Software In the Loop Simulation for Robot Manipulators

Mossaad Ben Ayed
CES Laboratory, Sfax University, Tunisia
and Al Majmaah University, KSA
mm.ayed@mu.edu.sa,
mossaad_benayed@yahoo.fr

Lilia Zouari
CES Laboratory
Sfax University, Tunisia
lilia6enis@yahoo.fr

Mohamed Abid
CES Laboratory
Sfax University, Tunisia
mohamed.abid@enis.rnu.tn

***Abstract*—In the last decades, the classical verification of robotic software component is postponed until the code is developed enough to function in real hardware. For this reason, the verification of code at early stages is essential to reduce development costs and necessary time for embedded systems such as robot manipulator. Therefore, Software In the Loop (SIL) simulation may be realized in the early stages of software development. It offers the possibility to execute tests before the hardware is available and thus detect errors. In this paper, we propose a Software In the Loop (SIL) test for robot manipulator driven by a Brushless DC Motor without a target system hardware. Simulation results prove the rapidity and the good performance of the developed code for the controller's part by the validation of the behavior of robot manipulator software.**

Keywords-software in the loop (SIL); robot manipulator; controllers; S-function

I. INTRODUCTION

The engineering of software and digital electronics represent an important place in the process of the design and the verification of the electronic systems [1, 2]. The levels of electronic components integration, the complexity and the size of embedded code are continually growing and this growth is accompanied by great sensitivity to the disturbance and uncertainties [3, 4]. The number of possible combinations states poses a problem of combinatorial explosion that led to huge numbers of possible states due to the complexity of mechatronic systems [5, 6]. This complexity is a challenge for dependability [1, 5, 7]. Indeed, new robotic applications representing the most known mechatronic system have required the lightest and most precise robots that can be driven with small amounts of energy [8-10]. Unfortunately, the flexibility of these robots leads to an oscillating behavior at the end of the link. Obtaining a precise trajectory tracking is an arduous task that requires a complex control in closed-loop [11-14]. To achieve the control objectives, such as the accuracy of the position and vibration suppression, many control techniques were applied to these flexible joint manipulators [15-18]. Thus, we consider the entire robotic system including actuator dynamics and flexibility in the joint. To guarantee a high performance of the control system, a flexible joint manipulator which is highly nonlinear, strongly coupled and uncertain is used [19-21]. The embedded systems in the robotic industry must accomplish high safety requirements [22]. Therefore, verification step is essential to verify the behavior of the proposed system and to detect and correct the errors at earlier stage without any hardware or physical environment [23].

This paper presents the SIL simulation technique for robot manipulator driven by Brushless DC Motor (BDCM). In section I, some related works were detailed. The second section described the manipulator control system. The third section defined the SIL simulation technique. The fourth section was devoted for SIL implementation method shown the speed and torque controller and the current hysteresis controller. Finally, simulation results in MATLAB/Simulink are shown in Section V. Extracted simulation time prove the role of the SIL technique to accelerate the simulation and prevent the risks.

II. RELATED WORKS

The Software In the Loop (SIL) approach allow us to integrate software component with an environment simulation [23-24]. Besides, this approach allows testing different scenarios and control algorithms very quickly and benefit from its flexibility. In [25], authors realized a SIL environment for the development of control strategies for hybrid vehicles. This environment allows the evaluation of different vehicle dynamic behavior for each additional masses and each position of gravity center of the vehicle. In [26], authors described the SIL validation of a flight control system needed to develop low cost Unmanned Aerial Vehicles (UAV). Moreover, this technique allows modeling, linearizing and decoupling of the highly nonlinear six degrees-of-freedom dynamics of the vehicle. In [27], authors presented a low cost SIL simulation to evaluate performance of control law for quadricopter using MATLAB/Simulink for control law and X-plane for vehicle dynamics and flight environment. In [22], authors described a cost effective SIL simulation methodology for mobile nodes representing large-scale pervasive system. This methodology permits deploying the real code once and replicating environments and over virtual devices as often as they wish. In [28], authors used the High Level Simulation Framework Open Virtual Platforms (OVP) for SIL simulation of embedded control applications to benefit from rapid prototyping in early stages of the development cycle. This approach was applied for two cases which are the motor control application and the image processing application. In [29], authors described SIL simulators for autonomous

navigation algorithms of planetary rovers due to temporarily unavailable hardware. This simulator is necessary to facilitate the continued algorithms development even in the absence of hardware. In [30], authors studied SIL technique for flying robots. In fact, the high production expenses need the validation of such system by simulation to predict the robot reactions in various flight modes. This technique was realized using MATLAB for control part and ADAMS for structural design.

