2016 Conference on Systems Engineering Research

An Improved Method of Physical Interaction and Signal Flow Modeling for Systems Engineering

Mehdi Dadfarnia,[a]* Conrad Bock,[a] Raphael Barbau[b]

[a]*U.S. National Institute of Standards and Technology*
[b]*Engisis, LLC*

Abstract

Many engineering disciplines that interact with systems engineering use simulation tools to build models based on equations, which are solved to predict physical and informational behavior over time. Both kinds of engineering are concerned with physical and informational behaviors within systems, but focus on different aspects of these behaviors, discouraging communication and producing conflicting or erroneous models. Systems engineering models usually specify how components are interconnected and broken down into subcomponents (system structure), limiting behaviors to occur within components or between interconnected components. Equation-based simulation models have equations relating variables that are not limited by system structure. The gap between systems and equation-based simulation models is smaller for simulation models that incorporate structure, in particular those covering physical interaction and signal flow. Unfortunately, adoption of this kind of simulation has been slowed by a lack of simulation modeling methods based on physical principles. This paper lays out an improved method for developing physical interaction and signal flow simulation models that align with systems models. To capture the underlying physical principles, this method emphasizes the transformation and transmission of conserved physical substances and numeric information within system structures, with equations chosen to describe these processes. Simulation models are easier to develop this way and are better aligned with systems engineering models. The method is presented using the Systems Modeling Language, extended in prior work for physical interaction and signal flow modeling that supports code generation to widely-used simulation tools.

Keywords: System structure; physical simulation; lumped-equation modeling

1. Introduction

Systems engineers use system modeling tools to organize their work, and interact with many other engineers who use equation-based simulation tools to predict system behavior over time.[1] Systems modeling and simulation tools are based on computable representations, to answer questions without the difficulties of prototyping and experimenting

* Corresponding author. Tel.: +1 301-975-4659.
E-mail address: mehdi.dadfarnia@nist.gov

on real systems. Both kinds of engineering are concerned with physical and informational behaviors within systems, but focus on different aspects of these behaviors and adopt different approaches to modeling them:

- Systems engineering models usually specify how components are interconnected and broken down into subcomponents (system structure).[2] This is needed to organize and build systems, especially complex ones. Interconnections between components are meant to reflect physical interactions and information exchanges involving multiple components, helping systems engineers coordinate others in specialized engineering disciplines, while depending on them to use equation-based simulation tools for complete descriptions of system behaviors.

- Equation-based simulation models have variables quantifying various aspects of desired systems. These equations are needed to predict system behaviors for comparison to required behaviors, especially in complex designs. The equations are meant to reflect physical and informational behaviors, but can relate variables about components that are widely separated in structures provided by systems engineers, or in structures that do not accurately reflect physical interactions.

The gap between modeling based on system structures and equation-based modeling discourages communication between systems engineers and those using equation-based simulation tools, producing conflicting or erroneous models, and requiring additional work to resolve the differences. Simulation and analysis engineers (typically in specialized engineering domains) need to modify their equation-based models to follow system structures defined by systems engineers, and systems engineers need to change system structures based on simulation feedback about complex physical and informational behavior.

The gap between modeling based on system structures and equations is smaller for equation-based simulation models that incorporate system structure (also known as “lumped parameter,” “one dimensional,” or “network” models,[3,4,5] referred to as physical interaction and signal flow models in this paper). These kinds of simulation models encourage communication between systems and simulation engineers based on their shared attention to interconnection and the breakdown of components. Simulation tools supporting physical interaction and signal flow modeling present graphical interfaces for specifying equations in system components and linking components together to form larger components. The tools generate additional equations from the links between components, then solve the equations and report predicted values of physical and informational behavior over time.[6,7,8]

Unfortunately, the adoption of physical interaction and signal flow simulation, and alignment with systems models, has been slowed by a lack of simulation modeling methods based on physical principles. Many engineering disciplines still use simulation methods that encourage development of equations relating variables about components widely separated in structures provided by systems engineers. The most basic tools of this kind are text-based, accepting equations entered in the order in which the tools should solve them without regard to system structures.[9,10,11] Graphical tools facilitate construction of these ordered equations by dividing them into “blocks” that define inputs and outputs as well as connecting outputs to inputs to specify solving order.[8,12,13] Equation-based simulation tools that encourage modeling based on system structure[6,7,8] lack methods to develop system structures that accurately reflect physical interactions, and align better with systems models.

