A Comparative Study of Computer-Aided Engineering Techniques for Robot Arm Applications

Lilia Zouari
CES Laboratory
Sfax University, Tunisia
lilia.zouari@gmail.com

Slim Chtourou
College of Engineering, Electrical Department
King Faisal University, Saudi Arabia
slim.chtourou@ieee.org

Mossaad Ben Ayed
AlGhat College of Science and Humanities
Al Majmaah University, Saudi Arabia and
University of Sfax, Sfax Tunisia
Tunisiamm.ayed@mu.edu.sa

Shaya A. Alshaya
Department of Computer Science
Majmaah University
Majmaah, Saudi Arabia
shaya@mu.edu.sa

***Abstract*-As systems become increasingly complex, their simulation techniques have to be more accurate and enhanced. Despite the wide use of robotic arms in industries, they still encompass a wide number of complexities. The control of a flexible robot arm driven by a Brushless DC Motor (BDCM) for tracking problems is a great challenge, not only for its complex algorithms but also for its verification process. Robotic systems are designed heterogeneously by combining continuous and event discrete models. Therefore, computer-aided engineering tools have to be enhanced in order to support the verification of the control algorithms. Ensuring definite and rapid simulations is a challenging task for robotic application systems. Although control strategies can be tested using the Matlab/Simulink environment to assess their performance, their verification at a low-level still remains a very difficult task. This paper studies different simulation techniques based on Model In the Loop (MIL), Hardware In the Loop (HIL), and Hardware Software In the Loop (HSIL) on a flexible robot arm driven by BDCM in order to overcome each method's limits, focusing on performance and cost and simulation time reduction. The HSIL achieves the highest accuracy and speed than MIL and HIL and provides reusability and portability of the control unit compared to the other techniques.**

Keywords- control; robot arm; brushless DC motor; model in the toop; hardware in the loop; hardware software in the loop

I. INTRODUCTION

Robots' control driven by BDCM has achieved an interesting breakthrough recently [1-2]. The utilization of BDCMs is widely spread due to their independence of bearings and brushes, providing them a long survival time, robustness, energy efficiency, and implementation simplicity [3-7]. Authors in [8] utilized BDCM in robotics to benefit from its advantages, such as the execution time decrease of control algorithms. They showed that this was a great advantage, especially when looking for robot manipulators either in the industry performing repetitive tasks, or in service robots performing manipulation tasks. Nevertheless, despite the progress in this field, there is still an urgent need to minimize cost and time for certain robotic applications [9-13]. Robotic systems still require many improvements in design, accuracy, and real-time processing. On the other hand, computer-aided design methods have to be up to date in order to support the complexity and the needs of these systems. This challenge is the main subject of this paper, as it attempts to evaluate the control algorithm performance for a flexible robot arm with one freedom degree using Model In the Loop (MIL), Hardware In the Loop (HIL), and Hardware Software In the Loop (HSIL) [14] simulation techniques.

The more complex is a control algorithm, the longer simulation time it requires. Hence, the design of accurate and rapid simulations is a great challenge for robotic systems, as the need for high performance and available real-time systems has increased their complexity. The design of these complex systems requires an efficient simulation tool for verification, ensuring high performance in less time. HIL appends the sincere device in the verification process. This may enhance the test capacity, diminish conceit time and predict the environment interaction. Despite its pertinence, compounding variant architecture conceits and protocols at the same time is a complex mission [15]. In [16], an environment to verify and test the hardware via HIL was presented and integrated in MATLAB/Simulink. The simulation model was synchronized via real-time communication, and logic control was performed by Simulink. Authors in [17] presented a technique to test real time embedded systems via HIL, benefiting from High-Level Architecture (HLA) for interoperability and synchronization of heterogeneous architectures. Their proposed test used Ptolemy to verify the patterns operating in hardware and their correspondent Ptolemy developed reference patterns in real-time. This approach developed further performers in Ptolemy to integrate it in an HLA and to interface the software in the hardware under verification. As substantiation of concept, it was applied on a navigation algorithm for a mobile robot.

Corresponding author: Mossaad Ben Ayed

Every data collected by the sensors and the reactions were transmitted in real-time to Ptolemy performing the check in the face of a reference model. This allowed multiple computation models to be exploited as reference models, in order to checkout multiple hardware architectures synchronously based on HLA [17]. In [18], a distributed co-simulation environment to integrate virtual components was introduced. The co-simulation used a modified HLA nominated as Distributed Cosimulation Backbone, which was the standard to exchange data, claiming specific appeals to run-time infrastructure functions. In general, the operation was similar to high-level architecture. Similarly, other works used HIL to test embedded systems, whereas they were restrained to a restricted set of models and devices, with specific architectures and computation models making the integration harder. In contrast, in [17] the co-simulation was performed for homogeneous models only. Unfortunately, these models lack any easy extensions for integration with heterogeneous systems, and their adaptation to diverse hardware platforms, synchronization and communication protocols require much effort. Authors in [19] attempted to support synchronization and heterogeneity by involving different computation patterns and integrating real-time hardware devices in Ptolemy using high-level architecture. This was an improvement to the integration of several instances of Ptolemy to improve performance without HIL [20]. In [21], Ptolemy was integrated with hardware devices for testing excluding real-time constraints. However, test integration of embedded systems with HIL via Ptolemy and high-level architecture has not been found.

