The Theory of Bond Graphs in Distributed Systems and Simulations

Stian Skjong **Eilif Pedersen**
PhD Candidate **Associate Professor**
stian.skjong@ntnu.no **eilif.pedersen@ntnu.no**

Department of Marine Technology
Norwegian University of Technology and Science

Keywords: Distributed Systems and Simulations · Bond Graph Theory ·Dynamic Stability·Solver Stability·Hybrid Causality Models

Abstract

The bond graph theory provides a firm and complete strategy for making mathematical models and are used in this work to obtain a good relation between connectivity, causality and model fidelity in distributed systems. By distributing a system more computational power is available which makes it possible to increase the model fidelity in large systems without increasing the time to solve the total system. Also, more complex models with causality switching properties may be used for simplifying the connectivity problem between distributed models and for representing changing dynamics that also affects the model causality.

Stability of distributed systems are dependent on both solver stability and dynamical stability, when neglecting the stability results based on cascaded systems with certain passivity properties. For linear distributed systems solver with fixed step size solvers a stability criterion involving the system dynamics, local solver time step and global synchronization time step can be formulated. In this work a stability criterion for linear distributed systems solved with the Euler integration method will be derived and a hybrid causality model, representing a small power plant, will be used to test the stability criterion.

1. INTRODUCTION

The bond graph modelling theory provides a complete, power conserving connectivity potential between different mathematical models as well as for energy domains. Different models or dynamic effects are connected by power variables through power bonds, dependent on model causality orientations [1]. Such a connection is closely related to real physical systems where each action forces a reaction. This modelling strategy is well suited when considering distributed systems and models, which is the main topic of study in this paper. In distributed systems the dynamics of surrounding submodels are not always known and subsystems may be ”black box” implementations where only the inputs and outputs are known. Then, a power conserving connectivity strategy would give a generic input-output mapping between different submodels since the inputs and outputs are power variables and only dependent on the energy domain and the causality orientation.

In general, a distributed system can be thought of as a collection of different mathematical models, as shown in figure 1. These mathematical models should be connected to each other through a well defined and generic input-output configuration. Such a configuration can be found in object oriented modeling languages and strategies such as the bond graph theory [2]. As can be seen in the figure, each subsystem may have internal control loops independent of the surrounding systems as well as control loops including several subsystems.

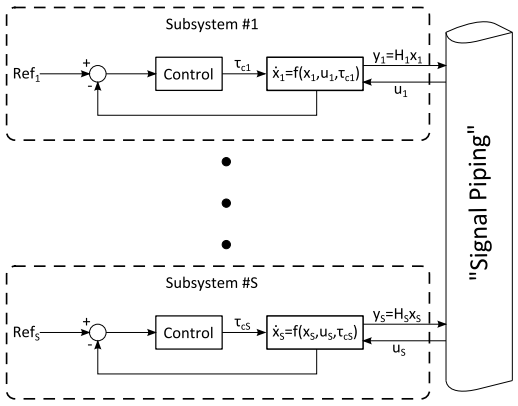

Figure 1. A general distributed system.

One of the many benefits with distributing a system is that a distributed model standard, such as the function mock-up interface, FMI [3], can be used when exchanging models. This means that one can use the favourable modelling and simulation software at hand as long as it has export functionalities in accordance with the chosen standard. On the other hand, a few new challenges arise when distributing a system, such as stability, connectivity and causality, among others. Also, a simulation master algorithm is needed for configuring the total distributed system in a co-simulation and controlling the global simulation time. Such a simulation master

ICBGM’2016 (SummerSim 16), July 24-27,
Montréal, Québec, CA

is considered and developed in the ViProMa-project, Virtual Prototyping of Marine Systems and Operations, tailored for the FMI-standard.

In distributed simulations each distributed submodel is often solved locally with a time step smaller than, or equal to, the global synchronization time step. Also, there will be one global synchronization time step delay between each input-output connection since information is only exchanged at each global synchronization time step. This may lead to a difference in the transmitted power through a power bond when comparing values locally in two connected submodels. This power difference is often normalized and referred to as power residuals and the simulation errors will increase when the power residuals increase. A global algorithm based on power residual calculations and implemented in the distributed simulation master may be used to compensate for the added simulation errors.

Stability in models are often split into dynamic stability [4, 5] and simulation or solver stability. However, in distributed systems and simulations these two stabilities are closely connected since the global synchronization time step sets the maximal frequency of the exchanged data which may suppress the dynamics in the submodels. Also, it can be shown that the stability of the solver dynamics are dependent on both the local time steps, the global synchronization time step and the surrounding submodel dynamics. However, the proof of this is omitted in this work.

By distributing models it is possible to model more complex systems due to gained computational power as well as the possibility to use different local solvers. To gain even more complexity in distributed simulations it is possible to implement submodels with hybrid causality properties. In this context a model with hybrid properties is defined as a model that is able to alter inputs and outputs in discrete time online during a simulation. This enables modelling of connection and disconnection of submodels, for example generator synchronization in a power plant , as well as modelling system failure characteristics that change the dynamics of a model completely. However this calls for a thorough system stability analysis as well as choosing power conserving initial conditions.

2. CONNECTIVITY AND CAUSALITY IN DISTRIBUTED SYSTEMS

In most modelling theories connectivity and causality are closely related. However, in bond graph theory connectivity can be expressed directly through causality for power bond connections. This, because the power variables are given by the energy domain and only equal energy domains can be connected. The power variables, that as a product gives power $[W]$, constitute a *true* power bond and are always present in pairs of efforts and flows. As an example, a force F is an effort and velocity v is a flow and the power is given as $P = F \cdot v$. In this context F and v form a pair of power variables and are inseparable. Where one of them appears in a model, so does the other one.