This paper presents a method for developing physical interaction and signal flow simulation models that are better aligned with systems models and more accurately reflect physical interactions. The method starts by sketching transformation and transmission of conserved physical substances and numeric information within the system, then developing system structures and equations to describe those processes. Starting from physical processes occurring in system structures is more intuitive than piecing together equations with less guidance. The method aligns the structure of systems and simulation models because physical processes occur within single components and between components that are closely connected structurally, and involve the transformation and transmission of conserved physical substances, which systems engineers need for coordinating others in specialized disciplines. The method guides development of equations based on physical processes through graphical representation of physical principles, relating equations to conserved substance flow within single components and between directly connected components in system structures. The method encourages systems and simulation analysis engineers to discuss the desired physical processes around shared system structures.

In addition to better alignment between systems engineering models and physical interaction and signal flow simulation models, the proposed method produces more realistic system models and facilitates construction of simulation models. System models with structures developed around physical processes are more likely to be successfully simulated, built, and operated. Simulation models with equations describing physical processes that identify conserved substances being exchanged are easier to develop because the range of possible equations is narrowed significantly by considering how the conserved substances propagate through system components incrementally.

Existing descriptions of physical interaction and signal flow modeling either skip over physical processes or address them only as energy exchange between components, as described in Section 2. Some introduce particular equation-based languages supported by automated solvers, rather than methods for developing models based on physical processes. Many give methods for developing system structure, which aids in aligning with systems engineering models, yet most methods do not relate the structure and equations to underlying physical processes. Some of these give methods for developing models based on physical processes to varying degrees but only address interactions between components, not processes within components. The authors are not aware of methods that go further into physical processes, particularly within components, or in the particular conserved substances flowing between them.

The method is presented using the Systems Modeling Language (SysML),[14] which was extended in prior work for physical interaction and signal flow modeling that supports code generation to widely-used simulation tools.[15] This enables structure models to be shared between systems and simulation analysis engineers and physical processes to be reflected in centralized systems models rather than the many simulation tools used to verify them.

The paper is divided as follows. Section 2 reviews methods for developing physical interaction and signal flow models and identifies their deficiencies for integrating with systems modeling. Section 3 gives the proposed method. Section 4 summarizes the paper.

2. Related Work

Many descriptions of physical interaction and signal flow modeling are introductions to particular equation-based languages supported by automated solvers rather than methods for developing models based on physical processes. These begin with example equations for simple systems, such as pendulums,[16] or simple equations unrelated to any application.[17] They then describe how to express those equations in textual languages supported by equation-solvers.[18] Physical processes behind the equations, if any, are not discussed. For example, the physical process involved in the movement of pendulums is the cyclical transformation of energy between momentum and potential in a gravitational field, but these introductions do not use physical concepts, such as momentum and potential energy, when explaining the example equations or how pendulums function.

Most descriptions of physical interaction and signal flow modeling give methods for developing system structure, which aids in aligning with systems engineering models, but mostly do not relate to physical processes. These methods show how to break systems into components, subcomponents, and their interconnections, and how to reuse structure to improve maintainability and reliability of models (by generalization, also called “inheritance”).[17] The benefit of system structure is explained in comparison to unstructured sets of equations and to block diagrams, a technique for separating equations into groups connected according to outputs and inputs.[16] One work makes similar comparisons, critiquing the ability of prior approaches to reflect physical processes and explaining the transition to physical interaction and signal flow modeling with a graphical modeling language for energy exchange between components.[19] Yet this graphical modeling language produces complicated networks bearing little resemblance to those of structured systems.

Some descriptions of system structure in physical interaction and signal flow modeling give methods for developing models based on physical processes to varying degrees, but these descriptions only address interactions between components and not processes within components. This might be sufficient when using predefined components (component libraries), but otherwise leaves the modeler without guidance, for example when modeling the environments in which systems are to operate. Many of these descriptions focus on variables for physical quantities shared between interconnected components[17] and sometimes mention that they express rates of energy flow between components for various physical domains (electrical, mechanical, thermal, etc).[16,20] Some also explain that components can generate, consume, or conserve energy depending on rates of energy exchange with other components.[3,6] The authors are not aware of methods that go further into physical processes, particularly within components, or in the particular conserved substances that carry energy between them.

