International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 5, No.3 (May-2016)
http://dx.doi.org/10.12785/ijcds/050308

Modeling Single Flexible Link Using Hardware Description Language

Hessa Al-Junaid

Department of Computer Engineering, College of Information Technology, University of Bahrain, Sakheer, Kingdom of Bahrain

Received 1 Dec. 2015, Revised 12 Jan. 2016, Accepted 22 Mar. 2016, Published 1 May 2016

Abstract: This paper presents a well known model of arm with flexible robotic link modeled in Hardware Description Language (HDL). The system is modeled on behavioural level described as nonlinear DAEs in a new developed hardware description language called SystemC-A. It is an extended version of the C++ based digital simulator SystemC which provides analogue, mixed-signal and mixed-domain modelling capabilities. The paper objective is of twofold, verifying the new language capabilities, and modelling a system of high complexity. The model was compared with SIMULINK/MATLAB and proved to be highly accurate.

Keywords: HDL, SystemC-A, mixed-signal, mixed-physical domain, analogue simulator, single flexible link.

1. Introduction

Hardware Description Languages (HDLs) are classified into three categories, digital, analogue, and Analogue-Mixed Signal (AMS) HDLs. Examples of digital HDLs are VHDL and Verilog, they are based on event-driven techniques and a discrete mode of time. Analogue HDLs support the description of systems of Differential Algebraic Equations (DAEs) and support network semantics and behavioural descriptions. AMS HDLs support both event-driven techniques and DAEs. The most popular AMS HDLs are VHDL-AMS [1] and Verilog-AMS [2].

HDL-AMS are used to model systems from a variety of physical domains [3], such as mechanical, automotive [4] [5], and mechatronic systems. HDL-AMS has also been used to model Micro-Electro-Mechanical Systems MEMS [6], electrical power generation [7], Aircraft Electrical System [8] and in the simulation of electrothermal systems [9]. HDL-AMS allows different parts of the system to be described at behavioral level as well as component level of abstraction. In HDLs, ports and signals are used to connect hierarchical modules of different parts of the system. HDLs provide mixed domain libraries which contain different types of ports, signals, cross and through quantities to be used in specific domains such as electrical, magnetic, mechanical. Further, currently HDL capabilities are extended to a new formalism for the modeling of complex biological networks [10].

HDL-AMS is the state of the art powerful technique in computer-based modeling and simulation which demonstrates great benefits to designers and shorten time to market. HDLs provides a convenient way to create subsystems, prototyping and reusability. New libraries of components and systems can be defined and reused. There are still many models which require integrating different simulators to different parts in the simulated system. This is due to some limitations in the current HDLs. For example, in [11] VHDL-AMS is used for fault modeling of analog circuits, where part of the system required modeling in MATLAB. Also in [12], VHDL-AMS is used to design energy efficient sensor nodes optimum performance and then MATLAB optimization toolbox is used. The current HDLs are not flexible in the sense that designers cannot reuse libraries written in C++. This limitation can be solved using proposed SystemC-A since designers can easily reuse libraries such as optimization algorithms and verifications within their designs.

SystemC-A [13] is an extension to SystemC [14] digital simulator intended to extend the digital modeling capabilities of SystemC to the analogue domain. SystemC-A provides the same powerful features of VHDL-AMS and Verilog-AMS in addition to a number of extra advantages such as high simulation speed, support for hardware-software co-design and for high levels of modeling. SystemC-A was validated in previous publications in modeling mixed-signal systems such as switched-mode power supply and phased locked loop [15] [16]. In this work, SystemC-A is validated by modeling single flexible link connected to a servo-motor

E-mail address: haljunaid@uob.edu.bh

and its controller. The system is analogue mixed-physical domain described by a set of DAEs.

For modeling and simulation of controlled systems, the tool used depends on the research discipline (mechanical, electronics, etc), however, Matlab/simulink is popular. The flexible link with controller system is modeled in the literature in Matlab/simulink [17] and Labview [18] at different abstraction levels. While achieving a certain control objectives, highly accurate models are needed as well as an integrated simulation tool for all parts of the design to be simulated concurrently [19]. When more variables from attached systems or from the environment are needed in the simulation, the simulator should have the flexibility of adding more models and libraries.

