ON NOISE MODELLING IN PHYSICAL MODELLING APPROACH

Dorel AIORDACHIOAIE

"Dunarea de Jos" University of Galati, Electrical Engineering Faculty, Domneasca-47, Galati – 800008, Romania, Email: Dorel.Aiordachioaie@ugal.ro

Abstract: Physical modelling is referred as the first representation of a process model and it is represented as a set of differential and algebraic equations. Noise added to model can improve the estimated behaviour of the process. Adding white noise to all variables is not recommended mainly because there could be variables based on the derivative of the white noise, and this is computational infeasible and physical impossible. The studied problem is to decide where is allowed to add noise from physical perspective early, at the modelling stage, and thus to avoid further numerical problems at the stage of simulation.

Keywords: Process Modelling, Object-Oriented Modelling Languages, Neutral Modelling, Noise Modelling, DAE models.

1. INTRODUCTION

Physical based and object-oriented modelling languages offer an interesting and useful approach in process modelling and simulation, very appreciated and useful in the world of engineers and scientists. Examples of such software based environment are Omola, Dymola and MathModelica. All these environments are connected with basic features of the Modelica modelling language, as a neutral representation of processes. More, based on object-matching features, it is used as standard representation formalism over the distributed simulation platforms.

Perhaps the first references which emphasis a strong call to new modelling principles is of (Cellier, 1991), which clearly shows the constraints of pure mathematical models. Other example could be of (Astrom, *et al*, 1998) and (VisiMod, 2004), a project under the resources of Foundation for Strategic Research of Sweden.

It is like the study is part of the global effort – as presented above – in order to solve some problems in the process modelling activity based on physical decomposition of the process and based on object-oriented technology. More, some problems of noise modelling in the equation-based modelling are considered to have an equal importance.

In section 2 the main features and problems of modelling following the physical decomposition of the process are considered. The object of the section 3 is related to methods of noise modelling. Some basic features of the Modelica modelling language are presented also. Section 4 contains the main contribution of the paper, looking to define the problem, to understand the method and to propose solutions related to the noise modelling. Naturally, there should be one more section, as a case study. Unfortunately, until now, some symbolic computational problems were not possible to be solved.

2. PHYSICAL PROCESS MODELLING

It is accepted that three basic features describe the principles of physical modelling:
1). the model is obtained by connecting sub-models which parallels the physical construction;
2). the equations are used as obtained, i.e. without manually manipulate the equations;
3). the variables are physic variables and not abstract quantities like in mathematical models.

By applying first modelling principles, a set of equations are obtained, which could be organized in two subsets: a subset of differential equations describing the dynamics of the process and a subset of algebraic equations describing the outputs and the constraints of the behaviour of the considered process. Such models generally respect a differential algebraic equation structure. In the linear case it can be written as

$$\mathbf{E}\cdot\dot{\mathbf{x}}(t)+\mathbf{F}\cdot\mathbf{x}(t)=\mathbf{B_u}\cdot\mathbf{u}(t) \qquad (1.a)$$

where $\mathbf{x}(t)$ is the state variable vector, $\mathbf{u}(t)$ is the input variable vector and $\mathbf{E},\mathbf{F},\mathbf{B_u}$ are matrices of appropriate dimensions. Obviously, the matrix $\mathbf{E}$ is supposed singular and the reason is that purely algebraic equations are present. There is a strong desire to reformat the equation models in some well known structures, as the state-space form is. Such transformations are obtained by properly use of specialized software tools.

In the context of real experiments where measurements should be considered as well for the purpose of identification and parameter estimation, the model (1.a) is improved with a measurement equation

$$\mathbf{y}(t)=\mathbf{C}\cdot\mathbf{x}(t)+\mathbf{e}(t) \qquad (1.b)$$

where $\mathbf{y}(t)$ is the measurement variable vector and $\mathbf{e}(t)$ is the sensor noise vector.

The name of the model representation (1) is also *implicit system*, *descriptor system* or *generalized system* (Schon, *et al*, 2003).

