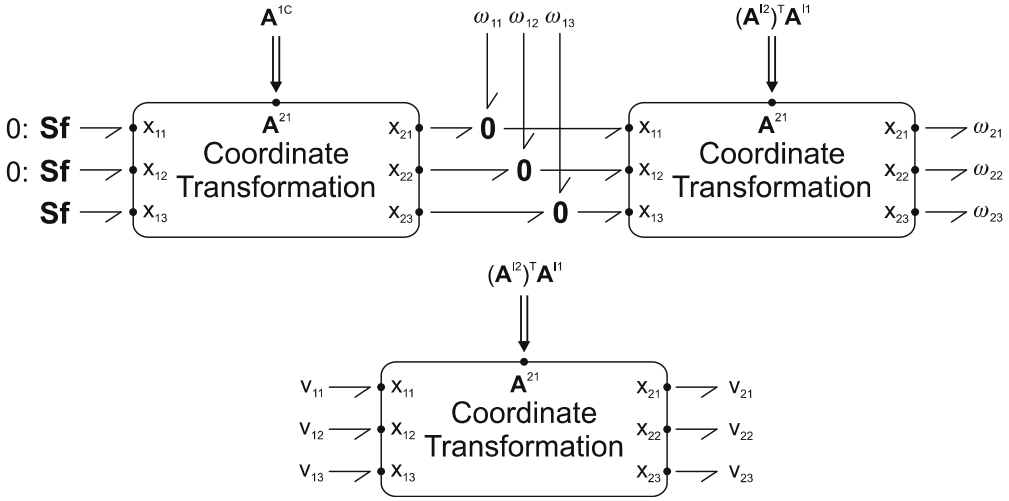

Fig. 18. The actuated rotational joint model.

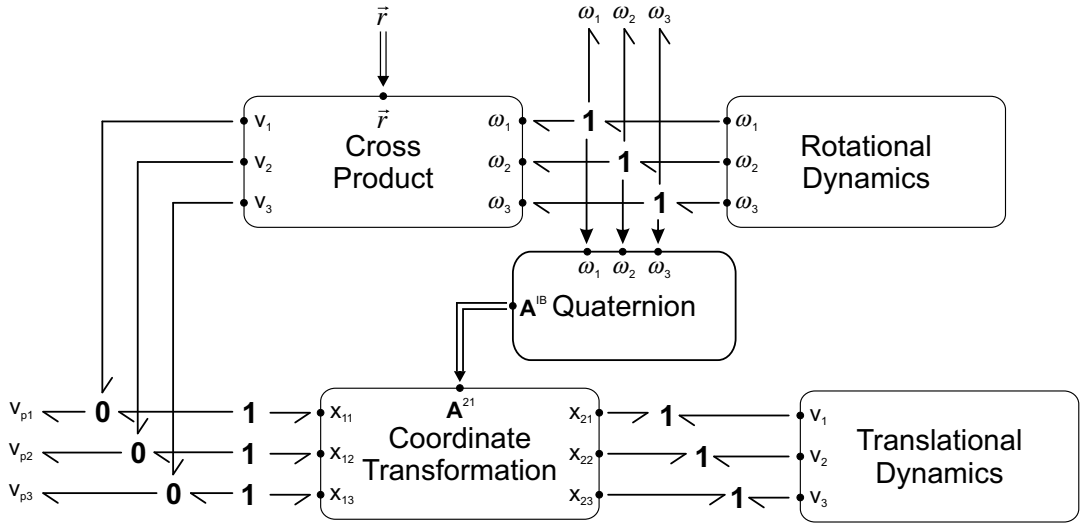

Fig. 19. The rigid body model.