Nowadays, designing complex systems requires tools and methods that allow verification and simulation in a practical and efficient way [22]. The use of different modeling and simulation tools during a unique design flow is a very common procedure. However, the integration of hybrid systems is not easy, as there is no guarantee that continuous and discrete systems that work smoothly when separated, will work well when combined [23]. This paper studies the development and evaluation of different verification techniques, based on high-level architecture, for a flexible robot arm driven by BDCM. To reach this goal, a verification environment for the control of the robot system is necessary. Therefore, at first, the MIL simulation based on Matlab/Simulink was used. Then, its results were compared with the results of HIL and HSIL, in order to feature a high performance and short-time method for heterogeneous and complex embedded robot systems. This approach of co-simulation was expected to allow the synchronization between heterogeneous models, such as continuous and discrete models, integrating different simulators.

II. PROBLEM FORMULATION AND CONTROL STRATEGY

This section describes the control strategy associated to the BDCM and the flexible robot arm.

A. BDCM Control

The DC voltage ensuring power over the three arms in high- and low-level was conventionally provided by an AC-DC converter which was easily achievable through batteries. In line with [24], it was assumed that the Insulated-Gate Bipolar Transistors (IGBTs) and the diodes constituting the inverter were ideal switches, in order to simplify the brushless motor-conventional inverter association formulation. The voltages of the BDCM's three phases, V_{an}, V_{bn}, and V_{cn} were expressed as functions of *S1-6* signals, characterizing the IGBTs of the three arms state. The BDCM phase voltages *a*, *b*, and *c* were determined considering the signals combinations *S1-6*. Each functioning sequence was divided into motor sub-sequence during which the power was transmitted from the battery to the motor stator phases through two IGBTs, and a generator sub-sequence resulting in a power recovery that was returned to the battery through two diodes. The IGBTs conduction outbreak was ensured by the hysteresis current controllers. The principle of the hysteresis controller was to maintain the real current within a centered given width band around the reference current I_{ref} . Figure 1 shows the flowchart of the BDCM control.

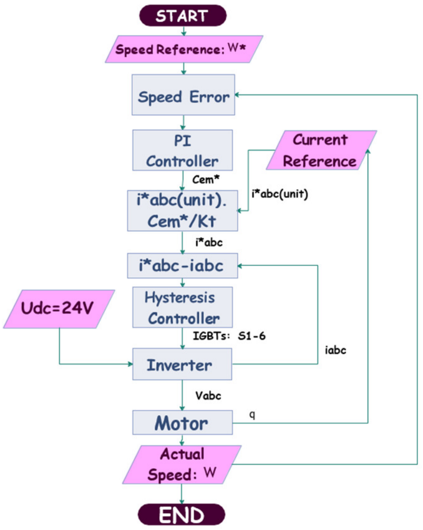


Fig. 1. Flowchart of BDCM control.

B. Flexible Robot Arm Control

The proposed design focused on a flexible robot arm with one degree of freedom. Most similar robots use an actuator brushless motor. The flexible robot arm based on BDCM provides fast, precise movements, and excellent dynamic response with a high dynamic and torque performance. The use of a transmission system is essential for the flexible robot arm. Figure 2 describes the flowchart of the flexible robot arm control.

III. VERIFICATION TECHNIQUES

The choice of the verification technique of the developed digital systems is a very important task. Any verification technique should be based on two essential requirements. At first, the existence of a software simulation tool is essential to monitor the performance and avoid potential risk scenarios in real tests. Additionally, the provision of a hardware simulation tool is imperative to verify its functionality and measure its performance under realistic conditions. The following section presents the three verification techniques used for the control of the flexible robot arm driven by BDCM.

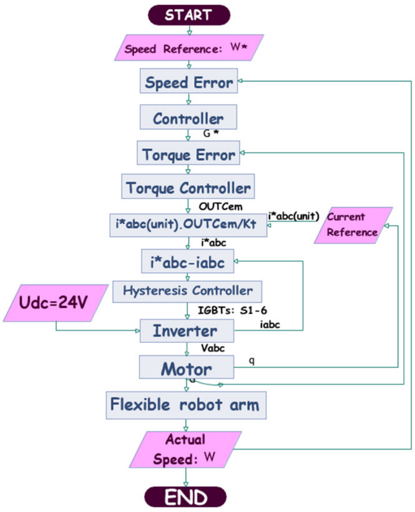


Fig. 2. Flowchart of flexible robot arm control.