The reason to introduce noise in process's model is mainly related to un-modelled dynamics and disturbances acting on the process. Considering noise, the state-equations of the model of the process could have the form

$$\mathbf{E}\cdot\dot{\mathbf{x}}(t)+\mathbf{F}\cdot\mathbf{x}(t)=\mathbf{B_u}\cdot\mathbf{u}(t)+\mathbf{B_w}\cdot\mathbf{w}(t) \qquad (2.a)$$

$$\mathbf{y}(t)=\mathbf{C}\cdot\mathbf{x}(t)+\mathbf{e}(t) \qquad (2.b)$$

where a noise component was added for state variables. The type of the noise regarding to the power and to the probability density function depends on the process. Usually, white noise is considered with variance connected with the dynamics of the process.

2.1. Object-Oriented Modelling

The term object-oriented means the equations describing a commonly used system can be packaged, stored and reused, (Gerdin and Schon, 2004).

Object-oriented languages have the advantage of being equation-based, which means that the user can enter the equations describing the process without having to transform them into ODE form. The equations will be in differential-algebraic equation (DAE) form, which means that modelling environments must be able to simulate such models.

2.2. About Modelica

Modelica is a general equation-based object-oriented language for continuous and discrete-event modelling of physical systems for the purpose of efficient simulation.

The language unifies and generalizes previous object-oriented modelling languages, e.g., Omola, Smile or Dymola. The language has been designed to facilitate exchange of models, model libraries and simulation specifications. The reader is referred to (Tiller, 2001), (Fritson, 2004) (Modelica, 2004) or (Elmqvist, *et al*, 1999), for a complete description of the language and its functionality from the perspective of the motivations and design goals.

Modelica is being considered also as a standard for a model representation in the Global CAPE-OPEN project, (GCO, 2000), which aims at developing standard interfaces for simulation components, (Braunschwig, *et al*, 1999). In this paper, Modelica will be used also as working language based on the trend and reality which defines Modelica a *de facto* standard in physical modelling and object-oriented modelling technologies.

3. NOISE METAMODELLING

Figs 1 and 2 present the methodology of noise modelling in the context of physical modelling, i.e. the integration on noise models into physical process models.

The class diagram of Fig. 1 shows the hierarchy of different models, which are used in the building of the system model with physical constraints.

A model is a generalisation of a process model. A process model is an aggregation of one or more models based on equations. An equation based model is an aggregation of some model, part with noise and part without noise. A noise model should be compliant with the laws of physics and the variables involved in the model should have a physical meaning.

Fig.2 says that a noise model could be generalized as a model. A noise model is a generalization of white noise model and band-limited noise, i.e. coloured noise. The last one has constraints from a physical model concerning the parameters like, e.g. the power of the noise.

Noise modelling is an important task in process modelling. There are unmodelled phenomena and unknown parameters. A noise model should describe the how the unmeasured inputs and the unmodelled dynamics could change the behaviour of the considered system.

Noise modelling means also the addition of one or more noise components to state variables, in order to model disturbances and or some random or unknown behaviour.

.