3. Method

The method proposed in this paper starts by developing an informal sketch of system components and their physical and informational interactions. Then systems and simulation analysis engineers collaborate to define SysML diagrams describing the conserved substances flowing throughout a system and the environment in which it will operate (*total system*). It identifies and analyzes the transformation and transmission of conserved substances and numeric information within and between system components, including components of system operating environments. It incrementally verifies system structure and assignment of equations to components against the desired behavior of conserved substances and numeric information. This enables underlying physical principles to drive simulation modeling, providing guidance in the selection and development of equations describing processes. Applying these principles within system structures ensures that resulting simulation models align with systems engineering models.

The method is explained with an example automobile cruise control system commonly used in describing simulation modeling methods,[21,22] including both physical interactions and signal flows. Cruise control systems attempt to keep the speed of a vehicle constant despite environmental disturbances that might affect it, such as changes in the slope of the road, rolling resistance of the tires, and air drag on the vehicle from wind. Cruise control systems respond to these environmental disturbances by adjusting their vehicle's throttle to maintain the desired speed initially set by the driver.

1. *Develop a sketch of the total system components with physical and informational interactions between them*

A sketch of the total cruise control system is shown in Figure 1, including its vehicle and the operating environment of the vehicle, as well as physical and informational processes. The figure shows physical interactions with solid, bidirectional arrows and signal flows with dashed, unidirectional arrows. The driver gives the desired speed to the cruise control system (modeled as a signal). The cruise controller uses the desired speed and the current speed of the vehicle (calculated from a signal transmitted from the wheel giving the wheel rotation rate), to determine how much fuel to inject into the engine subsystem. Then the cruise controller calculates how far to open the engine's throttle and sends the result as a signal to the engine. The engine interacts physically with the wheels, involving angular momentum as the conserved substance. The wheels and road transform angular momentum to and from linear momentum (also a conserved substance) to move the vehicle. In addition, a linear momentum interaction occurs between the vehicle and the air around it, which reduces or increases the momentum of the vehicle depending on the relative speed of the vehicle and air, and another occurs between the vehicle and the gravitational field, which reduces or increases the momentum of the vehicle depending on the slope of the road.

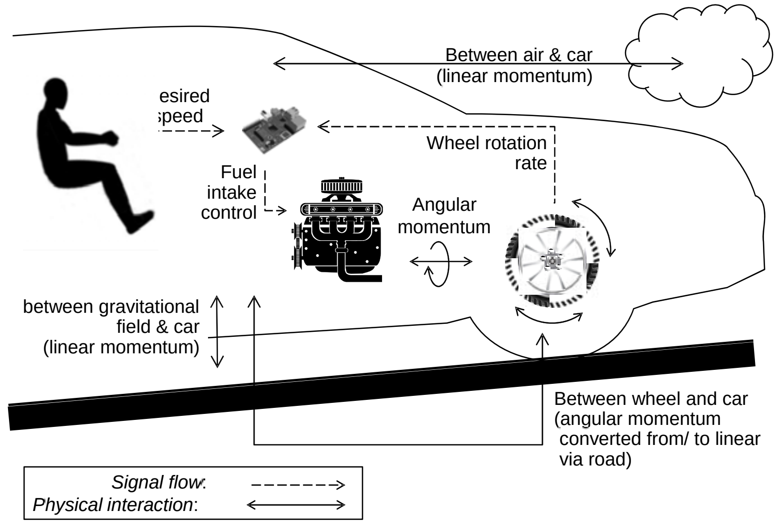

Fig. 1. Total system for a cruise controller.

2. *Model system structure to support physical and informational interactions*

Based on the total system sketch, a structure model is created for all components in the system and their interconnections. The only SysML diagrams that show interconnections between components are internal block diagrams. These diagrams are developed in stages, some of which depend on later steps in the method, but Figure 2 shows the complete internal block diagram for Figure 1, for brevity. Modeling starts with roles (SysML parts) played by components in the total system (like parts in a play). Part names appear in the titles of the rectangles of internal blocks diagrams, to the left of the colons. For example, the parts in Figure 2 include the driver and the impeller. The kinds of things (SysML blocks) that play the parts (part types) appear in titles of the part rectangles to the right of the colons. For example, the part types in Figure 2 are Person, Wheel, and so on. Small rectangles appearing on part rectangles are SysML ports, which are parts of parts (roles of roles) for modeling those aspects of parts related to particular interactions. For example, in Figure 2 impellers are a port of the vehicle, played by wheels.