A. MIL Simulation Technique

The simulation approach for dynamic systems analysis is a vital step in all robotics studies. For this reason, the scheme of the closed-loop robot manipulator control was modeled using Simulink as shown in Figure 3.

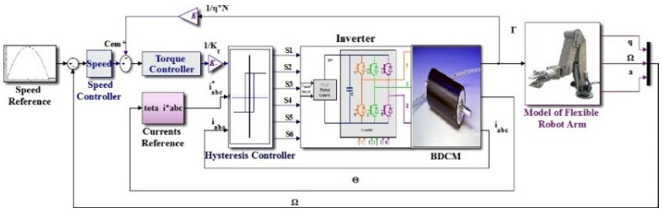


Fig. 3. Flexible robot arm control using MIL.

Machine-converter associations were particularly involved in this, study since these systems are quite complex, nonlinear, presenting multiple time scales with very small time constants, and variables depending on the operating point. Within a perspective to implement the control algorithms for brushless motors driving the robot arm, special focus was given on studying and checking the dynamic performance of this robot in Matlab/Simulink environment. This model uses the electrical and mechanical equations and quantities already developed from the considered BDCM, and this section focuses on the implementation of the control strategy, adopting the dynamic model of the robot with the determined controller parameters.

B. HIL Simulation Technique

There is an increased interest in using electrical drives and power electronics in the control of a large variety of different applications, particularly in the robotic field. However, authors in [25, 26] recommended a serious consideration of the design and strict control of electrical drives and power electronics to validate their control strategies. Therefore, HIL and rapid prototyping methods are described as a deep layer of verification that is required after the software simulation. HIL simulators were essentially used for the verification of complex electromechanical systems such as industrial robots [27]. HIL simulations always performed the real-time simulation of the system including actuator and sensor models. Furthermore, theoretical considerations have been reported along with simulation results related to differences and the evaluation of approaches [28]. This technique has multiple advantages [27]:

- During the initial design phases, the platform is not always available to perform real tests.
- Cost and simulation time will be reduced.
- In the presence of system uncertainties during the initial phases of the design, HIL is the safest simulation technique to avoid all scenarios risks and dangerous situations.
- The verification flexibility can be automated and reproducible.

Simulink/Matlab provides HIL by performing co-simulation between the FPGA board and the Quartus II software design [27, 29-33]. HIL simulation is used to achieve the control strategy of the flexible robot arm, and ensures the verification of the effectiveness of controllers in a repeatable and cost-effective way [29, 31]. The steps required to perform the verification of the controller's accuracy according to the HIL [30-33], are:

- Step 1: Provide the model of the controllers based on the DSP builder blocks in the Simulink/Matlab environment.
- Step 2: Compile the design based on the Signal Compiler. Compilation creates a compiled Quartus II project presented in Figure 4. The compilation procedure (Figure 5) comprises of four steps. The closed-loop flexible robot arm control driven by BDCM is described using the DSP builder tool to perform the HIL simulation based on the Altera DE2 board, as shown in Figure 6.
- Step 3: Select the adequate configuration of the HIL interface and program the FPGA board, as shown in Figures 7 and 8.

After finishing all these steps, the verification of the controllers' efficiency is performed through simulation with MATLAB/Simulink and an Altera FPGA board. However, despite the provided modeling language and the HIL multiple advantages, this verification technique is not suitable for hardware/software co-design. One of the contributions of this paper is to deal with HIL weaknesses by developing a new standard target based on the FPGA board [29, 33]. Shortcomings could be resumed as:

- Only a few targets are supported.
- Lack of reuse and portability: the model associated with the Control Unit (CU) target block set decreases reusability and portability.
- Unchangeable HW: the HW is created based on the generated code that prevents the modification of HW architecture.

- Waste of time: When the model related to the CU changes, the HIL technique platform requires to be performed again.

C. HSIL Co-Simulation Technique

In line with [29], the control part was an embedded system defined by SW and HW components. Each part had to be aware of the other part's characteristics to obtain optimized components. The most adequate strategy was the co-design, since it was expected to develop HW/SW components concurrently. The main idea was to set the HSIL to overcome its limits, as it was recommended in [29, 32-33]. The described method could improve HIL, having the following advantages:

- Expanding the HIL to attend the co-design strategy.
- The S-function can be integrated into the Simulink block to perform specific communication with the FPGA board. This is a very powerful way to extend the capabilities of Simulink, guaranteeing reusability and portability.
- Without modifying the architecture, the CU modification and verification is enabled.