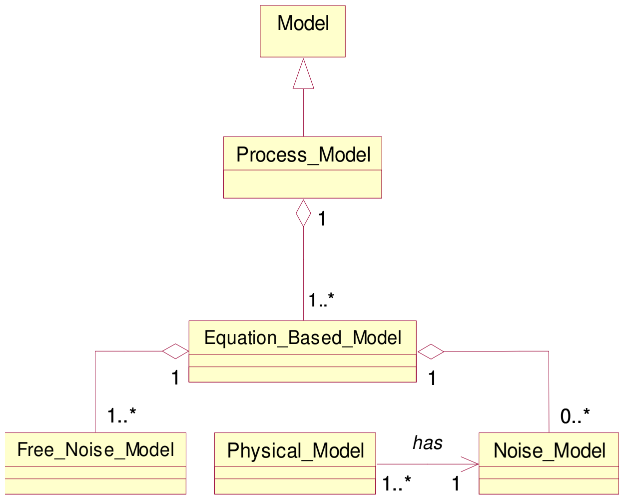

Fig. 1-Different types of models

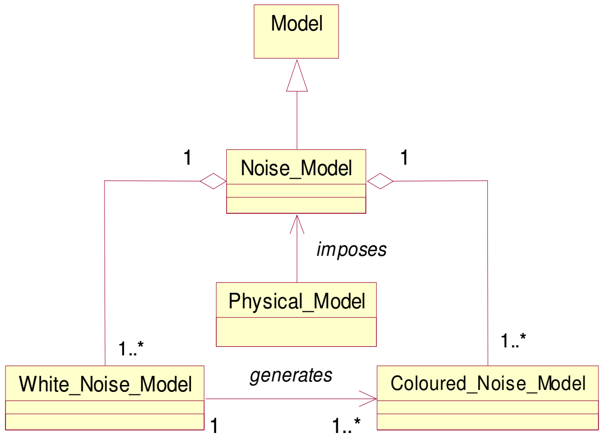

Fig. 2 -Connections among different types of noise models

Table 1 - Conversion example from declarative to state form representation

Declarative physical model under Modelica	*Declarative state model*
model motor_sim	*/* DSblock model generated by Dymola from*
..	*Modelica model motor_sim */*
parameter …	...
Real JL, JT, omega;	**TranslatedEquations**
Real Tm, alfa, e, u, i;	JL_0 = 0.5 * M_0 * rd_0 * rd_0;
…	JT_0 = Jm_0 + JL_0;
equation	...
u = u0;	**DynamicSection**
e = Kb * omega;	der_i = - divmacro(... ,"i*R+e-u", ...);
u = i * R + L * der(i) + e;	der_omega = - divmacro(... ,"b*omega-Tm", ...);
Tm = JT * alfa + b * omega;	u_0 = u0_0;
alfa = der(omega);	e_0 = Kb_0 * omega_0;
JL = 0.5 * M * rd * rd;	Tm_0 = Kt_0 * i_0;
JT = Jm + JL;	
Tm = Kt * i;	...
end motor_sim;	

The model (2) contains a noise component and the problem is to decide where to put the noise components in the considered process.

Naturally, the physics of the process should indicate which variables should have noise and which one not, which could be done manually for simple processes. The problem has two aspects: there are complex processes, difficult to mange; a software tool is more efficient and comfortable for any modeller.

Adding noise to all equations can lead to derivates of white noise and – as results – to non-causal process, as infinite values of some variables. Details on how the non-causality with respect to the input signal, $\mathbf{u}(t)$, can be handled are in (Gerdin, 2004). The problem itself is considered and solved, however by (Campbell, 1990), where it is suggested a band limited noise to avoid the problems. Also, (Varga, 1992) made an interesting analysis of descriptors systems, from the point of view of preliminary pre-processing steps in controllers and observers synthesis algorithms, e.g., by similarity transformations and model reductions

Attempts to introduce white noise in the continuous model (2) has been done, e.g., in (Schein and Denk, 1998) and (Winkler, 2004). (Schon, *et al*, 2003) derive a basis for the subspace of all possible causal disturbances. The basis is taken as $\mathbf{B_w}$ in (2) and the noise covariance matrix is used as the design variable to rotate and scale the basis. Mainly, there is a sorting procedure in two sets of state variables, with a set of variables accepting arbitrarily white noise.

The DAE model with noise will have the form

$$\mathbf{E}\cdot\dot{\xi}(t)=\mathbf{J}\cdot\xi(t)+\mathbf{K}_1\cdot\mathbf{u}(t)+\mathbf{K}_2\cdot v_1(t) \qquad (3.a)$$

$$\mathbf{y}(t) = \mathbf{L} \cdot \xi(t) + v_2(t) \quad (3.b)$$

where $v_1(t)$ represents the unmeasured inputs and $v_2(t)$ represents the transducers' noise. Both signals are continuous white noises.

The transformation of the model (3) in a state form could generate state variables that are functions of the derivative of white noise, and then the simulation is not possible or needs some special solvers which could – in turn – generate false behaviours. From physical point of view, variables depending on the derivative of the white noise do not correspond to any physical reality, i.e. infinite values of physical variables.

The driving idea is to define constraints of the matrix $\mathbf{K}_2$, so at the level of the raw physical model, in such a way to be able to decide where the noise generators are feasible and where are not. Thus, the problem of noise derivative is solved at the level of modelling and the problems at the level of simulation could be avoided.

