Mechanism and Machine Theory 131 (2019) 172–188

Contents lists available at ScienceDirect

Mechanism and Machine Theory

journal homepage: www.elsevier.com/locate/mechmachtheory

Research paper

Energy-leak monitoring and correction to enhance stability in the co-simulation of mechanical systems

Francisco González[a,c,*], Siamak Arbatani[a], Arash Mohtat[a,b], József Kövecses[a]

[a] *Department of Mechanical Engineering and Centre for Intelligent Machines, McGill University, 817 Sherbrooke Street West, Montréal, Québec H3A 0C3, Canada*
[b] *General Motors of Canada, 101 McNabb Street, Markham, ON L3R 4H8, Canada*
[c] *Laboratorio de Ingeniería Mecánica, University of A Coruña, Mendizábal s/n, Ferrol 15403, Spain*

ARTICLE INFO

Article history:
Received 16 May 2018
Revised 25 July 2018
Accepted 9 September 2018

Keywords:
Co-simulation
Energy monitoring
Adaptive damping
Passivity

ABSTRACT

Non-iterative co-simulation is an increasingly important technique for the simulation of complex mechanical systems. Adopting co-simulation schemes enables the simultaneous use of computational resources and makes it possible to select the most appropriate modelling techniques and algorithms to describe and solve the dynamics of each system component. However, it inherently requires the coupling of different subsystems at discrete communication times, which may compromise the stability of the overall integration process. One of the negative effects of discrete-time communication is the introduction of artificial energy in the system dynamics, which can render the simulation unstable if it accumulates over time. Excess energy can be dissipated introducing virtual damping elements in the subsystem models. The actual amount of damping must be adjusted as the simulation progresses to ensure that all the artificially generated energy is removed from the system while keeping the dynamics realistic. In this paper, we introduce a monitoring framework to keep track of this excess energy, and put forward a dissipation methodology to eliminate it. The ability of this framework to achieve stable non-iterative co-simulation was tested with several mechanical system examples.

1. Introduction

Forward-dynamics simulation is an important method to predict the behaviour of mechanical systems for complex industrial applications. Due to advances in computational power and solution algorithms, the range of problems that can be addressed this way has considerably expanded during the last decades. On the other hand, the expectations about simulation output have grown at a similar pace. Nowadays, dynamics simulation is expected to accurately predict the behaviour of sophisticated engineering systems in an efficient and stable way. In the case of mechanical systems, currently used models often include challenging phenomena such as contacts and friction, flexibility, and interactions with non-mechanical components such as hydraulics and electronics. Although the size and required level of detail of the systems under study continue to increase, efficient execution is required from the simulation software; in some cases, such as Human- and Hardware-in-the-Loop (HiL) environments, real-time performance must be achieved.

* Corresponding author at: University of A Coruña, Mendizábal s/n, Ferrol 15403, Spain.
E-mail addresses: f.gonzalez@udc.es (F. González), arbatani@cim.mcgill.ca (S. Arbatani), mohtat.arash@gmail.com (A. Mohtat), jozsef.kovecses@mcgill.ca (J. Kövecses).

https://doi.org/10.1016/j.mechmachtheory.2018.09.007
0094-114X/

The coupling of several solver tools in a co-simulation setup is a way to deal with these requirements [1,2] that represents a modular alternative to monolithic solution methods [3,4]. This can be done following different approaches; a wide variety of co-simulation techniques have been proposed during recent years and new methods keep on being developed to deliver efficient and robust simulation procedures [5,6]. Dividing the overall application into subsystems enables the selection of different solution strategies for each, making it possible to tailor the solver parameters to particular physical properties and time scale. Additionally, co-simulation makes it easier to share the computational workload among several processors or CPU cores when they are available [7]. Moreover, each subsystem needs to share only a limited amount of information, namely its inputs and outputs or *coupling variables*, with the rest of the components, avoiding the need to disclose its internal implementation details. This is an attractive feature when using software models protected by intellectual property rights. Co-simulation, however, brings in the need to synchronize the execution of the different solvers, and this can only take place through the exchange of the coupling variables at discrete communication instants. Between these communication points, in the time interval often referred to as the *macro time-step*, the integration of each subsystem proceeds on its own, without any inputs from the rest of the components with which it interacts.