Nowadays, validating complex systems in early stages needs a good simulation technique. This paper proposes the SIL simulation for robot arm driven by BDCM. The development of robot control algorithms for robot arm driven by BDCM often hinges on access to robot hardware that its access is usually very limited. In order to facilitate the continued development of these algorithms, the SIL simulation was used to integrate each algorithm change at low cost.

III. MANIPULATOR CONTROL SYSTEM

To implement control algorithms for the proposed manipulator, it is essential to check and study the performance of the manipulator in the MATLAB/Simulink environment. In fact, we focus on the implementation of the dynamic model of the proposed manipulator taking into consideration the controllers studied in the MATLAB/Simulink. Figure 1 presents the block diagram of the proposed manipulator.

Based on Figure 1, we distinguish three blocks describing the proposed system:

- The model of the system,
- The controllers,
- The reference trajectory.

IV. SIL TECHNIQUE

Designing a system with its controller requires mathematical model followed by adjusted certain parameters. Recently, the X In the Loop approach is often used to check the system in the middle stage of the design [31]. This approach is an effective way to produce a fast and reliable short-time system. In the X In the Loop approach, there are two levels of the simulation configurations based on the PC shown in Figure 2: Model In The Loop (MIL) and SIL [31]. Each configuration provides certain levels of progress and reduced the gap in the development process between the mathematical model and the proposed platform [31, 32]. Indeed, these techniques are essential to predict possible risks before the implementation phase and to accelerate the simulation time. Thus a check is made by exploiting several verification techniques. MIL simulation refers to the model verification in the modeling environment such as MATLAB/Simulink. Check system at MIL means that the model and its controllers are simulated in the modeling framework without physical hardware. When checking the implementation of complex systems models, it is necessary to distinguish between the verification of modules and the check of the overall system. This technique verifies the system at an earlier stage of the realization. In this step, any problem of the mathematical model of the controller can be quickly found and fixed [33].

In SIL simulation, the used model is replaced by executable code running on the same computer platform of a fixed-point arithmetic way. This step allows the developer to quickly find some bad choices of memory size and to ameliorate the controller code performance. Checking system in SIL level means that embedded software is verified in a simulation environment, but without any hardware such as mechanical or hydraulic components or sensors or actuators [33]. Normally, these two steps are performed using a single integrated platform in a PC [33].