Parts and ports are linked by SysML connectors along which physical and informational interactions take place in the system (except for binding connectors, explained below). The kinds of physical substances flowing along connectors appear in the labels of filled triangles on connector lines (this is also specified on blocks used as parts or ports, see step 3 of the method). For example, angular momentum flows between the engine and the wheel. Ports with one arrowhead inside are for signals and indicate the direction that numeric information flows through the port (in or out), while ports with two arrowheads inside are for physical interactions and indicate conserved physical substances can flow in either direction depending on circumstances.

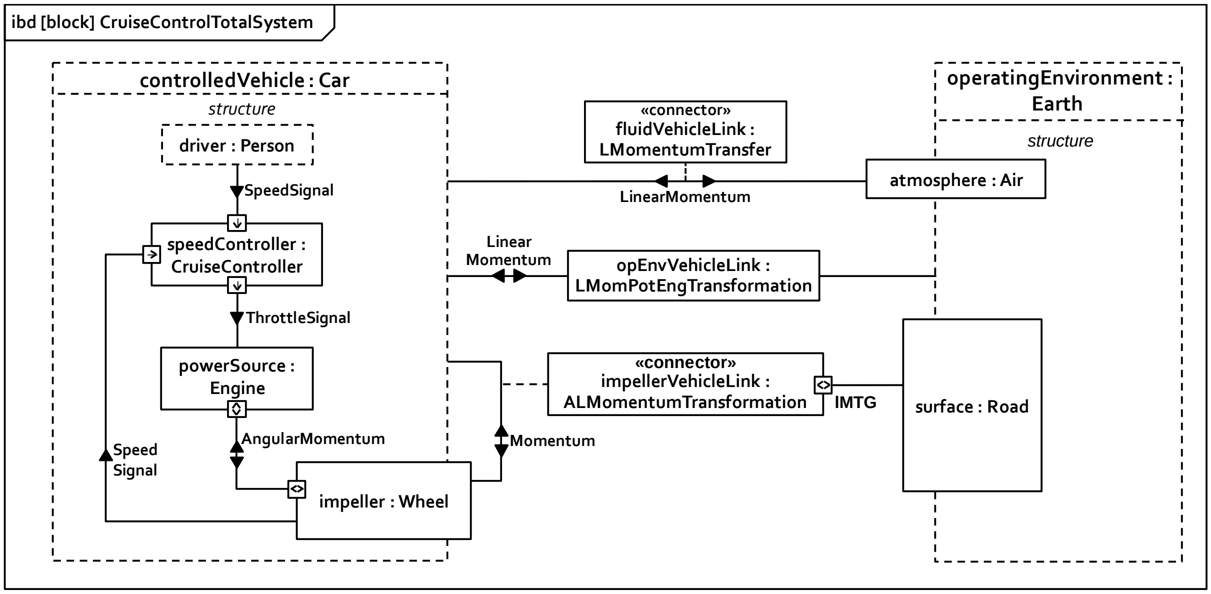

Fig. 2. Total system structure in SysML internal block diagram

The internal block diagram in Figure 2 shows three signal flows and one physical interaction inside the vehicle:

- The driver sends a desired vehicular speed as a signal to the cruise controller's user interface, modeled as a port of the cruise controller.
- The cruise controller receives a signal from the wheel speed sensor to calculate the vehicle's current measured speed. This information is used in the control algorithm to calculate the feedback required to have the car reach the driver's desired vehicular speed.
- The cruise controller sends a control signal to the engine's throttle actuator, affecting the flow rate of fuel into the engine.
- Angular momentum moves between the engine and wheel in a physical interaction (the transmission, drivetrain, and other parts are omitted for brevity).

The vehicle is involved in three physical interactions with its operating environment:

- The transformation between angular momentum of the wheels and linear momentum of the vehicle is modeled with a connector between the wheel and vehicle. The connector has a corresponding connector property (impellerVehicleLink) played by an association block describing the interaction between the wheels, vehicle, and road. The connector between the wheel and vehicle represents the vehicle propelling itself. This transformation only happens when the wheel and road are in contact, even though the road does not accept or provide momentum (the earth is taken as too large to accept significant amounts of momentum). The connector between the interaction and road (via the lMTG port) shows that the road is necessary for the transformation, but does not allow momentum to flow between the wheels and road.
- The transfer of linear momentum between the vehicle and the air is modeled by a connector between the vehicle and the air (ignoring turbulence, heat, and other less organized forms of momentum, for brevity).