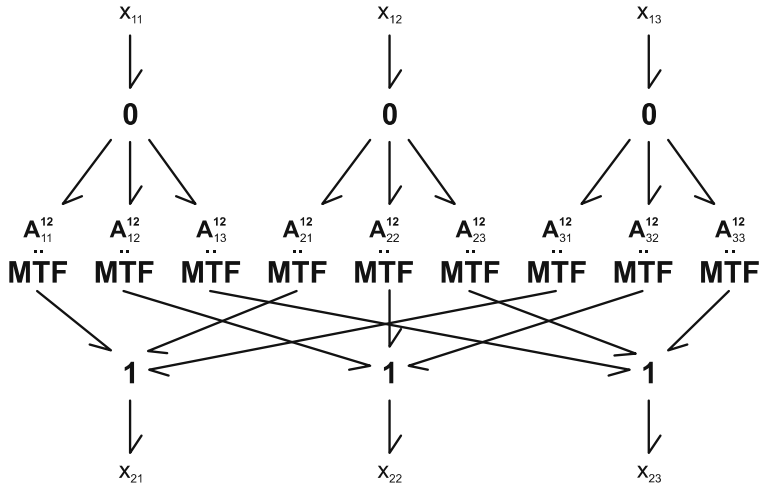

Fig. 20. The coordinate transformation model for effort-in–effort-out causality.

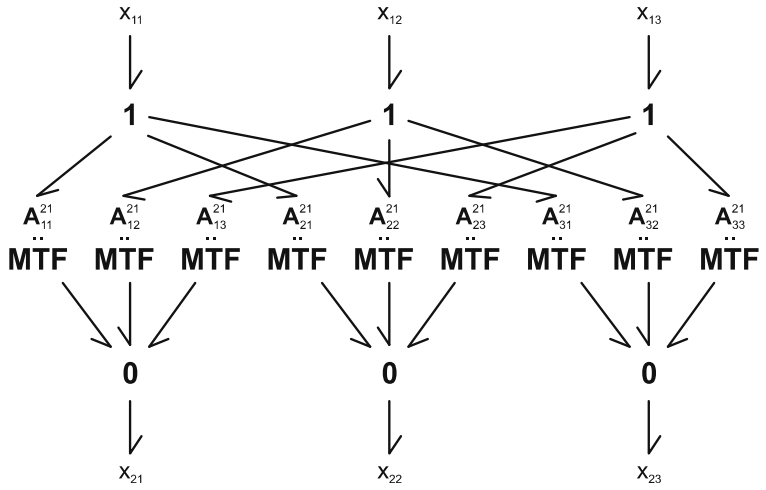

Fig. 21. The coordinate transformation model for flow-in–flow-out causality.

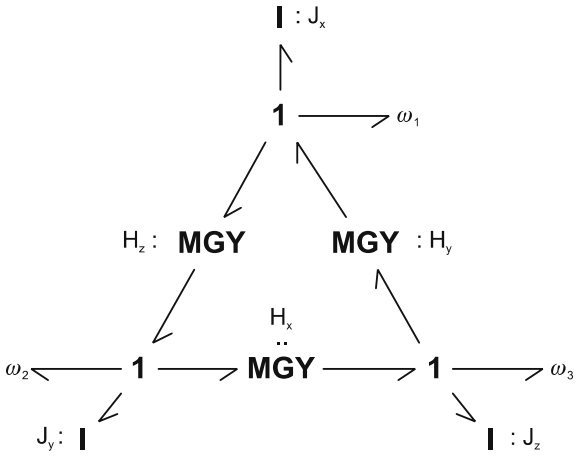

Fig. 22. The rotational dynamics model.

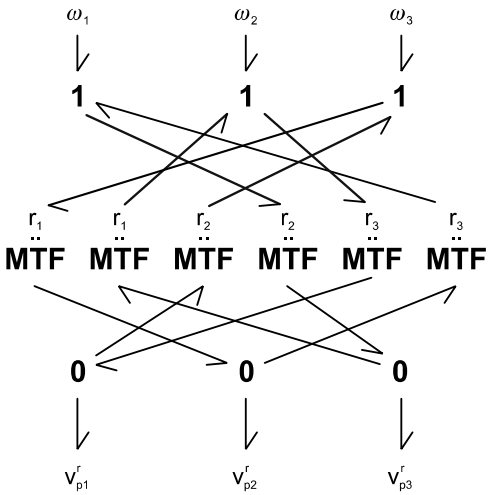

Fig. 23. The cross product model.