Researchers focused on different aspects of modeling flexible link [20]. The nonlinear model is complicated and the controller design is challenging [21]. For instance, finite element is employed in [22] and lumped mass model is used in [23], models employed depend on requirements in complexity, accuracy, or simulation time. Verity of controllers types are designed such as optimal [24], robust [25], and classical [26].

The focus of this work is to present a behavioral model of the flexible link in HDL for the first time with accurate simulations. Developing an accurate mathematical model or maximizing the accuracy and the efficiency of the controller compared to other designs are not from the objectives of this work. In section II, the flexible arm system and its attached motor equations are derived as well as the control design. In section III, SystemC-A language is briefly explained, while in section IV proposed SystemC-A model of the flexible link system is detailed.

2. FLEXIBLE LINK ARM SYSTEM

The flexible link arm is used in the industry in numerous applications such as robot arm, space shuttle arm, and nuclear maintenance because of its low power consumption and speed compared to rigid link manipulator. Also it is a benchmark system used in control systems to verify modeling methods and controller designs. The flexible arm is a lightweight and deflecting during rotation. At the base, the arm is connected to a dc motor for actuation. When the motor actuate the arm, it will start rotating causing vibrations because the arm is bendable. Then the control objective for this system is to design a controller to suppress the vibrations while the arm is positioned at a certain angle. Fig. 1 illustrates the flexible link attached to a DC motor at one end, with some required parameters used to derive the mathematical model.

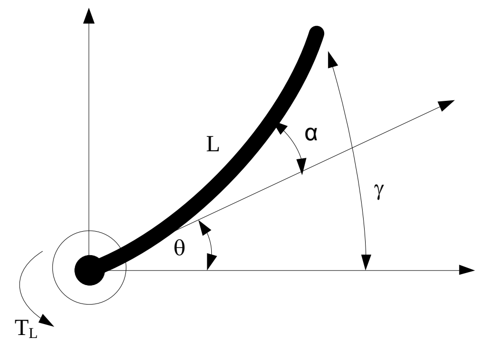

Figure 1. Schematic flexible arm attached to a DC motor.

A. Motor Mathematical Model

There are a number of models used in the literature [20]. The mathematical model adopted in this paper is behavioral model described by a set of DAEs. The system parameters are described in Table I and II and are taken from the literature. The closed loop of motor with gears is modeled as Single Input Single Output (SISO) system by the transfer function described in Eq. (1).

$$\frac{\theta_{out}(s)}{V_{in}(s)} = \frac{\frac{\eta K_m K_g}{RaJeq}}{s\left(s+\frac{Beq}{Jeq}+\frac{\eta K_m^2 K_g^2}{RaJeq}\right)} \quad (1)$$

The output torque $T_L(s)$ is,

$$T_L(s) = \frac{\eta K_m K_g}{R_a}\left(V_i(s) - K_m K_g \Omega(s)\right) \quad (2)$$

In time domain the output torque is,

$$T_L(t) = \frac{\eta K_m K_g}{R_a}\left(v_i(t) - K_m K_g \dot{\theta}(t)\right) \quad (3)$$

The system parameters are listed in Table I.

TABLE I
MOTOR PARAMETERS.

Symbol	Definition	Value
R_a	armature resistance	2.6Ω
K_m	motor voltage constant	0.00767*V.s/rad*
K_τ	motor torque constant	0.00767*N.m/A*
K_g	high gear ratio	(14)(5)
J_m	armature inertia	$3.87x10^{-7} kg.m^2$
J_{tach}	tachometer inertia	$0.7x10^{-7} kg.m^2$
η	motor efficiency	0.7395
J_{eq}	equivalence inertia	$0.0023 kg.m^2$

B. Flexible Link Mathematical Model

The flexible link parameters are defined in Table II.