The connector has a corresponding connector property (fluidVehicleLink) played by an association block describing the interaction between the vehicle and air.

- The transformation between linear momentum and gravitational potential energy is modeled by a part connected to the vehicle and the Earth (opEnvVehicleLink). This part represents the gravitational field's storage and return of the vehicle's momentum. The transformation is not modeled as a connector to the gravitational field as a part of Earth, as done with the air, because exchanging momentum with the gravitational field would enable the field to pass that momentum to and from other objects, as air can. The part between the vehicle and Earth can only exchange momentum with the vehicle, even though it requires a relationship to Earth to do this.

The connectors with connector property blocks (indicated by dashed lines in Figure 2) are treated as parts played by association blocks, see step 3, enabling them to have equations describing their physical processes, see step 4.

3. *Model the kinds of things used in the system structure*

This step specifies the kinds of things playing parts and ports in more detail than Figure 2. Kinds of things are modeled as SysML blocks, some of which are shown in Figure 3, such as Car and Earth (treating Earth as one of a kind). The lines with filled-in diamond shapes between blocks in Figure 3 are composition associations. These relationships indicate that an object, such as a car, contains other things, such as wheels. Reference associations do not use diamonds and indicate that an object is related to other things without containing them. Associations have property names on their ends, indicating the roles played by the kind of thing on that end. Some of these properties (roles) appear in Figure 2 as part and port names (to the left of colons), such as controlledVehicle and operatingEnvironment. Association blocks represent interactions. Dotted lines run from association blocks to a solid line that links the kinds of things involved in the interaction. For example, in Figure 3, the association block ALMomentumTransformation represents an interaction between the car and Earth through components they are composed of. Composition associations can show relationships between a kind of thing and interactions they contain, such as the relationship between the block CruiseControlTotalSystem and association block ALMomentumTransformation in Figure 3.