Table 1. Bond graph elements generating states.

Description	Bond graph	Relation
"Inertia element"	⟶I	$f = \Phi^{-1}(\int_0^t e dt)$
	⟼I	$e = \frac{d}{dt}[\Phi(f)]$
"Capacitive element"	⟼C	$e = \Phi^{-1}(\int_0^t f dt)$
	⟶C	$f = \frac{d}{dt}[\Phi(e)]$

The connectivity problem is then reduced to a combination of a causality problem and a fidelity problem when sticking to the use of power variables consistently when developing mathematical models. This is old news in the bond graph environment, where connectivity is most often discussed in the concept of causality for a given model fidelity. In general, the causality of a model is defined by the form of the equations extracted from the bond graph model and is divided into differential causality and integral causality. A system that, in general, has only one degree of freedom can be modelled having either integral causality or differential causality, depending on the model input. In other words, the model input may act as a model constraints. A system that has multiple degrees of freedoms may include both integral causalities and differential causalities in its model representation, but has one integral form for each unconstrained degree of freedom, since it is the only causality form producing state equations. Differential causality will be treated more thoroughly in section 3.. In summary, both the state space representation and the connectivity options for a model are explicitly given by the causality of the bond graph elements in a model representation for a given model fidelity. Moreover, the causality of bond graph elements are affected by the model input, due to possible model constraints, which proves consistency in the bond graph theory.

In this paper a system consisting of only integral causality forms, will be said to have a complete integral causality form. In bond graph theory only two bond graph elements have the potential of producing states, namely the inertia element, denoted **I**, and the capacitive element, denoted **C**, as shown in table 1 with corresponding equations.

To illustrate the concept of connectivity through causality and model fidelity, consider a simplified model of a diesel engine, as given in figure 2, where a **MSe**-element is used as a governor, mapping fuel energy into torque applied to the crankshaft, a **R**-element taking care of the friction losses, an **I**-element describing the engine inertia and a connecting port,

u_E, that gives the shaft speed as a model output and gets the shaft torque in feedback.

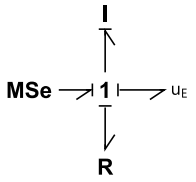

Figure 2. Bond graph model of diesel engine

Now, consider that the diesel engine model is supposed to drive a simplified generator model, as given in figure 3, where y_G is torque given as model input, the **I**-element to the left describes the rotational inertia and the **I**-element to the right is an electrical inductance. The **GY**-element transforms the mechanical power to electrical power and the **R**-elements are friction and an electrical load, from left to right in the figure, respectively.

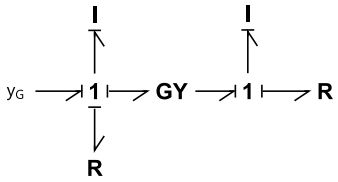

Figure 3. Bond graph model of generator

It can be seen from the two models that a connectivity problem arises if the two systems were implemented as two independent submodels, both having a complete integral causality form, and connected in a distributed system. This, because both models give shaft speed as output and get a torque in feedback. However, since both models should experience the same speed when connected, one of the **I**-elements describing rotational inertia should be constrained and change causality in order to assure connectivity. Another solution is to combine the two **I**-elements into one and put it in either the generator model or the engine model, which is done in figure 4.

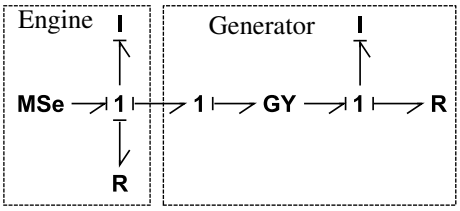

Figure 4. Distributed system of diesel engine and generator.

As can be seen in the figure, the engine has the shaft speed as output and gets a torque in feedback. Note that if the **I**-element were moved to the generator submodel, the engine output would be torque and the shaft speed would have been given in feedback from the generator model. This means that we can choose which submodel to implement the inertia effects without affecting the connectivity. Another possible way of solving the connectivity problem between the engine model and the generator model is to model a stiff coupling between the generator and the engine, increasing the total model fidelity. This would also increase the number of state in the total system by one. However, this solution is not always preferable, even though there often exists such couplings in real systems. This, because a coupling introduces a fast time constant that most likely would stand out from the other time constants in the total system which is usually not preferable.

In this section the causality, connectivity, model fidelity and power variables have been discussed and the strong concept of causality and power variables in bond graph theory help bringing explicit solutions to the connectivity problem in distributed systems. However, the modeller is still left with a lot of choices such as model fidelity and how to divide a total model into submodels for distribution purposes. This means that the bond graph theory doesn't set restrictions for the modeller, but adds constraints in form of causality and power variables that help solving connectivity problems.

Another nice contribution from the bond graph theory is the bond graph itself. When understanding the concept of bond graphs it is easy to characterize different dynamics and understand the model dynamics out from looking at the mnemonic bond graph elements as well as determining model connectivity and extracting state space equations. These are nice features when studying hybrid causality models.