The definition of the problem and two solutions are from (Schon, *et al*, 2003) and (Gerdin, 2004), where two solutions based on time and frequency methods are presented. In the next section the frequency method is presented only.

4. DESCRIPTION OF THE METHOD

The method is based on the matrix transfer from noise to internal variables, which should be proper. In the opposite case, the degree of the numerator is grater then the degree of numerator and a non-causal representation is obtained.

Let the transfer matrix from noise to internal variables as

$$\mathbf{G}(s) = (s\mathbf{E} - \mathbf{J})^{-1} \cdot \mathbf{K}_2 \quad (4)$$

This should be proper. Following, e.g., (Rough, 1996) a rational matrix can be transformed into a row reduced form by a pre-multiplication with a unimodular matrix thus

$$\mathbf{D}(s) = \mathbf{U}(s) \cdot (s\mathbf{E} - \mathbf{J}) \quad (5)$$

where

$$\mathbf{N}(s) = \mathbf{U}(s) \cdot \mathbf{K}_2 \quad (6)$$

then

$$\mathbf{D}^{-1}(s) \cdot \mathbf{N}(s) = (s\mathbf{E} - \mathbf{J})^{-1} \cdot \mathbf{K}_2 = \mathbf{G}(s) \quad (7)$$

where $\mathbf{D}(s)$ is the row reduced form of the matrix $(s\mathbf{E} - \mathbf{J})$, for a certain unimodular matrix $\mathbf{U}(s)$.

The condition to have a proper transfer matrix function is that the matrix

$$\mathbf{D}^{-1}(s) \cdot \mathbf{N}(s) \quad (8)$$

is proper. Following the theorem of Appendix B, the matrix (8) is proper if and only if the highest degree of the polynomials in each row in $\mathbf{N}(s)$ is lower than or equal to the highest degree of the polynomials in the corresponding row of $\mathbf{D}(s)$. The rows indexes for which the above condition does not hold show which variables should not have added noise in the model (3).

4.1. The algorithm

1). Write the DAE model of the system/process in the form (2).

2). Compute the transfer matrix $\mathbf{G}(s)$ from noise to internal variables.

3). Compute the row reduced form $\mathbf{D}(s)$ and storage the unimodular matrix $\mathbf{U}(s)$.

4). Decompose the matrix $\mathbf{U}(s)$ following the formula:

$$\mathbf{U}(s) = \sum_{i=0}^{n} \mathbf{U}_i \cdot s^i \quad (9)$$

5). Compute the degrees of the rows of the matrix $\mathbf{D}(s)$.

6). Find the values of the index *i* for which the inequality

$$\mathbf{u}_{ij} \cdot \mathbf{K}_2 = 0, \quad i > r_j[\mathbf{D}],\ j = 1,2,..n \quad (10)$$

does not hold. The obtained set values sets the index variables of the internal variables vector for which a noise in not allowed to add.

4.2 An example (Schon, *et al*, 2003)

Let a DAE model without noise as

$$J_1 \cdot \dot{x}_1 = M_1 + M_2$$
$$J_2 \cdot \dot{x}_2 = M_3 + M_4$$
$$M_2 = -M_3$$
$$x_1 = x_2$$

With c_i arbitrary constants, let the unimodular matrix as

$$\mathbf{U}(s) = \begin{bmatrix} c_1 & c_2 & 0 & s \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \mathbf{U}_0 + \mathbf{U}_1 \cdot s$$

then

$$\mathbf{D}(s) = \begin{bmatrix} 0 & 0 & c_3 & c_4 \\ 0 & c_5 \cdot s & 0 & -1 \\ 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 \end{bmatrix}$$

The row degrees of $\mathbf{D}(s)$ are

$$deg[\mathbf{D}(s)] = [0 \quad 1 \quad 0 \quad 0]^{\mathrm{T}}$$

The transfer matrix function is proper if

$$\mathbf{U}_1 \cdot \mathbf{K}_2 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \mathbf{K}_2 = \mathbf{0}$$

which means that the last raw of $\mathbf{K}_2$ must be zero, and thus the noise can not be added to the last equation, being impossible from physical point of view also.