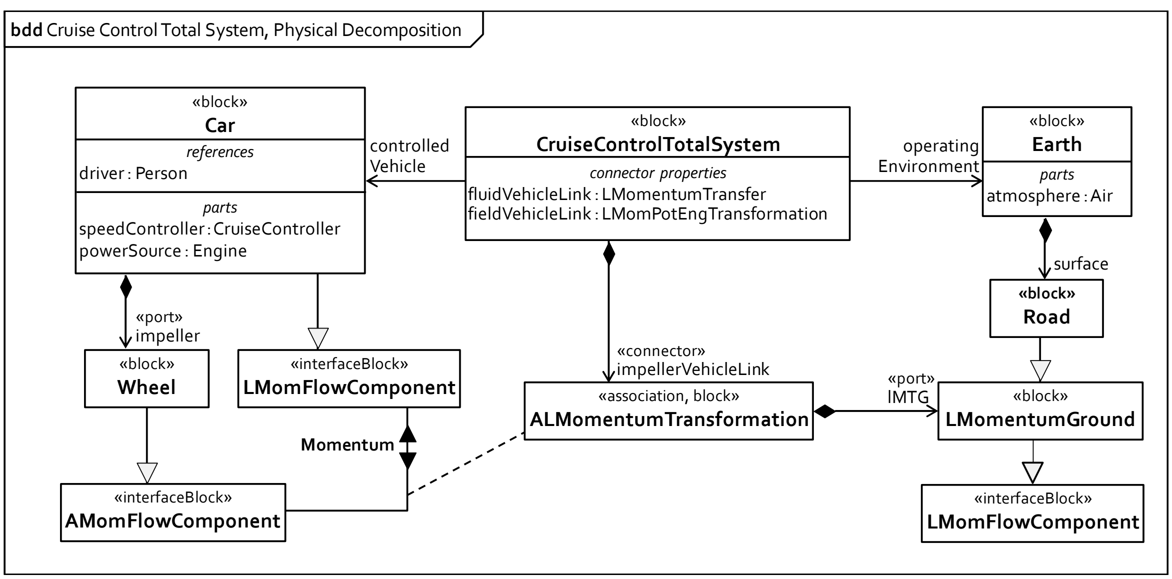

Fig. 3. Some components used by system structure in SysML block diagram

Interface blocks in Figure 3, such as AMomFlowComponent and LMomFlowComponent, model aspects of components that interact with other components. These blocks specify conserved substances or numeric information that flow across the boundary of components. Each kind of component is generalized by interface blocks for the substance or information that flow across its boundary (SysML generalizations appear as lines with hollow-headed arrows). Interface blocks can have SysML flow properties specifying conserved physical substances flowing through them, such as the flow property lMom typed by LinearMomentum (a conserved physical substance) on the interface block LMomFlowComponent in Figure 4. Physical interaction simulation models characterize these conserved substances by their rates of flow of and potentials to flow.[3,6] Rate of flow is a conserved variable, because the flows must balance between components, and potential to flow of a substance is non-conserved, because it remains the same between components. For linear momentum, force is the rate of flow and velocity is the potential to flow. Momentum can flow between objects only if they are moving at different velocities (giving a potential to flow), but actually flows only when they come into contact with each other to exert force (giving a rate of flow). This is modeled by simulation variables on simulation blocks, using an extension to SysML for physical interaction and signal flow modeling from previous work.[15] For example, the simulation variables for torque and angular velocity of the conserved substance angular momentum are trq and aV, typed by Torque and AngularVelocity, respectively. These simulation variables are specified on the simulation block AMomFlow (marked with simBlock) in Figure 4. Similar to physical flows, informational flows are modeled using only a potential to flow variable (omitted from Figure 4 for brevity).

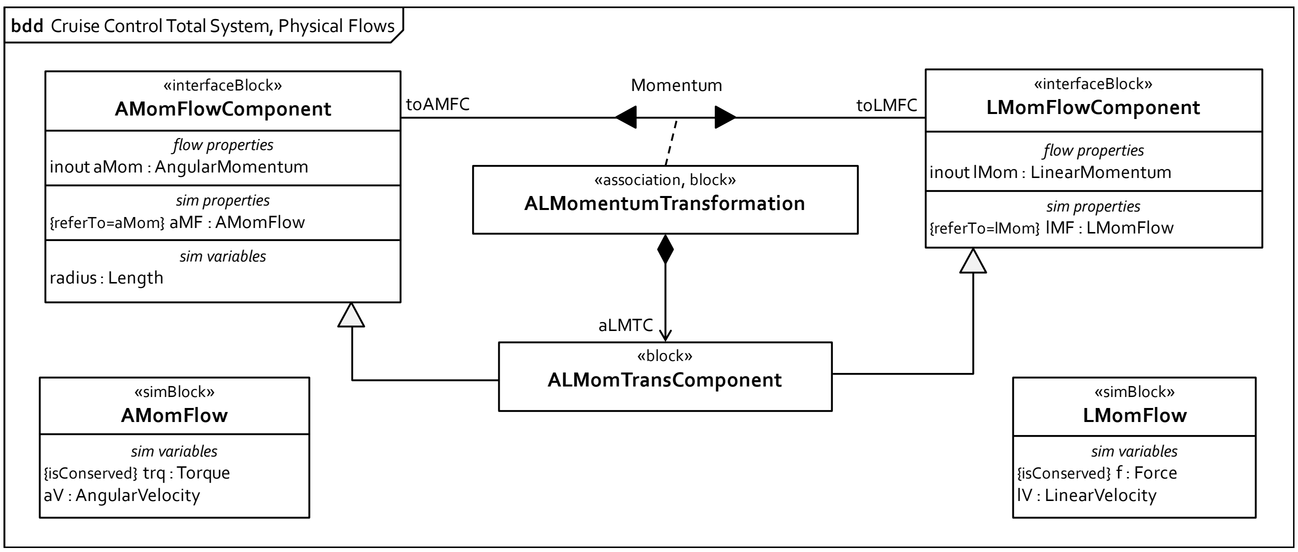

Fig. 4. Physical flows between some components in the system structure (not all flows shown for brevity)

4. *Develop equations based for components involved in physical interactions and signal flows*

This step develops equations (SysML constraints) for the components modeled in step 3, based on their physical and informational interactions in the total system described in steps 1 and 2. SysML constraints are equations contained in blocks that relate properties of their blocks instead of variables. SysML constraint blocks are applied to systems structures with binding connectors between their properties and other properties within internal block diagrams, applying the equations to system structure. Binding connectors are a kind of connector indicating that two properties represent the same thing rather than being involved in an interaction. These are marked by the keyword «equal» next to them. Figure 5 is a parametric diagram (but an internal block diagram could also be used) for the cruise control example applying equations to an interaction between the wheel, the road surface, and the car. The equations specify the transformation between angular momentum and linear momentum moving the car, due to interaction with the road. Properties of the constraint block (constraint parameters, appearing as small rectangles on the edges of the constraint block) are equated to interaction properties via binding connectors. The bound properties are simulation variables, constants, and other properties of interaction, including generalized properties from the participants aMFC (typed by the interface block AMomFlowComponent), lMFC (typed by the interface block LMomFlowComponent), and the port lMTG (for linking to the road's surface, which acts as a reference point for the car's movement).