3. HYBRID CAUSALITY MODELS

Model switching is a well known term in the field of mathematical modelling. In general it involves switching between different models online during a simulation but where the input- and output configurations are fixed. Such systems are well treated in [6] and will not be studied in this work. However, hybrid *causality* models that have more or less fixed dynamics but the opportunity to switch between inputs and outputs is of great interests in distributed systems. This, because of the connectivity. By implementing hybrid causality properties in a model the connectivity would only depend on the energy domain. Another reason for implementing hybrid causality properties in a model is to be able to switch causality online during a simulation. This is a nice feature when working with systems having discrete events, for example switches in power electronics. When establishing hybrid causality properties for a model it is also possible to add hybrid dynamics to it. However, this is not in the scope of this work.

One of the reasons for investigating hybrid causality models is to be able to make fast and robust simulation models of systems that seems to change causality when certain events occur. One example of such a system is a hydraulic system with an accumulator that fails if it runs dry. In this case, one

of the other hydraulic components that still works need to provide the pressure to the other hydraulic components, seen from a connective point of view. Another example is a set of generators in a marine power plant that are to be connected to a common weak power grid. Then, only one of the generators can set the power grid voltage, and the rest of the components connected to the power grid provides with a current, either as a consumer or a producer.

In this paper the theory of hybrid causality models will be presented in accordance with the bond graph theory together with the definition of discrete- and continuous time events, which will be elaborated later on. To present the hybrid causality the distributed diesel engine and generator model given in 2. will be used. However, now it is assumed that the engine model and the generator model, together called a *genset*, are implemented as one model and connected to a electrical load that may be distributed.

3.1. Example: Distributed Genset model Connected to an Electrical Load

Consider the genset model with complete integral causality as shown in figure 5, including the model parameters. Note that the engine torque is in this model set to a constant τ.

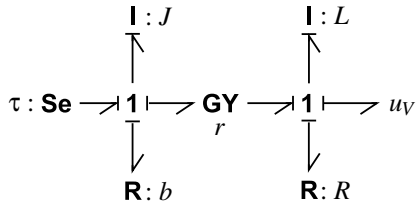

Figure 5. Distributed genset model.

In this model the voltage is given as an input, u_V. Assume that an electrical load model with current as output is to be connected to the genset model. The connectivity in this case would fail if the genset model isn't able to adapt to the required causality. However, connectivity of the two distributed models can be ensured by switching causality on the inductance **I**-element in the genset model. It is then possible to use a switched **1**-junction, denoted **1s** [7] to switch between the causalities. In addition, by adding one more port, u_i, a hybrid causality model can be modelled as shown in figure 6.

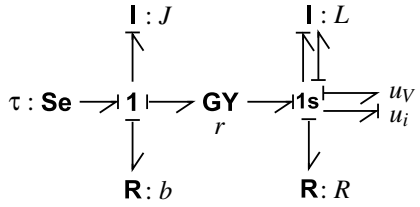

Figure 6. Distributed hybrid causality genset model.

The genset model shown in the figure has a complete integral causality option giving current as output and a causality option including a differential causality giving voltage as output, making the connectivity fit all possible loading models. Note that an external signal is given to the **1s**-element and the I-element in order to switch between the causality options.

From the bond graph a state space equations for the complete integral causality form can be expressed as

$$\begin{aligned} \dot{\omega} &= \frac{1}{J}(\tau - b\omega - ri) \\ \frac{d}{dt}i &= \frac{1}{L}(r\omega - u_V - Ri) \\ y &= i \end{aligned} \tag{1}$$

where ω is the shaft speed of the genset, i is the electrical current, J is the rotational inertia, b is a friction constant, r is the gyrator modulus, L is the inductance, R is an electrical resistance parameter and y is the model output. The equations for the differential causality option can be written as

$$\begin{aligned} \dot{\omega} &= \frac{1}{J}(\tau - b\omega - ru_i) \\ v &= r\omega - L\frac{d}{dt}u_i - Ru_i \\ y &= v \end{aligned} \tag{2}$$

where y is the voltage output and u_i is the current input. As expected, one state has been lost due to the differential causality. When the input current u_i is given it is possible to solve the differential causality algebraically, atleast if the models are linear. However, for a generic distributed model that doesn't have insight in the dynamics of its connected models it is difficult to solve the differential causality algebraically. It is important to stress that such a derivative term, $\frac{d}{dt}u_i$, requires suited solvers but additional simulation errors will be introduced. Hence, if the differential term can be replaced by something that completes the state space formulation, for example an estimate of the derivative term, simple solvers can be used. Then the model is not dependent on an advanced solver or knowing the algebraic solution for the differential causality problem.

3.2. Solution of the Differential Causality Problem

A low-pass filter with differential properties can be used to estimate the differential term and is given as the transfer function

$$H_f(s) = \frac{s}{Ts+1} \tag{3}$$

Transforming this transfer function in to the time domain results in

$$\begin{aligned} \dot{x}_f &= \frac{1}{T}(u - x_f) \\ y_f &= \frac{1}{T}(u - x_f) \end{aligned} \tag{4}$$

where x_f is the new state that completes the state space model and y_f is the estimate of $\frac{d}{dt}u$. By substituting (4) into (2) we obtain a complete state space model expressed as

$$\begin{aligned} \dot{\omega} &= \frac{1}{J}(\tau - b\omega - ru_i) \\ \dot{x}_f &= \frac{1}{T}(u_i - x_f) \\ y_d &= -(\frac{L}{T} + R)u_i + \frac{L}{T}x_f + r\omega \end{aligned} \tag{5}$$

where y_d is the output voltage. It can been shown that the filter time constant T can be chosen such that the errors introduced by the filtering properties are small and in most cases negligible. For stability analysis and simulation purposes the parameters used in the genset model are given in table 2. Before

Table 2. Genset parameters.