CONCLUSIONS

The objective of the work was to define the problem of of adding continuous white noise on DAE models and ply around into a case study, e.g. a d.c. motor model.

Up to now, the most difficult step is the computation of the row reduced form of the matrix fraction description, in order to compute the unimodular matrix $\mathbf{U}(s)$. References based on (Rugh, 1996) and on Polynomial Matlab Toolbox, (PolyX, 2004), are failed until now.

The principle of the method is interesting and very useful in practice mainly because it is possible to decide at the modelling stage which variables accept white noise and which not.

There are several ideas for further work, some of important directions are:

- Simulation of the noise in Modelica, as source signal components in specific libraries.
- Automatic translation of neutral models into the DAE form in Matlab. The advantage is of using the specialized functions for model conversion and transformations, ready available in Matlab.
- Debugging of simulation models, i.e. finding the cause of errors models of physical systems. There is a need to filter a broad range of errors without having to execute the simulation model. Static debugging tools should reduce the number of test cases usually needed to validate a simulation model.

Finally, it seems interesting that the same problems are studied also in other research centres and laboratories, e.g., (VisiMode, 2005).

ACKNOWLEDGEMENTS

The author would like to thank the Romanian National University Research Council for supporting part of the work, under the research grant 1350.

APPENDIX A - D.C. MOTOR PHYSICAL MODEL

The physical model is obtained by considering the aggregation of the sub-models on specific domains, i.e. mechanical, chemical, and electrical, etc. In electrical domain, the induced back emf voltage is

$$e(t) = K_b \cdot \omega(t) \tag{A.1}$$

wherein K_b is the motor back emf constant. The equations of voltages across the input circuit is

$$u(t) = i(t) \cdot R + L \cdot \frac{di(t)}{dt} + e(t) \tag{A.2}$$

In mechanical domain, the active and resistive moments are connected by equations

$$T_m(t) = J_T \cdot \alpha(t) + b \cdot \omega(t) \tag{A.3}$$

$$\alpha(t) = \frac{d\omega(t)}{dt} \tag{A.4}$$

$$\omega(t) = \frac{d\theta(t)}{dt} \tag{A.5}$$

$$J_T = J_M + J_L \tag{A.6}$$

$$J_L = \frac{1}{2} M \cdot r_d^2 \tag{A.7}$$

where J_T is the load inertia, $\omega(t)$ is the motor shaft angular velocity. The connection between the two domains is made by

$$T_m(t) = K_t \cdot i(t) \tag{A.8}$$

where T_m is the motor shaft torque and K_t is the motor torque constant.

The equations from (A.1) till (A.8) define a set of differential algebraic equations, i.e. a DAE model which was considered across the paper, as the start point in physical modelling.

It is interesting to show that in almost all applications with objective of control, detection and estimation a state representation is used as

$$L \cdot \frac{di(t)}{dt} = -R \cdot i(t) - K_b \cdot \omega(t) + u(t) \tag{A.9.a}$$

$$J_T \cdot \frac{d\omega(t)}{dt} = K_t \cdot i(t) - b \cdot \omega(t) \tag{A.9.b}$$

$$y(t) = \omega(t) \tag{A.9.c}$$

APPENDIX B – ALGEBRA KNOWLEDGE

The definitions and the theorems are adapted from (Gerdin, 2004), (Rugh, 1996) and (PolyX, 2004).

A polynomial matrix is *non-singular* if it has full normal rank. A polynomial matrix is *row reduced* if its leading row coefficient matrix has full row rank. Another equivalent definition says that a non-singular polynomial square matrix is row reduced is the degree of the matrix is the sum of the rows' degrees.

A polynomial matrix **P**(s) has *full column rank* (or full *normal column rank*) if it has full column rank everywhere in the complex plane except at a finite number of points. Similar definitions hold for *full row rank* and *full rank.*

Any polynomial matrix with full row rank may be transformed into *row reduced form* by pre-multiplying it by a suitable unimodular matrix.