$$\dot{\theta} = \omega \quad (4)$$

$$\dot{\alpha} = \nu \quad (5)$$

The torque of the flexible link acceleration is defined as,

TABLE II FLEXIBLE LINK PARAMETERS

Sympol	Definition
θ	servo gear angular displacement
ω	servo gear angular velocity
α	flexi link angular deflection
ν	flexi link angular velocity
L	flexible link length L = 0.381m
γ	total deflection $\gamma = \theta + \alpha$
m	mass of flexible link m = 65gm
D	end point arc length deflection $D = \alpha L$
ω_{FL}	link's damped natural frequency 6π
J_{arm}	link's moment of inertia J_{arm} = 0.0031452
K_{gage}	strain gage calibration factor 1*volt=inch*
K_{stiff}	link's stiffness K_{stiff} = 1.1175
B_{eq}	Equivalent viscous friction=0.0004*Nm/(rad/sec)*

$$T_{Jarm} = J_{arm}\frac{d^2\gamma}{dt^2} = J_{arm}(\ddot{\theta} + \ddot{\alpha}) = J_{arm}(\dot{\omega} + \dot{\nu}) \quad (6)$$

The torque due to torsional spring K_{stiff} is assumed to be proportional to the link's deflection α, i.e.,

$$T_{Kstiff} = K_{stiff}\alpha \quad (7)$$

$$T_{Jarm} + T_{Kstiff} = 0 \quad (8)$$

Or,

$$J_{arm}(\dot{\omega} + \dot{\nu}) + K_{stiff}\alpha = 0 \quad (9)$$

The output torque of the servomotor is defined in Eq.(10), it overcomes the inertia torque due to *Jeq* and the frictional torque, also, it is assumed to overcome the torque due to link's acceleration, i.e.,

$$T_L = J_{eq}\dot{\omega} + B_{eq}\omega + J_{arm}(\dot{\omega} + \dot{\nu}) \quad (10)$$

substituting for $J_{arm}(\dot{\omega} + \dot{\nu})$ from Eq.(9) into Eq.(10) resulted in,

$$\dot{\omega} = \frac{K_{stiff}}{J_{eq}}\alpha - \frac{1}{J_{eq}}T_L + \frac{B_{eq}}{J_{eq}}\omega \quad (11)$$

substituting for T_L from Eq.(3) yields,

$$\dot{\omega} = \frac{K_{stiff}}{J_{eq}}\alpha - \frac{\eta K_m^2 K_g^2 + B_{eq}R_a}{J_{eq}R_a}\omega + \frac{\eta K_m K_g}{J_{eq}R_a}\nu_i \quad (12)$$

substituting for $\dot{\omega}$ from Eq.(12) into Eq.(9) yields,

$$\dot{\nu} = -\frac{(J_{arm}+J_{eq})K_{stiff}}{J_{eq}J_{arm}}\alpha + \frac{\eta K_m^2 K_g^2 + B_{eq}R_a}{J_{eq}R_a}\omega + \frac{\eta K_m K_g}{J_{eq}R_a}\nu_i \quad (13)$$

To obtain a state-space model of the system comprising the servomotor and the flexible link, the following state variables are chosen,

$$x(t) = [\theta \quad \alpha \quad \omega \quad \nu]^T \quad (14)$$

Then using Eq. (4),(5), (12) and (13), the state-space model is written as,

$$\begin{bmatrix}\dot{\theta}\\ \dot{\alpha}\\ \dot{\omega}\\ \dot{\nu}\end{bmatrix} = \begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 0 & \frac{K_{stiff}}{J_{eq}} & -\frac{\eta K_m^2 K_g^2 + B_{eq}R_a}{J_{eq}R_a} & 0\\ 0 & -\frac{K_{stiff}}{J_{eq}}J_{arm} & \frac{\eta K_m^2 K_g^2 + B_{eq}R_a}{J_{eq}R_a} & 0\end{bmatrix} . \begin{bmatrix}\theta\\ \alpha\\ \omega\\ \nu\end{bmatrix} + \begin{bmatrix}0\\ 0\\ \frac{\eta K_m K_g}{J_{eq}R_a}\\ -\frac{\eta K_m K_g}{J_{eq}R_a}\end{bmatrix} . \dot{\nu}_i \quad (15)$$