Symbol	Value
J	$10\,kgm^2$
b	$0.1\,{}^{Nm}/_{s}$
r	$10\,NmHs$
L	$0.05\,H$
R	$0.01\,\Omega$
τ	$200\,Nm$
T	$0.001\,s$

performing simulation of the genset model it is important to discuss stability in the sense of hybrid causality models and is treated in the following.

3.3. Stability of Hybrid Causality Models

Stability of hybrid causality models for continuous systems can be studied by analysing the dynamics in both the solver and the system for all causality orientations. In addition, one must assure that the power is conserved during discrete time events such as when switching causality. Stability for distributed hybrid causality models are left out here, but will be studied in section 4..

Starting with the complete integral causality model, the dynamics can be reformulated as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u_v + \mathbf{C} \tag{6}$$

where $\mathbf{x} = [\omega,\ i]^T$ is the state vector, $\mathbf{B}$ is the input mapping matrix, $\mathbf{C}$ is a vector including the constant engine torque τ and

$$\mathbf{A} = \begin{bmatrix} -\frac{b}{J} & -\frac{r}{J} \\ \frac{r}{L} & -\frac{R}{L} \end{bmatrix} \tag{7}$$

It can be verified that the eigenvalues of $\mathbf{A}$ are $\lambda = -0.105 \pm 14.1418i$ for the parameters used in table 2. Also, by specifying that $u_V = R_l i$, where $R_l = 10\,\Omega$ is the electrical load, removes the $\mathbf{B}$-matrix and the $\mathbf{A}$-matrix can be rewritten as

$$\mathbf{A} = \begin{bmatrix} -\frac{b}{J} & -\frac{r}{J} \\ \frac{r}{L} & -\frac{R+R_l}{L} \end{bmatrix} \tag{8}$$

The corresponding eigenvalues for the $\mathbf{A}$-matrix is now given as $\lambda_1 = -1.0141$ and $\lambda_2 = -199.1959$. If the Euler integration method is used for solving the system in a simulation the stability criterion for a stable solution is given as

$$\Delta t \leq \frac{2}{\lambda_i},\ \forall i \tag{9}$$

whenever the eigenvalues are real, where Δt is the solver time step. The same analysis can also be done for the reformulated differential causality. Setting $\mathbf{x} = [\omega,\ x_f]^T$ gives

$$\mathbf{A}_d = \begin{bmatrix} -\frac{b}{J} & 0 \\ 0 & -\frac{1}{T} \end{bmatrix} \tag{10}$$

and the eigenvalues $\lambda_1 = -0.01$ and $\lambda_2 = -1000.0$. Note that the subscript $_d$ is used to separate the complete integral causality model and the reformulated differential causality model. If also in this case $u_i = \frac{y}{R_l}$ it can be shown that the $\mathbf{B}_d$-matrix disappear and the $\mathbf{A}_d$-matrix is rewritten as

$$\mathbf{A}_d = \begin{bmatrix} -\frac{b}{J} - \frac{r^2}{JR_l\left(\frac{R}{R_l}+\frac{L}{R_lT}+1\right)} & -\frac{Lr}{JR_lT\left(\frac{R}{R_l}+\frac{L}{R_lT}+1\right)} \\ \frac{r}{R_lT\left(\frac{R}{R_l}+\frac{L}{R_lT}+1\right)} & \frac{L}{R_lT^2\left(\frac{R}{R_l}+\frac{L}{R_lT}+1\right)} - \frac{1}{T} \end{bmatrix} \tag{11}$$

which gives the eigenvalues $\lambda_1 = -1.014$ and $\lambda_2 = -165.968$. For both these causality models $\Delta t = 0.001$ would give stable results when using Euler integration and is chosen as the solver time step.

Next, we need to assure that the power is conserved during a causality change if the model is to switch causality online during a simulation. This can be assured by choosing the initial conditions correctly.

3.4. Initial Conditions and Simulation

If the power is to be conserved during causality switching in a hybrid causality model, the values of the model inputs and outputs right before and after a causality change must be conserved. When switching causality from complete differential causality to reformulated differential causality the

shaft speed would remain the same, but the initial conditions for the new state would be given as

$$x_f = \frac{T}{L}\left[u_V + \left(\frac{L}{T} + R\right) i - r\omega\right] \tag{12}$$

where i, ω and u_V are the last obtained or received values before the discrete event. It can be verified that this initial condition gives $y_d = u_V$ by inserting (12) into y_d given in (5). The same idea can be used to find initial conditions for i when switching from the reformulated differential causality model to the complete integral causality model. Hence,

$$i = \frac{y_d}{R_l} \tag{13}$$

which gives $i = u_i$.

The hybrid causality genset model is implemented, together with the simple load models, and simulated with the parameters presented above. The total simulation time is set to 5 s and the global distributed communication time step, T_d, is set equal to the solver time step. This is the same as not distributing the system since the same solver is used, with the same time steps, and data is exchanged after each solved time step. The simulation starts with the complete integral causality model, switches to the reformulated differential causality model at t=1 s before returning to the complete integral causality model at t=3 s. The simulation results are shown in figure 7 and 8.

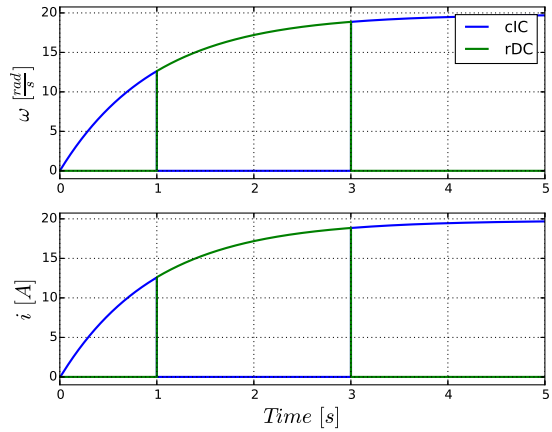

Figure 7. Simulation of shaft speed and current.