The main advantage of the proposed method is the synchronization mechanism that ensures the relationship between the Simulink/Matlab simulator and the HW architecture emulated on the FPGA board. The aim of the flexible robot manipulator control design was to verify the CUs. The model had a speed controller, a torque controller, and current controllers were considered as SW applications. Three steps were crucial for the implementation of the controller units [29-30,34]:

- Target architecture: The verification needs several tools to satisfy continuous and discrete models. The discrete model was implemented on the ALTERA DE2-70 FPGA board, using the Quartus II, NIOSII IDEs. MATLAB was used for the continuous model.
- Synchronization block integration: This block ensures communication and synchronization between the Matlab/Simulink simulator and the target architecture described in S-function block. The endpoints related to the synchronization interface represent the input signals of the continuous model. When an activation signal is triggered, the synchronization block ensures a context switch to the board by sending a data packet. Upon resuming execution, the data packet read by the interface synchronization defines the next activation time.
- Simulation results: The signals' update events were not periodic, so the simulation was performed using the synchronization scheme.

IV. EXPERIMENTS

Figures 4 and 5 represent the design in the Matlab/Simulink environment associated to the MIL and HIL techniques respectively. Controllers were replaced with the HIL block, and the control algorithm was computed on the board. The HIL technique was performed using the DSP builder tool provided by Matlab/Simulink. Figure 6 shows the different characteristics mounted. A Cyclone version of FPGA was used, based on architecture done in Quartus II (BDCM DSPbuilder was the file architecture). As shown in Figure 6, the compilation passed through four steps: Analyze, Synthesis, Fitter, and Program. Figures 7 and 8 describe HIL technique's parameters.

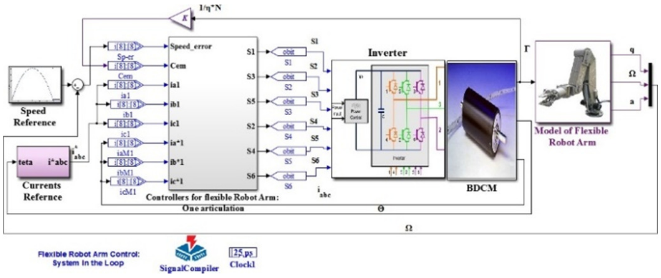


Fig. 4. Flexible robot arm control using DSP Builder blocks.

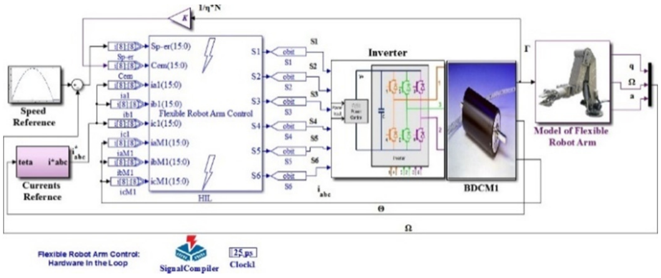
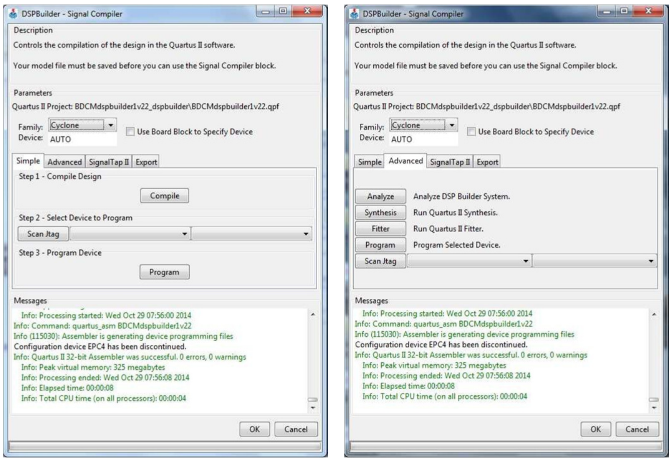


Fig. 5. Flexible robot arm control using HIL.


DSPBuilder - Signal Compiler

Description

Controls the compilation of the design in the Quartus II software.

Your model file must be saved before you can use the Signal Compiler block.

Parameters

Quartus II Project: BDCMdspbuilder1v22_dspbuilder\BDCMdspbuilder1v22.qpf

Family: Cyclone

Device: AUTO

☐ Use Board Block to Specify Device

Simple | Advanced | SignalTap II | Export

Step 1 - Compile Design

Compile

Step 2 - Select Device to Program

Scan Jtag

Step 3 - Program Device

Program

Messages

```
  Info: Processing started: Wed Oct 29 07:56:00 2014
Info: Command: quartus_asm BDCMdspbuilder1v22
Info (115030): Assembler is generating device programming files
Configuration device EPC4 has been discontinued.
Info: Quartus II 32-bit Assembler was successful. 0 errors, 0 warnings
  Info: Peak virtual memory: 325 megabytes
  Info: Processing ended: Wed Oct 29 07:56:08 2014
  Info: Elapsed time: 00:00:08
  Info: Total CPU time (on all processors): 00:00:04
```

OK Cancel

DSPBuilder - Signal Compiler

Description

Controls the compilation of the design in the Quartus II software.

Your model file must be saved before you can use the Signal Compiler block.