Theorem 1: If a matrix $\mathbf{D}(s)$ is row reduced, then $\mathbf{D}^{-1}(s)\cdot\mathbf{N}(s)$ is proper if and only if each row of $\mathbf{N}(s)$ has degree less than or equal the degree of the corresponding row of $\mathbf{D}(s)$, i.e.,

$$r_i[\mathbf{N}] \le r_i[\mathbf{D}], i = 1,2,...,n \qquad \text{(B.1)}$$

■

Theorem 2: Let the transfer matrix function $\mathbf{G}(s) = (s\mathbf{E}-\mathbf{J})^{-1}\cdot\mathbf{K}_2$ with squares matrices **E** and **J**. Let $\mathbf{D}(s)$ the row reduced form of the matrix $(s\mathbf{E}-\mathbf{J})$, thus $\mathbf{D}(s) = \mathbf{U}(s)\cdot(s\mathbf{E}-\mathbf{J})$. Let $\mathbf{u}_{ij}$ the *j*-th row of the matrix $\mathbf{U}_i$. The matrix $\mathbf{G}(s)$ is proper if and only if

$$\mathbf{u}_{ij}\cdot\mathbf{K}_2 = 0, \quad i > r_j[\mathbf{D}],\ j = 1,2,..n \qquad \text{(B.2)}$$

■

REFERENCES

Åström, K.A., Elmqvist, H., Mattsson, S.E., (1998). Evolution Of Continuous-Time Modelling and Simulation. *The 12th European Simulation Multiconference,* ESM'98, Manchester, UK.

Braunschweig, B.L., Pantelidis, C.C., Britt, H.I., Sama S., (1999). Open Software Architecture for Process Modelling: Current Status and Future Perspectives. *Proceedings of the FOCAPD'99 Conference*, Breckenridge, Colorado, USA.

Campbell, S.L. (1990). Descriptor systems in the 90's. *Proceedings of the 29th Conference on Decision and Control*, Honolulu, Hawaii, USA, December.

Cellier, F.E., (1991). Continuous System Modeling. Springer Verlag, New York.

Elmqvist, H., Mattsson S.E. and Otter, M., (1999). Modelica-A Language for Physical System Modelling, Visualization and Interaction, *Proceedings of the 1999 IEEE Symposium on Computer-Aided Control System Design*, Hawaii, August 22-27.

Fritzson, P. (2004). Principles of Object-Oriented Modelling and Simulation with Modelica 2.1, Wiley-IEEE Press, New York.

Gerdin, M., (2004). Parameter Estimation in Linear Descriptor Systems, Licentiate Thesis 1085, Linkoping University, Sweden.

GCO, (2000). The Global CAPE-OPEN Consortium, http://www.global-cape-open.org.

Modelica, (2004). Modelica and Modelica Association, http://www.modelica.org/.

(PolyX, 2004). PolyX Ltd: http://www.polyx.cz, or http://www.polyx.com

Tiller, M. (2001). Introduction to Physical Modelling with Modelica, Kluwer Academic Publisher.

Schein O. and Denk, G, (1998). Numerical solution of stochastic differential-algebraic equations with applications to transient noise simulation of microelectronic circuits, *Journal of Computational and Applied Mathematics*, **100**, pp. 77-92.

Schon, T., Markus Gerdin, M., Glad T. and Gustafsson F., (2003). A Modelling and Filtering Framework for Linear Differential-Algebraic Equations, *In Proceedings of the 42nd Conference on Decision and Control*, Maui, Hawaii, USA.

Rugh, W.J., (1996). Linear System Theory, Information and System Sciences Series, Prentice Hall.

Varga, A., (1972). Numerical Algorithms and Software Tools for Analysis and Modelling of Descriptor Systems,. *In Proceedings of 2nd IFAC Workshop on System Structure and Control,* Prague, Czechoslovakia, pages 392-395.

VISIMOD, (2005). A research project funded by the Swedish Foundation for Strategic Research, http://www.visimod.org/.

Winkler, R., (2004). Stochastic Differential Algebraic Equations of Index 1 and Applications in Circuit Simulation, *Journal of Computational and Applied Mathematics*, **163**, pages 435-463.