In figure 7 cIC and rDC are abbreviations for the complete integral causality model and the reformulated differential causality model, respectively. As can be seen in the figure both the shaft speed and the current seem to be continuous throughout the whole simulation and is verified by figure 8, which shows the voltage and current from the genset model's input and output ports. Both u_V and u_i are continuous and u_i is identical to the results obtained in figure 7.

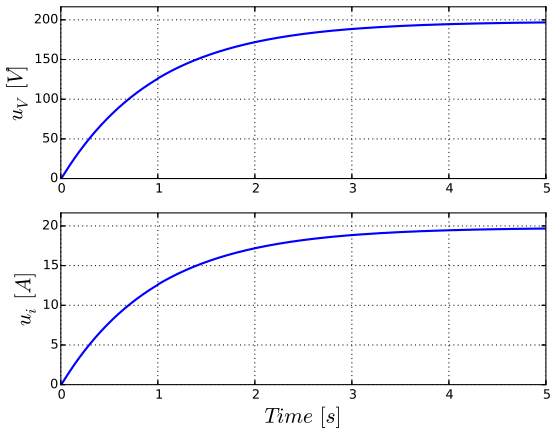

Figure 8. Simulation results of load voltage and current.

It should be mentioned that the filter time constant T affects the dynamics of the genset and should be set small in order to minimize the changes. However, since the reformulated differential causality model is initialized after a causality switch with power conserving initial conditions, it will not affect the results in discrete time, only slightly the results in continuous time when T is kept relatively small.

Next, it would be of interests to distribute the genset model and the loading model. This is done by implementing the genset model as one submodel, the loading models as one submodel and connect the models together in a distributed simulation where $T_d \geq \Delta t$. This requires a through stability analysis of the total distributed system in order to guarantee total system- and simulation stability.

4. STABILITY IN DISTRIBUTED SYSTEMS

A stability analysis of a simulation model involves in general two different stability analysis, namely dynamic stability analysis and solver stability analysis. Dynamic stability analysis for continuous systems or simulation models are covered in [8], [5] for nonlinear systems and in [4] for linear systems. In general, dynamic system stability can be analysed by the use of Lyapunov stability theory, where a positive definite storage function describing the energy in the system is used as a lyapunov function candidate in the analysis. However, the stability of a distributed system simulation is more complex than a continuous system because the solver stability and the dynamic stability are more close and more or less dependent of each other. However, if each distributed subsystem has certain *passivity* properties, dynamic stability and solver stability can be analysed separately as often done for continuous systems.

In this paper stability of linear distributed systems solved by fixed step solvers will be studied. This involves both the

system dynamics and the solver dynamics, and, as will be shown, a new stability criterion can be found for such systems without considering passivity properties.

4.1. Stability of Linear Distributed Systems with Fixed Step Solvers

Consider a linear system of differential equations describing a distributed submodel i given as

$$\begin{aligned} \dot{\mathbf{x}}_i &= \mathbf{A}_i\mathbf{x}_i + \mathbf{B}_i\mathbf{u}_i \\ \mathbf{y}_i &= \mathbf{H}_i\mathbf{x}_i \end{aligned} \tag{14}$$

where $\mathbf{x}_i$ is the state vector, $\mathbf{u}_i$ is the input vector, $\mathbf{y}_i$ is the output vector and the matrices $\mathbf{A}_i$, $\mathbf{B}_i$ and $\mathbf{H}_i$ are given by the system dynamics. If the Euler integration method is used to solve this subsystem with the time step Δt_i, the results for the next solver time step can be expressed as a function of the previous results,

$$\begin{aligned} \mathbf{x}_i(t_0+\Delta t) &= \mathbf{x}_i(t_0) + \Delta t_i[\mathbf{A}_i\mathbf{x}_i(t_0) + \mathbf{B}_i\mathbf{u}_i(t_0)] \\ &= (\mathbf{I} + \Delta t_i\mathbf{A}_i)\mathbf{x}_i(t_0) + \Delta t_i\mathbf{B}_i\mathbf{u}_i(t_0) \end{aligned} \tag{15}$$

where $\mathbf{I}$ is the identity matrix. By assuming that the model inputs and outputs are held constant between each global time step T_d, chosen such that

$$n_i = \frac{T_d}{\Delta t_i} \tag{16}$$

where n_i is a positive integer for all subsystems i in the distributed system, the next simulation step can be expressed as

$$\begin{aligned} \mathbf{x}_i(t_0+2\Delta t) &= \mathbf{x}_i(t_0+\Delta t) + \Delta t_i[\mathbf{A}_i\mathbf{x}_i(t_0+\Delta t) + \mathbf{B}_i\mathbf{u}_i(t_0)] \\ &= (\mathbf{I} + \Delta t_i\mathbf{A}_i)^2\mathbf{x}_i(t_0) + [(\mathbf{I} + \Delta t_i\mathbf{A}_i) + \mathbf{I}]\Delta t_i\mathbf{B}_i\mathbf{u}_i(t_0) \end{aligned} \tag{17}$$