C. Controller Design

The flexible link system controlled in the literature using linear and nonlinear methods for different accuracies targets. In this work, a linear method of state feedback is used. It can be used when the system is controllable, and full state feedback is used when all states are available and measurable. Using Matlab commands for pole placement, the gain K is calculated and then the control Law is,

$$u = -K * x(t) = -K * [\theta \quad \alpha \quad \omega \quad \nu]^T \quad (16)$$

where, $K = [6.5 \quad 0.1 \quad 0.01 \quad 0.0003]$

3. SYSTEMC-A LANGUAGE

SystemC-A is a language based on C++ and it consists of language constructs to provide modeling capabilities to model analogue systems at different abstraction levels from circuit level to system level. It has a library of primitive circuit components such as resistor, capacitor, and voltage sources. Also SystemC-A provides flexibility to support user-defined components. It has a synchronization mechanism to link the analogue engine to SystemC event driven engine. Also, it support the smooth integration of digital and analogue parts using special modules for interfaces. A modular analogue simulator has been built using efficient linear and nonlinear numerical methods and it proved to be accurate and perform fast simulations. Fig. 2 illustrates the architecture of SystemC-A and its synchronization to SystemC. Some details of the language is explained in section IV and for more details please refer to previous publications [13] [27].

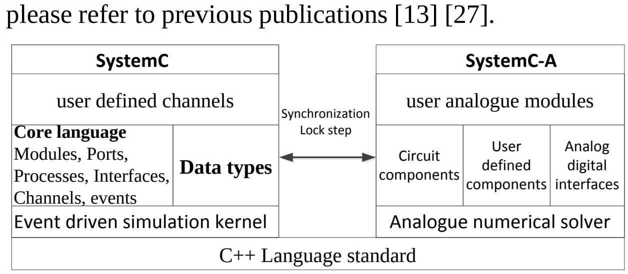

Figure 2. SystemC-A language architecture.

4. SYSTEMC-A MODEL OF THE CONTROLLED FLEXIBLE ARM

The model consists of three C++ modules linked to SystemC-A library. A module to model the arm and motor, and another for controller, and the last is a testbench module. The system is modeled using mixed-level models where some parts are modeled in behavioral level and others at circuit level. The testbench illustrated in Fig. 3 includes instances of all parts connected together by signals or nodes.

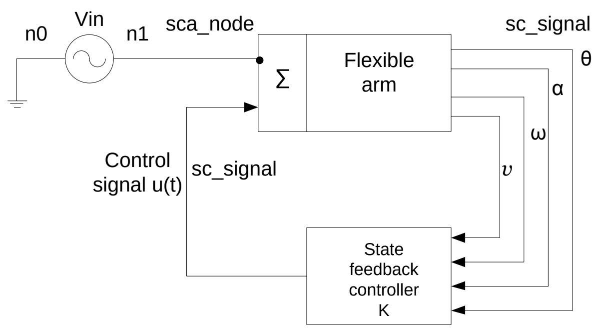

Figure 3. Flexible arm with state feedback modules in SystemC-A.

The system could be tested with different types of inputs: step, sinusoidal, and random noise. The tested signals are modeled as component module as described before. For example, the sinusoidal input, is a voltage source from SystemC-A library of circuit components. The sinusoidal stimulus is then connected to the arm module via nodes in the testbench module as shown in Listing 1.