Parameters

Quartus II Project: BDCMdspbuilder1v22_dspbuilder\BDCMdspbuilder1v22.qpf

Family: Cyclone

Device: AUTO

☐ Use Board Block to Specify Device

Simple | Advanced | SignalTap II | Export

Analyze — Analyze DSP Builder System.

Synthesis — Run Quartus II Synthesis.

Fitter — Run Quartus II Fitter.

Program — Program Selected Device.

Scan Jtag

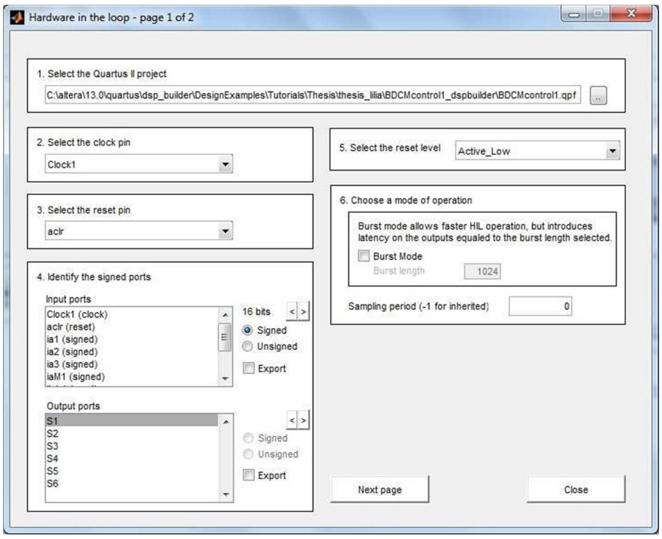
Messages

```
  Info: Processing started: Wed Oct 29 07:56:00 2014
Info: Command: quartus_asm BDCMdspbuilder1v22
Info (115030): Assembler is generating device programming files
Configuration device EPC4 has been discontinued.
Info: Quartus II 32-bit Assembler was successful. 0 errors, 0 warnings
  Info: Peak virtual memory: 325 megabytes
  Info: Processing ended: Wed Oct 29 07:56:08 2014
  Info: Elapsed time: 00:00:08
  Info: Total CPU time (on all processors): 00:00:04
```

OK Cancel

Fig. 6. Compilation using the DSP BUILDER.

The simulation results for the three techniques, MIL using Simulink, HIL using DSP Builder tools and Altera FPGA board, and the HSIL using S-function and NIOSII were compared. The implementation steps of the HSIL techniques are described in [33]. Figures 9-11 show speed evolution, position, speed and position error, and the control law respectively using MIL, HIL, and HSIL techniques. However, as the curves in the MIL and HSIL are similar, only the curves using MIL are presented. As it can be observed, the flexible robot arm followed the desired trajectory with a tiny static error rate, confirming the robustness of the controllers orienting the flexible robot arm to the desired position respecting a definite speed. Moreover, when the desired final position is reached, the electromagnetic torque is equal to the load torque. In addition, it would be worth noting that the controllers applied on a flexible robot arm driven by the BDCM yielded an acceptable performance using all techniques. Therefore, the desired speeds and accelerations were reached.



Hardware in the loop - page 1 of 2

1. Select the Quartus II project
C:\altera\13.0\quartus\dsp_builder\DesignExamples\Tutorials\Thesis\thesis_lilia\BDCMcontrol1_dspbuilder\BDCMcontrol1.qpf

2. Select the clock pin: Clock1

3. Select the reset pin: aclr

4. Identify the signed ports

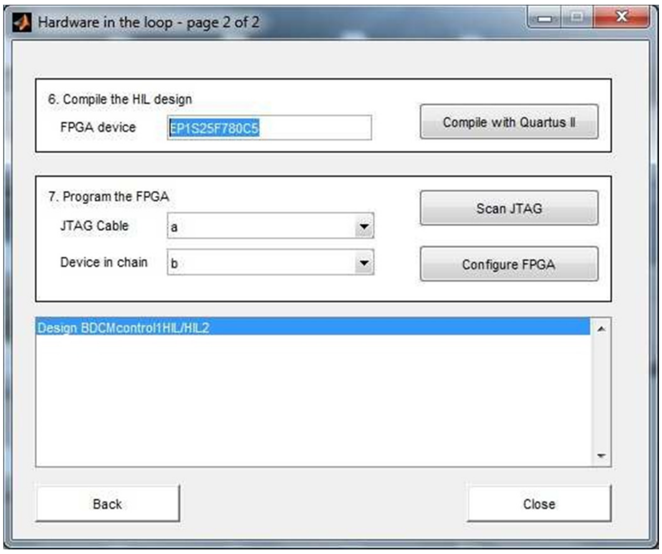
Input ports: Clock1 (clock), aclr (reset), ia1 (signed), ia2 (signed), ia3 (signed), iaM1 (signed)
16 bits ● Signed ○ Unsigned □ Export

Output ports: S1, S2, S3, S4, S5, S6
○ Signed ○ Unsigned □ Export

5. Select the reset level: Active_Low

6. Choose a mode of operation
Burst mode allows faster HIL operation, but introduces latency on the outputs equaled to the burst length selected.
□ Burst Mode
Burst length: 1024
Sampling period (-1 for inherited): 0

Fig. 7. Selection of the HIL configuration.