When n_i steps have been taken, the global time is given as $t_0 + T_d$ and the solution can be expressed as

$$\mathbf{x}_i(t_0+T_d) = (\mathbf{I}+\Delta t_i\mathbf{A}_i)^{n_i}\mathbf{x}_i(t_0) + \sum_{j=1}^{n_i}(\mathbf{I}+\Delta t_i\mathbf{A}_i)^{j-1}\Delta t_i\mathbf{B}_i\mathbf{u}_i(t_0) \tag{18}$$

because $\mathbf{u}(t_0) = \mathbf{u}(t_0+\Delta t) = ... = \mathbf{u}(t_0+T_d)$. Further, if $\mathbf{A}_i$ is nonsingular the sum can be rewritten as

$$\sum_{j=1}^{n_i}(\mathbf{I}+\Delta t_i\mathbf{A}_i)^{j-1} = \frac{1}{\Delta t_i}\mathbf{A}_i^{-1}[(\mathbf{I}+\Delta t_i\mathbf{A}_i)^{n_i} - \mathbf{I}] \tag{19}$$

and thus, the solution for each global time step T_d can be expressed as

$$\mathbf{x}_i(t_0+T_d) = \mathbf{A}_{n_i}\mathbf{x}_i(t_0) + \mathbf{B}_{n_1}\mathbf{u}_i(t_0) \tag{20}$$

where

$$\begin{aligned} \mathbf{A}_{n_i} &= (\mathbf{I}+\Delta t_i\mathbf{A}_i)^{n_i} \\ \mathbf{B}_{n_i} &= \mathbf{A}_i^{-1}(\mathbf{A}_{n_i} - \mathbf{I})\mathbf{B}_i \end{aligned} \tag{21}$$

By assuming that a distributed system has two submodels with the input-output mappings given as

$$\begin{aligned} \mathbf{u}_1 &= \mathbf{H}_2\mathbf{x}_2 \\ \mathbf{u}_2 &= \mathbf{H}_1\mathbf{x}_1 \end{aligned} \tag{22}$$

the solution of the total distributed system can be written as

$$\mathbf{x}_d(t_0+T_d) = \mathbf{S}_d\mathbf{x}_d \tag{23}$$

where $\mathbf{x}_d = [\mathbf{x}_1,\ \mathbf{x}_2]^T$ is the total distributed system state vector and $\mathbf{S}_d$ is denoted the total distributed system solution matrix given as

$$\mathbf{S}_d = \begin{bmatrix} \mathbf{A}_{n_1} & \mathbf{B}_{n_1}\mathbf{H}_2 \\ \mathbf{B}_{n_2}\mathbf{H}_1 & \mathbf{A}_{n_2} \end{bmatrix} \tag{24}$$

The stability for a linear distributed system can be assured if the magnitude of the eigenvalues of $\mathbf{S}_d$ are less than 1,

$$|eig(\mathbf{S}_d)| < 1 \tag{25}$$

if $n_i \in \mathcal{N}_{\geq 1}\ \forall i$ where $\mathcal{N}_{\geq 1}$ is all integers larger than 1. This last condition assures global time synchronization between the submodels. Also, the distributed system is said to be only marginally stable if atleast one of the eigenvalues has a magnitude equal to 1.

These stability results can be applied to the genset model for both causality orientations and if the condition in (25) holds, the distributed hybrid causality model of the genset connected to the load model will be stable. This, because stability during switching has already been assured through the choice of initial conditions.

4.2. Stability of Distributed Genset Simulation

Starting with the complete integral causality model, the system dynamics can be rewritten as

$$\begin{aligned} \dot{\mathbf{x}}_1 &= \mathbf{A}_1\mathbf{x}_1 + \mathbf{B}_1 u_V + \mathbf{C}_1 \\ y_1 &= \mathbf{H}_1\mathbf{x}_1 \end{aligned} \tag{26}$$

where subscript $_1$ refers to the complete integral causality model, $\mathbf{x}_1 = [\omega,\ i]^T$ is the state vector,y_1 is the output from the genset model and

$$\begin{aligned} \mathbf{A}_1 &= \begin{bmatrix} -\frac{b}{J} & -\frac{r}{J} \\ \frac{r}{L} & -\frac{R}{L} \end{bmatrix}, \quad \mathbf{B}_1 = \begin{bmatrix} 0 \\ -\frac{1}{L} \end{bmatrix} \\ \mathbf{C}_1 &= \begin{bmatrix} \frac{\tau}{J} \\ 0 \end{bmatrix}, \qquad \mathbf{H}_1 = \begin{bmatrix} 0 & 1 \end{bmatrix} \end{aligned} \tag{27}$$

Next, the input from the load model is given as $u_V = R_l i$, which means that we might write

$$\mathbf{x}_1(t_0 + T_d) = \mathbf{S}_d \mathbf{x}_1(t_0) + \mathbf{C}_{n_1} \tag{28}$$

where

$$\mathbf{S}_d = \mathbf{A}_{n_1} + \mathbf{B}_{n_1} \mathbf{H}_1 R_l \tag{29}$$

$\mathbf{A}_{n_1}$ and $\mathbf{B}_{n_1}$ is given as in (21) and

$$\mathbf{C}_{n_1} = \mathbf{A}_1^{-1}(\mathbf{A}_{n_1} - \mathbf{I})\mathbf{C}_1 \tag{30}$$

The $\mathbf{C}_{n_1}$ matrix gives only a bias to the total solution and is not relevant in the stability analysis. When the solver time step is set to Δt =0.001 s it is possible to plot the magnitude of the eigenvalues in $\mathbf{S}_d$ as a function of n_i and the results are shown in figure 9.