Listing 1. SystemC-A testbench of the flexible arm system.

```
void testbench::system(){
 // global connecting signals
 sc_signal<double> x1,x2,x3,x4,feedback;
 // instantiating nodes and components
 n2 = new Node("0");
 n1 = new Node("n1");
 voltageS_sin *Vin1  =new voltageS_sin("Vsin", n1,n2,
    0.2,0,6.5,0,0);//freq,offset,amplitude,delay,damping
  Arm *arm1  =new Arm("arm1",n1,&feedback,&x1,
                                        &x2,&x3,&x4);
  controller *cont1 =new controller("control1",&x1,&x2,&x3,
                                     &x4,&feedback);
  sc_start(5,SC_SEC); \\start simulation for 5 Seconds}
```

The arm and controller modules are C++ modules defined as SystemC-A component, where the components constructor defines the number and type of the components inputs and outputs. System variables are defined in the constructor, as shown in Listing 2 which is SystemC-A model of controlled flexible arm. The arm constructor instantiate system variables $\theta, \alpha, \omega, \upsilon$ in Line 8-11. In Listing 2 BuildB() function include DAEs of the flexible arm system. Eq.(15) defined earlier in section II is modeled in Line 30-33. The arm module contains four system variables which need four Equation() functions to be defined in BuildB(). The arm module is connected to the controller via SystemC signals as shown in line 13-18. This syntax is analogy to other HDL such as Verilog-AMS and VHDL-AMS.

Listing 2. SystemC-A model of controlled flexible arm.

```
1  ...
2  Arm::Arm(char nameC[5],SystemVariable*node_a1,
3   sc_signal<double>*error_sig1,sc_signal<double>*x1_sig1,
4   sc_signal<double>*x2_sig1,sc_signal<double>*x3_sig1,
5   sc_signal<double>*x4_sig1):component(nameC,0, 0, 0){
6
7  //system variables
8   y1 = new sc_free_variable("y1");//theta
9   y2 = new sc_free_variable("y2");//alpha
10  y3 = new sc_free_variable("y3");//omega
11  y4 = new sc_free_variable("y4");//v
12
13  error_sig=error_sig1;
14  x1_sig=x1_sig1;
15  x2_sig=x2_sig1;
16  x3_sig=x3_sig1;
17  x4_sig=x4_sig1;
18  node_a=node_a1;
19 }
20  void Arm::BuildB(){
21  ...
22  Vin=X(node_a);// sinusoidal system input
23  u1=error_sig->read()+Vin;// control input
24
25  x1_sig->write(Y1n);// writing to output signals
26  x2_sig->write(Y2n);
27  x3_sig->write(Y3n);
28  x4_sig->write(Y4n);
29  // four arm equations
30  Equation(y1,-Y1dotn + Y3n);
31  Equation(y2,-Y2dotn + Y4n);
32  Equation(y3,-Y3dotn + 480*Y2n-17*Y3n+13.2*u1);
33  Equation(y4,-Y4dotn - 842*Y2n+17*Y3n-13.2*u1);   }
```

Listing 3 shows SystemC-A model of the state feedback controller. The first four signals defined in the constructor (line 2-4) are the state inputs, while the fifth signal is the control output signal. Line 15-16 is reading input signal values then multiplied them by K, the state feedback gain *feedback* = *Kx*. Line 17 is writing the feedback value to the output signal.

Listing 3. SystemC-A model of state feedback controller.

```
1   ...
2   controller::controller(char nameC[5],sc_signal<double>
3   *x1_sig1,sc_signal<double>*x2_sig1,sc_signal<double>*x3_sig1,
4   sc_signal<double>*x4_sig1,sc_signal<double>*feedback_sig1):
5       component(nameC,0, 0,0){
6         x1_sig=x1_sig1;
7         x2_sig=x2_sig1;
8        x3_sig=x3_sig1;
9        x4_sig=x4_sig1;
10       feedback_sig=feedback_sig1;
11  }
12  void controller::buildB(){
13  // controller parameters:
14  //calculating control feedback signal
15  feedback= -6.5*x1_sig->read() - 0.1*x2_sig->read()
16      -0.01*x3_sig->read() - 0.0003*x4_sig->read();
17  feedback_sig->write(feedback);//writing to output signal
18 }
```

The system was simulated using two types of input signals, a step and a sinusoid. The sinusoid has an amplitude of 6.5v and a frequency of 0.2Hz. Fig. 4 and 5 shows a 5 seconds simulation results of the system states, $\theta, \alpha, \omega, \upsilon$. The input represents the voltage applied to drive the motor to move the arm in a certain angle with a certain speed.