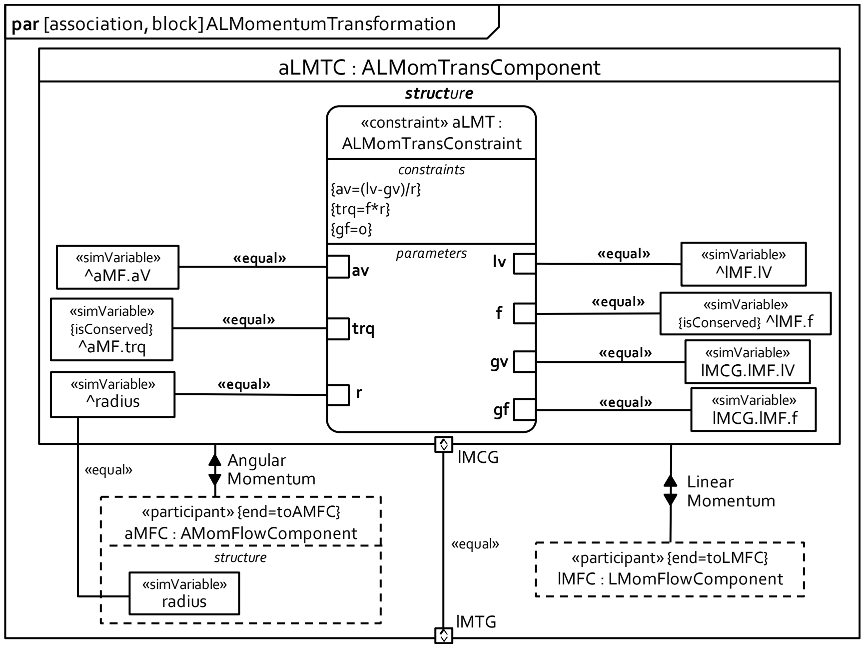

Fig. 5. Equations applied to system structure in SysMLparametric diagram

Models completed by this method contain all information needed to generate code for physical interaction and signal flow simulators. The resulting simulation models will have the same system structure as the source SysML models. The SysML extension for physical interaction and signal flow from previous work enables modelers to add the information for simulation tools to execute.[15]

4. Conclusion

This paper presents an improved method of modeling physical interaction and signal flow simulation to encourage collaboration with systems engineering and model alignment. Physical interaction and signal flow simulation models support structure that could be shared with systems models, but adoption of this kind of simulation has been slowed by a lack of modeling methods based on underlying physical principles. Many existing methods explain equation-based languages supported by automated solvers rather than how to model physical processes. Most show how to model system structure without modeling physical processes within the structural components. Some methods develop system structure based on physical processes to varying degrees, but only address interactions between components, usually as propagation of variable values instead of the conserved physical substances. The few methods for modeling processes within components treat them only as energy operators rather than transformers and transmitters of conserved physical substances.

The method in this paper engages physical principles based on objects transforming and transmitting conserved physical substances and numeric information within total system structures, including objects in the operating environment of the system being specified. For example, modeling gravitational effects on vehicles includes the Earth as a component interacting with vehicles in a total system, instead of immediately reducing Earth to an acceleration constant ("g") in equations. System structure is modeled to support movement of conserved physical substances and numeric information as opposed to using structure to support equations. In the vehicle example, gravitational fields are modeled between Earth and vehicles, converting between linear momentum and potential energy rather than force equations that omit physical substances. Equations resulting from this structural and physical analysis are more likely to describe the desired system accurately.

The method presented here provides more guidance in selecting and developing equations than other methods, especially within components. Equations are chosen to describe physical processes rather than just to solve for variable values. The method begins with informal sketches of system components and their physical and informational interactions. Then the system structure in these sketches is captured more formally in SysML internal block diagrams. These diagrams show component breakdown and interconnections supporting physical and informational interactions. The kinds of things used in the system structure are specified in SysML block diagrams, including ports of components that support physical and informational interactions with other components in the system structure. Based on these diagrams and the desired physical and informational interactions, equations are defined in components as SysML constraints blocks and applied to system structures in SysML parametric diagrams. The resulting models have all the information needed to generate code to physical interaction and signal flow simulators based on the same system structures. The method provides a more straight-forward way to develop physical interaction and signal flow models that are also better aligned with systems engineering models.