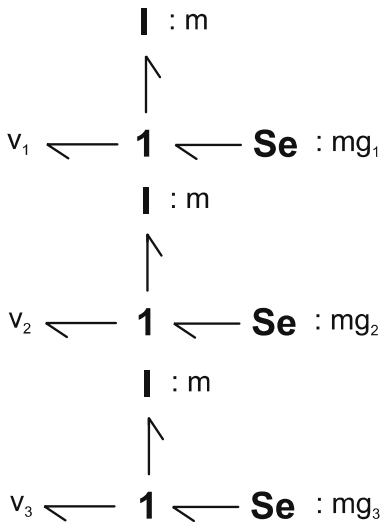

Fig. 24. The translational dynamics model.

Acknowledgment

This work was supported by the Engineering Research Center for Reconfigurable Manufacturing Systems of the National Science Foundation under Award Number EEC 9529125.

Appendix A. The modules

This Appendix shows all the bond-graph modules used in this paper in detail (see Figs. 16–24).

References

[1] 20-sim, v3.6, Controllab Products B.V., Enschede, The Netherlands, 2005.
[2] O.P. Agrawal, S. Saigal, Dynamic analysis of multi-body systems using tangent coordinates, Computers and Structures 31 (1989) 349–355.
[3] D. Assanis, W. Bryzik, N. Chalhoub, Z. Filipi, N. Henein, D. Jung, X. Liu, L. Louca, J. Moskwa, S. Munns, J. Overholt, P. Papalambros, S. Riley, Z. Rubin, P. Sendur, J. Stein, G. Zhang, Integration and use of diesel engine, driveline and vehicle dynamics models for heavy duty truck simulation, in: Proceedings of 1999 SAE Congress, SAE, 1999.
[4] J.S. Bay, Fundamentals of Linear State Space Systems, McGraw-Hill, 1998.
[5] R. Bladh, M.P. Castanier, C. Pierre, Reduced order modeling and vibration analysis of mistuned bladed disk assemblies with shrouds, Journal of Engineering for Gas Turbines and Power 121 (1999) 515–522.
[6] T. Ersal, Realization-Preserving Simplification and Reduction of Dynamic System Models at the Graph Level, Ph.D. Dissertation, University of Michigan, Ann Arbor, 2007.
[7] T. Ersal, H.K. Fathy, J.L. Stein, Orienting body coordinate frames using Karhunen-Loève expansion for more effective structural simplification, Simulation Modelling Practice and Theory, in press, doi:10.1016/j.simpat.2007.08.002.
[8] T. Ersal, J.L. Stein, L.S. Louca, A bond graph based modular modeling approach towards an automated modeling environment for reconfigurable machine tools, in: Proceedings of IMAACA'04, Bond Graph Techniques for Modeling Dynamic Systems, SCS, 2004.
[9] F. Felgner, R. Merz, L. Litz, Modular modelling of thermal building behaviour using Modelica, Mathematical and Computer Modelling of Dynamical Systems 12 (2006) 35–49.
[10] K. Glover, All optimal Hankel-norm approximations of linear multivariable systems and their I-infinity error bounds, International Journal of Control 39 (1984) 1115–1193.
[11] P. Holmes, J.L. Lumley, G. Berkooz, Turbulence, Coherent Structures, Dynamical Systems and Symmetry, Cambridge University Press, Cambridge, 1996.
[12] S.K. Ider, F.M.L. Amirouche, Coordinate reduction in the dynamics of constrained multibody systems-a new approach, Journal of Applied Mechanics 55 (1988) 899–904.
[13] R.E. Kalman, Mathematical description of linear dynamical systems, SIAM, Journal on Control Series A 1 (1963) 152–191.
[14] D. Karnopp, R.C. Rosenberg, System Dynamics: A Unified Approach, John Wiley & Sons, 1975.
[15] R. Khazaka, P.K. Gunupudi, M.S. Nakhla, Efficient sensitivity analysis of transmission-line networks using model-reduction techniques, IEEE Transactions on Microwave Theory and Techniques 48 (2000) 2345–2351.
[16] S.S. Kim, M.J. Vanderploeg, QR decomposition for state space representation of constrained mechanical dynamic systems, Journal of Mechanisms, Transmissions, and Automation in Design 108 (1986) 183–188.
[17] T. Kozaki, H. Mori, H.K. Fathy, S. Gopalswamy, Balancing the speed and fidelity of automotive powertrain models through surrogation, in: Proceedings of 2004 ASME International Mechanical Engineering Congress and Exposition, vol. 73, ASME, New York, NY, 2004, pp. 249–258.
[18] R. Kübler, W. Schiehlen, Modular simulation in multibody system dynamics, Multibody System Dynamics 4 (2000) 107–127.
[19] C.G. Liang, G.M. Lance, A differentiable null space method for constrained dynamic analysis, Journal of Mechanisms, Transmissions, and Automation in Design 109 (1987) 405–411.
[20] R.M. Lin, M.K. Lim, Structural sensitivity analysis via reduced-order analytical model, Computer Methods in Applied Mechanics and Engineering 121 (1995) 345–359.
[21] L.S. Louca, J.L. Stein, G.M. Hulbert, J. Sprague, Proper model generation: an energy-based methodology, in: Proceedings of 1997 International Conference on Bond Graph Modeling, vol. 29, SCS, 1997, pp. 44–49.
[22] N.K. Mani, E.J. Haug, K.E. Atkinson, Application of singular value decomposition for analysis of mechanical system dynamics, Journal of Mechanisms, Transmissions, and Automation in Design 107 (1985) 82–87.
[23] B.C. Moore, Principal component analysis in linear systems: controllability, observability, and model reduction, IEEE Transactions on Automatic Control 26 (1981) 17–32.
[24] A.Y. Orbak, O.S. Turkay, E. Eskinat, K. Youcef-Toumi, Model reduction in the physical domain, Proceedings of the Institution of Mechanical Engineers Part I: Journal of Systems and Control Engineering 217 (2003) 481–496.
[25] N. Orlandea, M.A. Chace, D.A. Calahan, A sparsity-oriented approach to the dynamic analysis and design of mechanical systems – part 1, Journal of Engineering for Industry 99 (B) (1977) 773–779.
[26] D.G. Rideout, J.L. Stein, L.S. Louca, System partitioning and physical-domain model reduction through assessment of bond graph junction structure, in: Proceedings of IMAACA'04, Bond Graph Techniques for Modeling Dynamic Systems, SCS, 2004.