A. Comparison with Simulink

The system is also built in Simulink and Fig. 4 and 5 shows the system states simulated in Simulink as well. The results matched SystemC-A simulations to a very high accuracy. The Root Mean Square Error (RMSE) is shown in Table III, for $\theta, \alpha, \omega, \upsilon$. The error is more in the first second of the simulation and it is nearly zero afterwards. It is worth to mention that the solver used in SystemC-A is based on Trapezoidal numerical method while in Simulink fourth order Runga-Kutta is chosen because Trapezoidal is not within the options in Simulink. The small difference in simulations is therefore due to different solvers. For fair comparison the absolute tolerance and relative tolerance should be the same in both simulators.

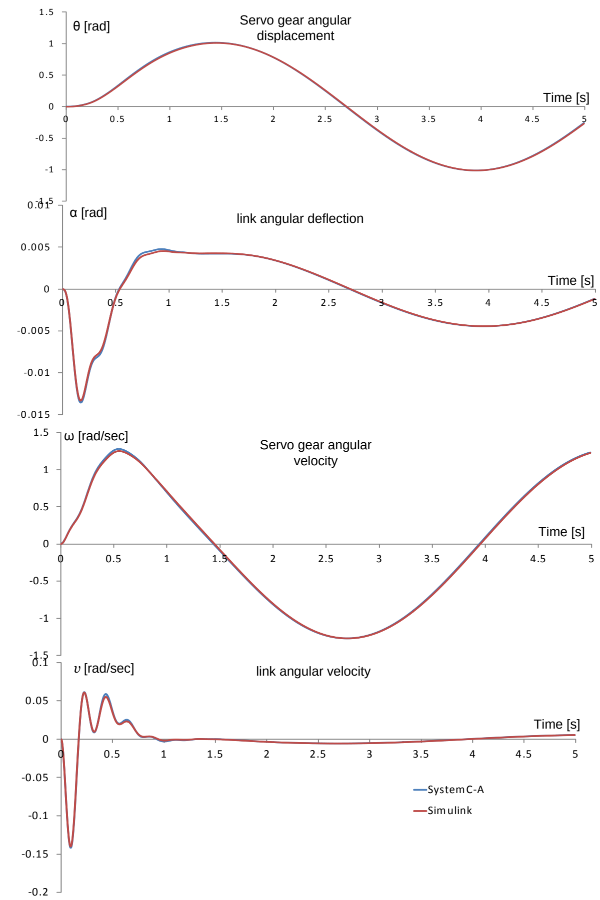

Figure 4. Flexible arm with state feedback controller simulation results of SystemC-A and Simulink for a sinusoidal input.

TABLE III
RMSE OF THE SYSTEM STATES BETWEEN SYSTEMC-A AND SIMULINK

	Step	Sinwave
θ	0.0056	0.0074
α	0.0017	1.0949e-004
ω	0.0438	0.0127
ν	0.0479	6.9756e-004