Hardware in the loop - page 2 of 2

6. Compile the HIL design
FPGA device: EP1S25F780C5

7. Program the FPGA
JTAG Cable: a
Device in chain: b

Design BDCMcontrol1HIL/HIL2

Fig. 8. The interface of the HIL program.

Table I shows the simulation time for the hysteresis, PID, and torque controllers and all systems using MIL, HIL, and HSIL techniques. Table I shows that:

- The output signals obtained using HIL have an error compared to the output signals using MIL and HSIL. At first, this error was not expected as the same control algorithm was used in both cases. After a deep research it was concluded that this error happened due to the insufficient memory of the FPGA board. This lack of memory led to data corruption which affected the computation results.
- Satisfactory performance is obtained to achieve the desired speeds and positions at a lower cost.
- The presence of a precision loss using HIL compared to simulation results using MIL, due to the high sampling time. This caused the failure to get all the points for more accuracy, while it allowed having a faster system.

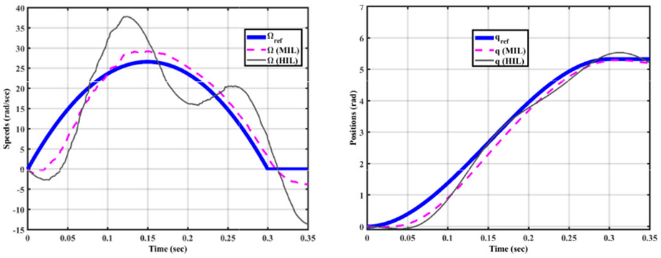


Fig. 9. Speed and position.

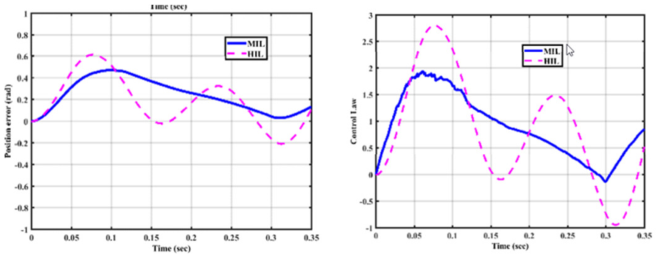


Fig. 10. Speed error and position error.

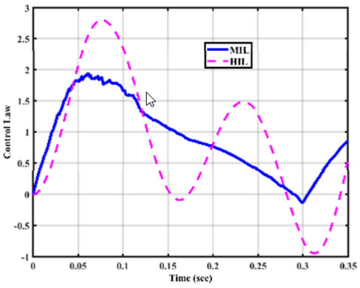


Fig. 11. Control law.

TABLE I. SIMULATION TIME

Parts	Simulation with MIL (s)	Simulation with HIL (s)	Simulation with HSIL (s)
Hysteresis controller	48	12	6
Torque controller	16	4	2
PID controller	24	6	3
Flexible robot arm control	8424	2107	1053.5

After verifying the control algorithm of the flexible robot arm driven by a BDCM using MIL, the HIL verification results were closer to the real case. This could enable a faster run of verification scenarios than with the software simulation. Following the previously described design methods, HIL simulation was much faster and more confident than software simulation. In addition, the refinement of the development and the rapidity of the control strategy were continued by the combination of the HSIL co-simulation, as it benefited from each technique's advantages. These results proved that HIL simulation was 4 times faster than software simulation. Furthermore, the HSIL simulation was twice faster than the HIL simulation, proving that HSIL is the best verification method.

V. Conclusion

This paper aimed to evaluate three Computer-Aided Engineering techniques in robotic systems, focusing on a BDCM control fitted to a flexible robot arm and simulating in Matlab/Simulink environment via verification methods. The model and the control of the BDCM-inverter flexible robot arm were presented. Then, different control strategies such as a speed controller, a torque controller and a hysteresis controller for controlling currents were described. MIL, HIL, and HSIL simulations were studied and analyzed, showing the HSIL's benefits. According to simulation results via the Simulink environment and the FPGA board, the HSIL achieves the highest accuracy and speed than MIL and HIL techniques. In addition, HSIL provides reusability and portability of the control unit compared to the other techniques.

References

[1] S. Li, A. Samin, Y. F. Zheng, and L. Cao, "The effects of radiation-induced demagnetization on the performance of the brushless DC motor in robot servo systems," in *2014 International Symposium on Fundamentals of Electrical Engineering (ISFEE)*, Nov. 2014, pp. 1–6, https://doi.org/10.1109/ISFEE.2014.7050562.