[27] J.R. Rinderle, B.L. Subramaniam, Automated bond graph modeling and simplification to support designProceedings of Winter Annual Meeting of the ASME, vol. 34, ASME, New York, NY, 1991, pp. 45–68.
[28] W. Schiehlen, Computational aspects in multibody system dynamics, in: Proceedings of Second World Congress on Computational Mechanics, vol. 90, 1991, pp. 569–582.
[29] R.P. Singh, P.W. Likins, Singular value decomposition for constrained dynamical systems, Journal of Applied Mechanics 52 (1985) 943–948.
[30] S. Skogestad, I. Postlethwaite, Multivariable Feedback Control: Analysis and Design, John Wiley & Sons, 1996.
[31] B.J. Sohns, J. Allison, H.K. Fathy, J.L. Stein, Accelerating parameterization of large-scale dynamic models through the use of activity analysis, in: Proceedings of 2006 ASME International Mechanical Engineering Congress and Exposition, ASME, New York, NY, 2006.
[32] C. Sueur, G. Dauphin-Tanguy, Bond graph approach to multi-time scale systems analysis, Journal of the Franklin Institute 328 (1991) 1005–1026.
[33] W.C. Walton, E.C. Steeves, A new matrix theorem and its application for establishing independent coordinates for complex dynamical systems with constraints (NASA Technical Report, TR-326, 1969).
[34] R.A. Wehage, E.J. Haug, Generalized coordinate partitioning for dimension reduction in analysis of constrained dynamic systems, Journal of Mechanical Design 104 (1982) 247–255.
[35] J. Wittenburg, Dynamics of Systems of Rigid Bodies, Teubner, Stuttgart, 1977.