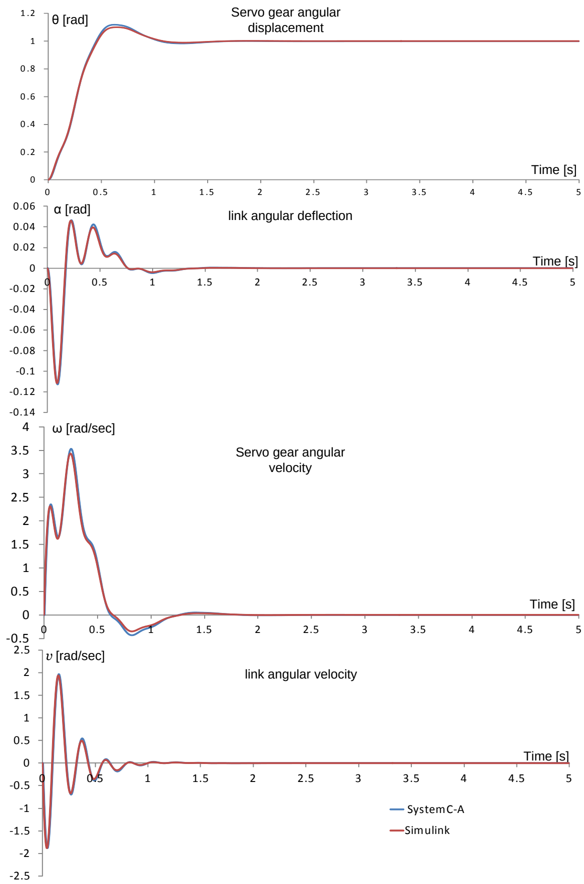

Figure 5. Flexible arm with state feedback controller simulation results of System C-A and Simulink for a step input.

5. CONCLUSION

A hardware description model of a flexible link system with state feedback controller has been developed in SystemC-A language. This has demonstrated the ability of SystemC-A to model complex systems in HDL context. Also, SystemC-A simulations were compared to Simulink simulations, showing highly comparable numerical figures, which proves that SystemC-A can be compared to well established HDLs. The work also present one integrated simulation tool for modeling and simulating complete design of mixed-physical domain systems at different abstraction level. The advantage of that SystemC-A is built on C++, more analysis could be done by re-use C++ libraries already designed by others.

REFERENCES

[1] P. J. Ashenden, G. D. Peterson, D. A. Teegarden, The system designer's guide to VHDL-AMS: analog, mixed-signal, and mixed-technology modeling, Morgan Kaufmann, 2002.

[2] Accellera Systems Initiative, www.accellera.org, Verilog-AMS Language Reference Manual 2.3 (August 2008).

[3] P. Nikitin, R. Shi, "Vhdl-ams based modeling and simulation of mixed-technology microsystems: A tutorial", Integration the VLSI Journal 40 (3) (2007) 261–273. doi:10.1016/j.vlsi.2005.12.002.

[4] K. Tsuji, K. Yamamoto, "Power plant model of fuel consumption and vibration for vehicle concept planning", SAE Technical Paper 2015-01-2253.

[5] H. Aljunaid, T. Kazmierski, L. Wang, "Systemc-a modelling of an automotive seating vibration isolation system", in: Proceedings of Forum on Specification and Design Languages, Darmstadt Germany, 2006.

[6] C. Zhao, L. Wang, T. J. Kazmierski, "An efficient and accurate mems accelerometer model with sense finger dynamics for applications in mixed-technology control loops", in: Behavioral Modeling and Simulation Workshop, 2007. BMAS 2007. IEEE International, IEEE, 2007, pp. 143–147.

[7] Q. Ha, Y. Maret, J. Rodriguez Estupinan, A. Vachoux, "Vhdl-ams virtual prototyping of a generator circuit breaker ablation monitoring system", in: Circuits and Systems (ISCAS), 2015 IEEE International Symposium on, 2015, pp. 1854–1857. doi:10.1109/ISCAS.2015.7169018.

[8] X. Li, S. Kher, S. Huang, V. Ambalavanar, Y. Hu, "System level component modeling of aircraft electrical system using vhdl-ams", in: Proceedings of the World Congress on Engineering and Computer Science, Vol. 2, 2015.

[9] P. Nikitin, E. Normark, C.-J. Shi, "Distributed electrothermal modeling in vhdl-ams", in: Behavioral Modeling and Simulation, 2003. BMAS 2003. Proceedings of the 2003 International Workshop on, 2003, pp. 128–133. doi:10.1109/BMAS.2003.1249871.

[10] C. Lallement, M. Madec, "Challenges in design-oriented modeling in biology", in: Mixed Design of Integrated Circuits Systems (MIXDES), 2015 22nd International Conference, 2015, pp. 17–22. doi:10.1109/MIXDES.2015.7208471.

[11] L. Xia, M. U. Farooq, I. M. Bell, "High level fault modeling of analog circuits through automated model generation using chebyshev and newton interpolating polynomials", Analog Integrated Circuits and Signal Processing 82 (1) (2015) 265–283.