[2] S. S. Patel *et al.*, "Modeling and implementation of intelligent commutation system for BLDC motor in underwater robotic applications," in *2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, Jul. 2016, pp. 1–4, https://doi.org/10.1109/ICPEICES.2016.7853695.

[3] L. Zouari, A. B. Rhouma, and M. Abid, "On the Potentialities of Reduced Structure Inverter Integrated in Robot Application," in *Proceedings of the World Congress on Engineering*, London, UK, Jul. 2012, vol. II.

[4] Lin Bai, "Electric drive system with BLDC motor," in *2011 International Conference on Electric Information and Control Engineering*, Apr. 2011, pp. 359–363, https://doi.org/10.1109/ICEICE.2011.5777507.

[5] D. Chowdhury, A. Mitra, S. Mondal, and M. Chattopadhyay, "A New Implementation Scheme in Robotic Vehicle Propulsion Using Brushless DC Motor," in *Proceedings of the First International Conference on Intelligent Computing and Communication*, Singapore, 2017, pp. 387–394, https://doi.org/10.1007/978-981-10-2035-3_39.

[6] K. Rajagopal, S. Vaidyanathan, A. Karthikeyan, and P. Duraisamy, "Dynamic analysis and chaos suppression in a fractional order brushless DC motor," *Electrical Engineering*, vol. 99, no. 2, pp. 721–733, Jun. 2017, https://doi.org/10.1007/s00202-016-0444-8.

[7] H. Hu, T. Wang, S. Zhao, and C. Wang, "Speed control of brushless direct current motor using a genetic algorithm–optimized fuzzy proportional integral differential controller -, 2019," *Advances in Mechanical Engineering*, vol. 11, no. 11, Nov. 2019, https://doi.org/10.1177/1687814019890199.

[8] H. Melkote and F. Khorrami, "Nonlinear adaptive control of direct-drive brushless DC motors and applications to robotic manipulators," *IEEE/ASME Transactions on Mechatronics*, vol. 4, no. 1, pp. 71–81, Mar. 1999, https://doi.org/10.1109/3516.752086.

[9] N. Hemati, "Strange attractors in brushless DC motors," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 41, no. 1, pp. 40–45, Jan. 1994, https://doi.org/10.1109/81.260218.

[10] C. L. Lin, H. Y. Jan, and N. C. Shieh, "GA-based multiobjective PID control for a linear brushless DC motor," *IEEE/ASME Transactions on Mechatronics*, vol. 8, no. 1, pp. 56–65, Mar. 2003, https://doi.org/10.1109/TMECH.2003.809136.

[11] S. Ahmed, A. Topalov, N. Dimitrov, and E. Bonev, "Industrial implementation of a fuzzy logic controller for brushless DC motor drives using the PicoMotion control framework," in *2016 IEEE 8th International Conference on Intelligent Systems (IS)*, Sep. 2016, pp. 629–634, https://doi.org/10.1109/IS.2016.7737493.

[12] X. Zhou, X. Chen, M. Lu, and F. Zeng, "Rapid Self-Compensation Method of Commutation Phase Error for Low- Inductance BLDC Motor," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1833–1842, Aug. 2017, https://doi.org/10.1109/TII.2017.2653812.

[13] J. Iqbal, "Modern Control Laws for an Articulated Robotic Arm: Modeling and Simulation," *Engineering, Technology & Applied Science Research*, vol. 9, no. 2, pp. 4057–4061, Apr. 2019, https://doi.org/10.48084/etasr.2598.

[14] M. B. Ayed, L. Zouari, and M. Abid, "Software In the Loop Simulation for Robot Manipulators," *Engineering, Technology & Applied Science Research*, vol. 7, no. 5, pp. 2017–2021, Oct. 2017, https://doi.org/10.48084/etasr.1285.

[15] C. Obermaier, R. Riebl, C. Facchi, A. Al-Bayatti, and S. Khan, "Limitations of HIL Test Architectures for Car2X Communication Devices and Applications," presented at the CSCS '19: ACM Computer Science in Cars Symposium, Kaiserslautern Germany, Oct. 2019, https://doi.org/10.1145/3359999.3360493.

[16] S. M. Shah and M. Irfan, "Embedded hardware/software verification and validation using hardware-in-the-loop simulation," in *Proceedings of the IEEE Symposium on Emerging Technologies, 2005.*, Sep. 2005, pp. 494–498, https://doi.org/10.1109/ICET.2005.1558931.

[17] J. C. V. S. Junior, A. V. Brito, and T. P. Nascimento, "Testing Real-Time Embedded Systems with Hardware-in-the-Loop Simulation Using High Level Architecture," in *2015 Brazilian Symposium on Computing Systems Engineering (SBESC)*, Foz do Iguacu, Brazil, Nov. 2015, pp. 142–147, doi: 10.1109/SBESC.2015.34.