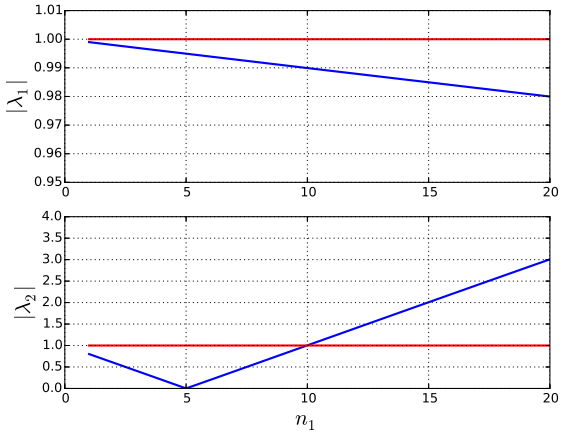

Figure 9. Magnitude of eigenvalues in the $\mathbf{S}_d$-matrix as a function of n_1.

As can be seen in the figure, $|\lambda_2| > 1$ when $n_1 \geq 10$, which means that T_d =0.009 s is the largest stable global time step for the chosen solver time step. This is illustrated in figure 10 where the first plot is the simulation results when T_d =0.009 and the second plot is the results with T_d =0.01 s.

The results show clearly that the model is not stable when T_d =0.01 s, which proves the stability analysis of the total distributed system.

The reformulated differential causality model can be expressed as

$$\begin{aligned} \dot{\mathbf{x}}_2 &= \mathbf{A}_2 \mathbf{x}_2 + \mathbf{B}_2 u_i + \mathbf{C}_2 \\ y_2 &= \mathbf{H}_2 \mathbf{x}_2 + K_2 u_i \end{aligned} \tag{31}$$

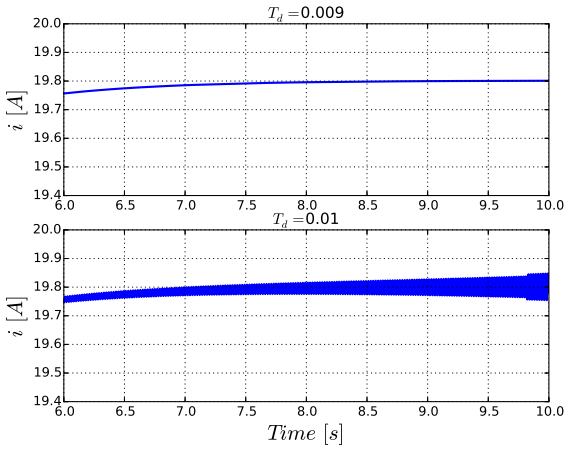

Figure 10. Comparison of distributed simulation results for the complete integral causality model with different T_d.

where $\mathbf{x}_2 = [\omega,\ x_f]^T$, y_2 is the model output and

$$\begin{aligned} \mathbf{A}_2 &= \begin{bmatrix} -\frac{b}{J} & 0 \\ 0 & -\frac{1}{T} \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} -\frac{r}{J} \\ \frac{1}{T} \end{bmatrix} \\ \mathbf{C}_2 &= \begin{bmatrix} \frac{\tau}{J} \\ 0 \end{bmatrix}, \qquad\qquad \mathbf{H}_2 = \begin{bmatrix} r & \frac{L}{T} \end{bmatrix} \\ K_2 &= -\frac{L}{T} - R \end{aligned} \tag{32}$$

In this case $u_i = \frac{y}{R_l}$ which means the total solution of the distributed system can be expressed as

$$\mathbf{x}_2(t_0 + T_d) = \mathbf{S}_d \mathbf{x}_2(t_0) + \mathbf{C}_{n_2} \tag{33}$$

where

$$\mathbf{S}_d = \mathbf{A}_{n_2} + \frac{1}{R_l + K_2} \mathbf{B}_{n_2} \mathbf{H}_2 \tag{34}$$

and where $\mathbf{A}_{n_2}$ and $\mathbf{B}_{n_2}$ is given as in (21) and

$$\mathbf{C}_{n_2} = \mathbf{A}_2^{-1}(\mathbf{A}_{n_2} - \mathbf{I})\mathbf{C}_2 \tag{35}$$

Figure 11 shows the magnitude of the eigenvalues of $\mathbf{S}_d$ when the solution matrix is a function of n_2.

As can be seen in the figure the reformulated differential causality model seems to be stable for any value of n_2, which means that the complete integral causality model will be the restrictive model when simulating the hybrid causality model in a distributed setting with the load model. The results from a distributed simulation of the total system, including causality switching as in figure 8, are given in figure 12 and 13 when Δt =0.001 s and T_d =0.009 s. The total simulation time is set to t =10 s.

As can be seen in the figures the simulation results look identical with the ones given in figure 7 and 8, which means that the hybrid causality genset model works quite well also in a distributed setting when connected to the given load models.

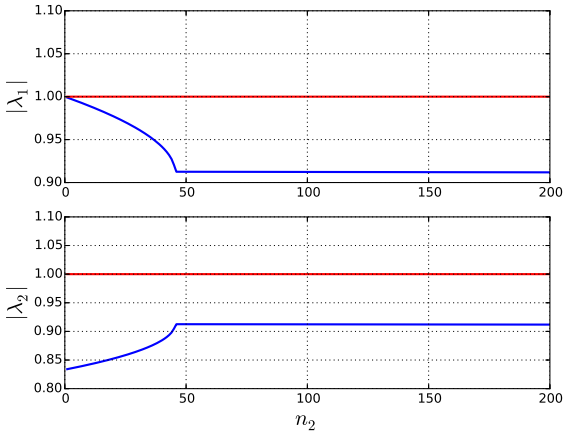

Figure 11. Magnitude of eigenvalues in the $\mathbf{S}_d$-matrix as a function of n_2.