The discrete-time nature of the communication between subsystems in co-simulation environments gives rise to a series of issues that do not exist when a single simulation tool is used to solve the dynamics of the whole system. From the implementation point of view, it is necessary to define communication standards according to which all the components can exchange information in a unified format. This need has been addressed with the definition of the Functional Mock-up Interface (FMI) [8]. Another problem derived from subsystem coupling is the introduction of discontinuities in the numerical integration that may render the simulation unstable if they are not handled appropriately. The origins of this instability can be explained in different ways. In many cases the inputs and outputs exchanged between subsystems, i.e., the coupling variables, are updated at the communication points and kept constant during the integration of the subsystems within macro time-steps, following a zero order hold (ZOH) extrapolation approach. This introduces a discontinuity in the subsystem inputs every time that they are updated; the problem remains even when polynomial approximations are used to extrapolate the inputs within the macro time-step [9]. In co-simulation environments in which physical, real-time components are coupled to numerical models, additional communication problems such as time delays, data loss and noise may arise and further compromise stability [10]. Ways to deal with these discontinuities and errors include the use of iterative co-simulation coupling schemes [1] and using information about the Jacobian matrices that relate coupling variables and subsystem states [11]. These options cannot always be used, however. It is not guaranteed that the outputs of every subsystem will include the necessary Jacobian matrices. Also, some co-simulation environments cannot use iterative coupling schemes and must rely on single-step co-simulation, either because one or more subsystems do not allow retaking an integration time-step, or because the available time to carry out the computations is limited. Another possibility is monitoring the coupling error in the frequency domain, and adjusting the macro step-size accordingly [12]; this method can be employed in non-iterative co-simulation and uses information obtained from the coupling signal itself.

Considering the energy exchanges in the simulated system is another possible way to assess its stability properties and the accuracy of the numerical integration [13]. The interpretation of the coupling errors due to input extrapolation as generated or dissipated energy led to the definition of the NEPCE (Nearly Energy-Preserving Coupling Element) [14], that corrects the coupling variables to ensure energy conservation at the interface. Also, when the coupling variables carry information about the power exchanged between subsystems at the interface, it is possible to adjust the macro step-size accordingly and improve simulation accuracy [15].

Besides its use in co-simulation applications, monitoring energy generation and flow has also been used in passivity control of haptic devices [16,17]. Haptic devices are human-operated mechanical systems that interact with a virtual environment through a physical-virtual interface in which force and velocity quantities are exchanged. Haptics applications can be considered as a special case of co-simulation with two coupled subsystems, one of which is the human-operated haptic device and the other is a computational model of a virtual environment. The communication between these subsystems takes place at discrete instants only, and this can cause *energy leaks*, i.e., the artificial introduction or removal of energy in the system [16].

Several approaches have been proposed in the haptics research community to detect and remove energy leaks. The concepts of *passivity observer* (PO) and *passivity controller* (PC) were introduced in [16]. A passivity observer keeps track of the energy that flows in and out of the subsystems, based on the inputs and outputs that they exchange. A passivity controller dissipates the superfluous energy that can cause passivity violations by acting on the input variables of the subsystems. Early passivity observers and controllers made use only of the information conveyed by the coupling variables at the interface between subsystems. However, when information about the internal energy of the controlled subsystems is available, more robust algorithms can be designed. These strategies include the modification of the exchanged force values at the interface [17] and the introduction of adaptive damping coefficients that are adjusted to dissipate the energy leaks monitored by the PO [18,19]. We propose in this paper to develop similar energy-based methods for co-simulation setups.

Most software tools for the simulation of multibody dynamics are able to provide information about internal system energy. In a computational environment for the co-simulation of mechanical systems, this information can be used to keep track of energy leaks and introduce actions to remove them and make the co-simulation stable. The approach can be particularly convenient in co-simulation setups in which the coupling variables carry information about the exchanged energy, e.g., in force-displacement coupling cases. Ideally, in co-simulation environments, energy leaks must be monitored in all the coupled subsystems. In a haptic device the physical component and the operator can often be assumed to be