Acknowledgements

This material is based in part on work supported by U.S. National Institute of Standards and Technology grant award 70NANB14H249 to Engisis, LLC.

Identification of any commercial equipment and materials is only to adequately specify procedures. It is not intended to imply recommendation or endorsement by the U.S. National Institute of Standards and Technology, nor does it imply that the materials or equipment are necessarily the best available for the purpose.

References

1. Blanchard B. Fabrycky W. *Systems Engineering and Analysis*. New Jersey: Prentice Hall; 2010.
2. IEEE Standards Association. 1220-2005 - IEEE Standard for Application and Management of the Systems Engineering Process. https://standards.ieee.org/findstds/standard/1220-2005.html. 2005.
3. Wellstead P. *Introduction to Physical System Modelling*. London: Academic Press; 1979 (republished by Control Systems Principles; http://www.control-systems-principles.co.uk/ebooks/Introduction-to-Physical-System-Modelling.pdf. 2005).
4. Elmqvist H. A Structured Model Language for Large Continuous Systems. Ph.D. thesis: Lund Institute of Technology, Sweden; http://www.control.lth.se/Publication/elm78dis.html. May 1978.
5. Koenig H.E. Tokad Y. Kesavan H. *Analysis of Discrete Physical Systems*. New York: McGraw-Hill; 1967.
6. The MathWorks, Inc. Simscape™ Language Guide. http://www.mathworks.com/help/pdf_doc/physmod/simscape/simscape_lang.pdf. 2015.
7. Open Source Modelica Consortium. OpenModelica User's Guide. https://openmodelica.org/doc/OpenModelicaUsersGuide/OpenModelicaUsersGuide-latest.pdf. 2015.
8. Controllab Products. Getting Started with 20-sim. http://www.20sim.com/downloads/files/20simGettingStarted45.pdf . 2015.
9. The MathWorks, Inc. MATLAB Primer. http://www.mathworks.com/help/pdf_doc/matlab/getstart.pdf. 2015.
10. Eaton J. Bateman D. Hauberg S. Wehbring R. GNU Octave. https://www.gnu.org/software/octave/octave.pdf. March 2015.
11. Wolfram. NDSolve. https://reference.wolfram.com/language/ref/NDSolve.html. 2014.
12. The MathWorks Inc. Simulink User's Guide. http://www.mathworks.com/help/pdf_doc/simulink/sl_using.pdf. 2015.
13. National Instruments Corporation. LabVIEW Control Design and Simulation Module. http://www.ni.com/labview/cd-sim. 2015.
14. Object Management Group. OMG Systems Modeling Language version 1.4. http://www.omg.org/spec/SysML/1.4. September 2015.
15. Bock C. Matei I. Barbau R. Integrating Physical Interaction and Signal Flow Simulation with Systems Engineering Models. In: Proceedings of the NAFEMS World Congress. Glasgow: NAFEMS; 2015.
16. Tiller M. *Introduction to Physical Modeling with Modelica*. Kluwer Massachusetts, Academic Publishers; 2001.
17. Fritzson P. *Introduction to Modeling and Simulation of Technical and Physical Systems with Modelica*. New Jersey: IEEE Press; 2011.
18. Modelica Association. Modelica - A Unified Object-Oriented Language for Systems Modeling Language Specification version 3.3 revision 1. https://www.modelica.org/documents/ModelicaSpec33Revision1.pdf. July 2014.
19. Cellier F. Elmqvist H. Otter M. Modeling from Physical Principles. In: Levine W, editor. *Control System Fundamentals*. Florida: CRC Press; 1999, p. 99-108.
20. Controllab Products. Reference Manual 20-sim. http://www.20sim.com/downloads/files/20simReference43.pdf. 2013.
21. Angstrom K. Murray R. *Feedback Systems: An Introduction for Scientists and Engineers*. New Jersey: Princeton University Press; 2008.
22. Gene F. Powell J. Emami-Naeini A. Feedback Control of Dynamic Systems. 4th ed. New Jersey: Prentice Hall; 2002.