[18] U. R. F. Souza, J. K. Sperb, B. A. de Mello, and F. R. Wagner, "Tangram - virtual integration of heterogeneous IP components in a distributed co-simulation environment," in *16th Symposium on Integrated Circuits and Systems Design, 2003. SBCCI 2003. Proceedings.*, Sep. 2003, pp. 125–130, https://doi.org/10.1109/SBCCI.2003.1232817.

[19] T. W. B. Silva *et al.*, "A Distributed Functional Verification Environment for the Design of System-on-Chip in Heterogeneous Architectures," in *2018 31st Symposium on Integrated Circuits and Systems Design (SBCCI)*, Aug. 2018, pp. 4849–4854, https://doi.org/10.1109/SBCCI.2018.8533247.

[20] A. V. Brito, A. V. Negreiros, C. Roth, O. Sander, and J. Becker, "Development and Evaluation of Distributed Simulation of Embedded Systems Using Ptolemy and HLA," in *2013 IEEE/ACM 17th International Symposium on Distributed Simulation and Real Time Applications*, Oct. 2013, pp. 189–196, https://doi.org/10.1109/DS-RT.2013.28.

[21] J. Claudio, V. S. Junior, A. V. Brito, and T. P. Nascimento, "Verification of Embedded System Designs through Hardware-Software Co-Simulation," *International Journal of Information and Electronics Engineering*, vol. 5, no. 1, pp. 68–73, 2015, https://doi.org/10.7763/IJIEE.2015.V5.504.

[22] J. Chen, W. Yu, Y. Zhang, and C. Du, "Design and Implementation of Universal Configurable Digital Emulators," in *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, May 2019, pp. 943–947, https://doi.org/10.1109/ITAIC.2019.8785587.

[23] T. W. B. Silva, D. C. Morais, H. G. R. Andrade, A. M. N. Lima, E. U. K. Melcher, and A. V. Brito, "Environment for integration of distributed heterogeneous computing systems," *Journal of Internet Services and Applications*, vol. 9, no. 1, Jan. 2018, Art. no. 4, https://doi.org/10.1186/s13174-017-0072-1.

[24] G. Das, M. De, and K. K. Mandal, "Design of Flyback Converter by Using an Ideal Switch and a MOSFET Switch," in *2018 IEEE Electron Devices Kolkata Conference (EDKCON)*, Nov. 2018, pp. 110–114, https://doi.org/10.1109/EDKCON.2018.8770407.

[25] M. Mauri, "Hardware in the Loop Simulation of Renewable Distributed Generation Systems," in *Renewable Energy*, IntechOpen, 2009, p. 133.

[26] S. Lentijo, S. D'Arco, and A. Monti, "Comparing the Dynamic Performances of Power Hardware-in-the-Loop Interfaces," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 4, pp. 1195–1207, Apr. 2010, https://doi.org/10.1109/TIE.2009.2027246.

[27] G. Enriquez, "Design of a Hardware-in-the-Loop (HIL) Simulation of a Small Unmanned Aerial Vehicle (sUAV) for Self Autonomous Mission Capability," in *Student Research Symposium (SRS)*, 2019, [Online]. Available: https://commons.erau.edu/db-srs/2019/poster/19.

[28] C. Graf, J. Maas, T. Schulte, and J. Weise-Emden, "Real-time HIL-simulation of power electronics," in *2008 34th Annual Conference of IEEE Industrial Electronics*, Nov. 2008, pp. 2829–2834, https://doi.org/10.1109/IECON.2008.4758407.

[29] M. B. Ayed, F. Bouchhima, and M. Abid, "A novel verification technique for control units," *International Journal of Engineering and Technology*, vol. 5, no. 2, pp. 1990-1999, 2013.

[30] L. Zouari, M. B. Ayed, and M. Abid, "A hardware in the loop simulation for electrically driven robot manipulator," in *2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE)*, May 2015, pp. 1495–1501, https://doi.org/10.1109/CCECE.2015.7129502.

[31] L. Zouari, M. B. Ayed, and M. Abid, "Embedded control of robot arm driven by Brushless DC motor on FPGA," in *2014 Second World Conference on Complex Systems (WCCS)*, Nov. 2014, pp. 722–727, https://doi.org/10.1109/ICoCS.2014.7060985.

[32] L. Zouari, M. B. Ayed, and M. Abid, "Improved performance of a brushless DC motor using hardware in the loop control technique," in *2015 IEEE 12th International Multi-Conference on Systems, Signals Devices (SSD15)*, Mar. 2015, pp. 1–6, https://doi.org/10.1109/SSD.2015.7348100.

[33] M. B. Ayed, F. Bouchhima, L. Zouari, and M. Abid, "An Accelerated Hardware Software in The Loop Technique for Control Units," *International Journal of Mechanical & Mechatronics Engineering*, vol. 14, no. 3, pp. 10–22, 2014.

[34] M. B. Ayed, F. Bouchhima, and M. Abid, "CODIS+: Co-simulation environment for heterogeneous systems," *Journal of Control Engineering and Applied Informatics*, vol. 20, no. 1, pp. 98-107, Mar. 2018.