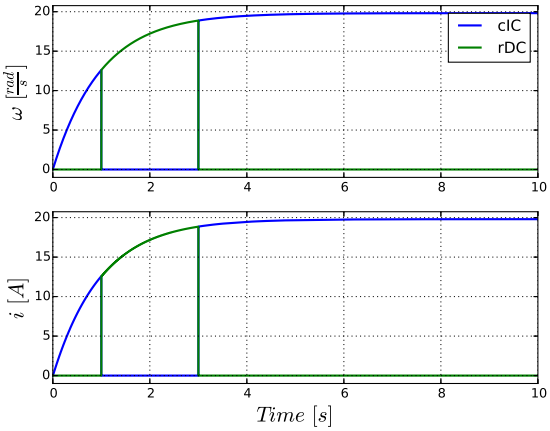

Figure 12. Simulation of shaft speed and current.

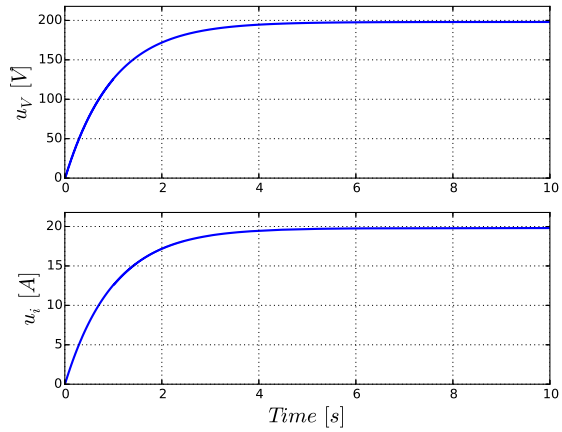

Figure 13. Simulation results of load voltage and current.

5. CONCLUSION

In this work linear distributed systems have been studied by the use of bond graph theory, with the main focus on connectivity, hybrid causality and stability. The reason for studying these topics is because connectivity and stability are important topics in distributed systems. Hybrid causality models, and switched models in general, are state of the art in the field of mathematical modelling and closely related to the bond graph theory.

The bond graph theory provides a clear relation between connectivity, causality and model fidelity, which is quite helpful when making distributed mathematical models. As it turns out, connectivity is only dependent on the causality for a given energy domain when sticking to the bond graph modelling theory. This helps establishing a standard for connecting distributed models in a given energy domain, through the power variables. Moreover, when considering hybrid causality models, the criteria for connecting submodels is reduced to only assuring that the energy domains are equal.

Hybrid causality models opens for modelling more complex systems with discrete time events, such as opening or closing of switches in power electronics. In this work the differential causality part of the hybrid causality model has been reformulated to integral causality in order to complete the state space model and enable the use of simple solvers in simulations. However, stability of such systems must be analysed, both for the solver dynamics and the system dynamics, in order to guarantee stable solutions in simulations. Stability during causality switching can be established by choosing the initial conditions carefully.

Stability of linear distributed systems can be analysed by including both the solver dynamics and the system dynamics in the same analysis, as have been done in this paper. This results in a criterion for choosing the local solver time steps and the global synchronization time step. However, only the Euler integration method has been considered in this work. The solution matrix can also be derived for both Runge Kutta 2 and 4 integration methods as well as for the Euler integration method.

Simulations for a simple distributed hybrid causality genset model connected to a distributed electrical load has been performed and the results verifies the stability criterion. Also, the results show that the model outputs are continuous which indicates that the reformulation of the differential causality gives good results.

ACKNOWLEDGEMENTS

Thanks to the partners in the ViProMa project, Norwegian Research Counsel project number 225322, for providing financial support for studying distributed systems for marine systems and operations.

REFERENCES

[1] Dean C. Karnopp, Donald L. Margolis, and Ronald C. Rosenberg. System Dynamics: Modeling and Simulation of Mechatronic Systems. John Wiley & Sons, Inc., New York, NY, USA, 2006.

[2] W. Borutzky. Bond graph modeling from an object oriented modeling point of view. Simulation Practice and Theory, 7(5-6):439 – 461, 1999.

[3] T. Blochwitz, M. Otter, M. Arnold, C. Bausch, C. Clauß, H. Elmqvist, A. Junghanns, J. Mauss, M. Monteiro, T. Neidhold, D. Neumerkel, H. Olsson, J. v. Peetz, S. Wolf, Atego Systems Gmbh, Qtronic Berlin, Fraunhofer Scai, and St. Augustin. The functional mockup interface for tool independent exchange of simulation models. In In Proceedings of the 8th International Modelica Conference, 2011.

[4] Chi-Tsong Chen. Linear System Theory and Design. Oxford University Press, Inc., New York, NY, USA, 3rd edition, 1998.

[5] H.K. Khalil. Nonlinear Systems. Pearson Education. Prentice Hall, 2002.

[6] R. Goebel and R.G. Sanfelice. Hybrid Dynamical Systems: Modeling, Stability, and Robustness. Princeton University Press, 2012.

[7] A.C. Umarikar and L. Umanand. Modelling of switching systems in bond graphs using the concept of switched power junctions. Journal of the Franklin Institute, 2005.

[8] Francois E. Cellier and Ernesto Kofman. Continuous System Simulation. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.