[12] T. J. Kazmierski, L. Wang, M. Aloufi, "Energy efficient sensor nodes powered by kinetic energy harvesters–design for optimum performance", Electronics 16 (1) (2012) 65–76.

[13] H. Al-Junaid, T. Kazmierski, "An Analogue and Mixed-Signal Extension to SystemC", IEE Proc. Circuits, Devices & Systems 152 (6) (2005) 682–690.

[14] Accellera Systems Initiative, www.accellera.org, IEEE 1666-2011 Standard SystemC Language Reference Manual (2011).

[15] H. Aljunaid, T. Kazmierski, "SEAMS - A SystemC environment with analog and mixed-signal extensions", in: Proceedings of the International Symposium on Circuits and Systems ISCAS04, Vol. 5, Vancouvor Canada, 2004.

[16] T. Kazmierski, H. Aljunaid, "Synchronisation of Analogue and Digital Solvers in Mixed-Signal Simulation on a SystemC Platform", in: Proceedings of of Forum on Specification and Design Languages, Frankfurt Germany, 2003.

[17] I. Akyuz, E. Yolacan, H. Ertunc, Z. Bingul, "Pid and state feedback control of a single-link flexible joint robot manipulator", in: Mechatronics (ICM), 2011 IEEE International Conference on, 2011, pp. 409–414. doi:10.1109/ICMECH.2011.5971320.

[18] M. Doosthoseini, M. Korayem, A. Shafei, B. Kadkhodaei, "An experimental interface design for a single-link elastic manipulator system", International Research Journal of Applied and Basic Sciences 3 (8) (2012) 1726–173.

[19] P. Aarnio, "Simulation of a hybrid locomotion robot vehicle", Tech. rep., Helsinki University of Technology (2002).

[20] S. Dwivedy, P. Eberhard, "Dynamic analysis of flexible manipulators, a literature review", Mechanism and Machine Theory 41 (2006) 749777.

[21] R. Cannon, J. Schmitz, "Initial experiments on the end-point control of a flexible one-link robot", International Journal of Robotics Research 3 (3) (1984) 6275.

[22] J. Martins, Z. Mohamed, M. Tokhi, M. B. J. da Costa, “Approaches for dynamic modelling of flexible manipulator systems”, IEE Proceedings- Control Theory and Applications 150 (4) (2003) 401–411.

[23] G. Zhu, S. Ge, T. Lee, “Simulation studies of tip tracking control of a single-link flexible robot based on a lumped model”, Robotica (17) (1999) 7178.

[24] H. Wai, M. Lee, “Intelligent optimal control of single-link flexible robot arm”, IEEE Transactions on Industrial Electronics 51 (1) (2004) 201–220.

[25] J. Yeon, J. Yim, J. Park, “Robust control using recursive design method for flexible joint robot manipulators”, Journal of Mechanical Science and Technology 25 (11) (2011) 3205–3213.

[26] T. Mansour, A. Konno, “Robot Manipulators New Achievements”, InTech, 2010, Ch. Vibration Based Control for Flexible Link Manipulator, pp.435–457.

[27] H. Al-Junaid, “SystemC-A An Analogue and Mixed-Signal Language”, Ph.D. thesis, University of Southampton (2006).

Dr. Hessa Al-Junaid is an Assistant Professor of Computer Engineering at the University of Bahrain, College of Information Technology. She received the B.Sc. and M.Sc. degrees in electronics engineering from the University of Bahrain, Bahrain, in 1996 and 2001, respectively. In 2006 Dr. Al-Junaid received her Ph.D. degree from the University of Southampton, Southampton United Kingdom. Dr. Al-Junaid was a chairperson for Computer Engineering departments between 2010-2013. Her research interests include modeling and simulation of mixed-signal circuits and systems and mixed-disciplinary systems, brainwaves modeling and analysis for Brain Machine Interface and robotics applications. Dr. Al-Junaid is an active member in the Institution of Engineering and Technology IET Bahrain chapter.