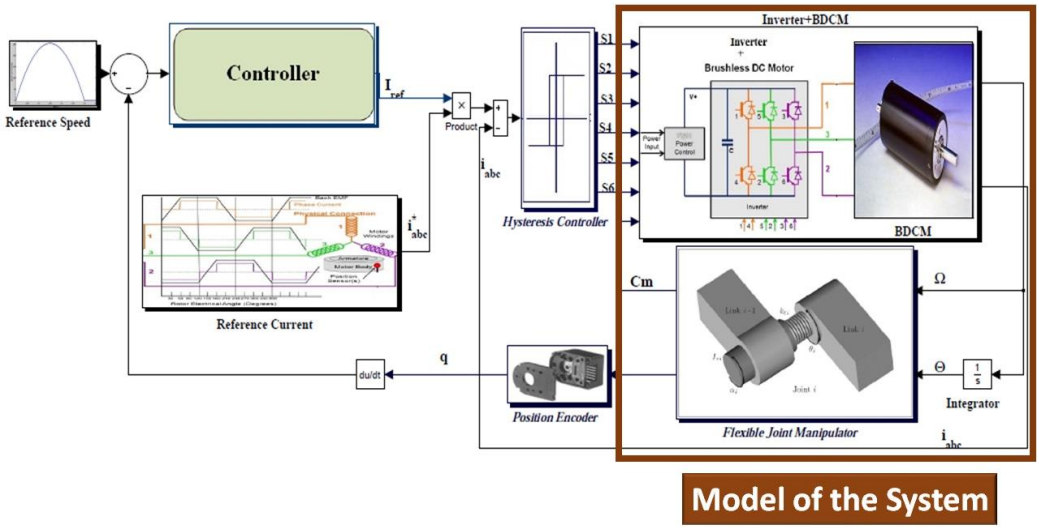

Fig. 1. Block diagram of the proposed manipulator

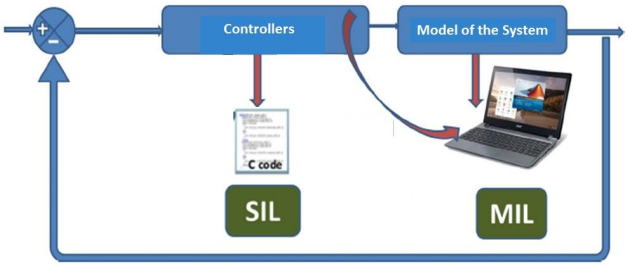

Fig. 2. Different simulation techniques

V. SIL IMPLEMENTATION METHOD

After the algorithms in MIL stage operate as designed, the model of the controller can be converted into C code with a coder as Simulink Coder and TargetLink. The exported code can as well be connected to the external sources of code. The created code can first be tested in Simulink and also communicated with other systems and subsystems. In this section, we will present the modeling of the control part and simulation via SIL simulation. In SIL, the used model in MIL technique is replaced with executable code running on the same computer platform in a fixed-point arithmetic way. This step is performed using an integrated platform in a single computer. Figure 3 presents the block diagram of the proposed manipulator replacing the control block by the adequate S-function block to communicate with the executable code as well as an image of the C code area. To perform a SIL simulation, the flexible joint manipulator with brushless DC motor must be modeled. The control part to be replaced by the equivalent C code is articulated on two controllers that are the speed and torque controller and the current controller. Stills from the fuzzy logic PI controller code and hysterisis controller in C code are shown in Figure 4. For the speed and torque control part, the inputs are: the speed error (rad/s) and the position error (rad) and the output is the amplitude of the reference current. In order to obtain a compromise in performance between the simulation time and the reliability of the studied controllers and simulation results on previous works [34-25], we adopt the fuzzy PI controller. For the current control part that uses the hysteresis controller, the inputs are the current errors in the 3 phases (A) and the outputs are the states of IGBTs (S1-6) (1 or 0). In this part, the states of IGBTs applied to the inverter through the hysteresis controller will be verified through the SIL simulation replacing it by its equivalent C code.

VI. SIMULATION RESULTS

In order to implement these control algorithms of the studied manipulator, it is essential to check and study as a first step the controllers of the manipulator driven by BDCM in MATLAB/Simulink environment. To evaluate the effectiveness and performances of SIL simulation technique for different controllers, we are interested to the following principal variables: speed (rad/sec), position (rad), position error (rad), speed error (rad/sec), control law and simulation time. The system parameters are given in Table I. Figure 5 represents respectively the evolution of the speed, the position, the error of the position and speed and the electromagnetic torque in the case of fuzzy logic PI controller. To evaluate the performance of the SIL technique, we adopted the simulation results given on the MATLAB/Simulink environment specifying the execution time of the various controllers' codes. After checking the control algorithm in MATLAB/Simulink, SIL technique produces the same results in the case of MIL simulation by reaching the desired speed and position. The simulation time is decreased compared to those of MIL simulation technique. Table II represents a simulation time recapitulation for each system for both MIL and SIL simulation techniques.

TABLE I. SYSTEM PARAMETERS

Symbol	Description	Numeric value
R	Resistance	0.625 Ω
L	Inductance	1.595e-3 H
J_m	du Motor inertia	1e-5Kg/m^2
m_l	Manipulator mass	0.8619 Kg
l_l	Length of manipulator	0.3 m
J1	Inertia of the manipulator	0.0065 N/m^2
N	Reduction ratio	74
η	Transmission efficiency	0.72
f	Friction	1.164e-3 Kgm/s^2
K_t	Torque constant	0.0382
K_E	Electromotive constant	0.0382

TABLE II. SIMULATION TIME OF SYSTEMS FOR MIL AND SIL SIMULATION TECHNIQUES

System	MIL	SIL
Motor	6 mn 40 sec	4 mn 40 sec
Manipulator	18 h 9 mn 34 sec	15 h 13 mn 55 sec

VII. CONCLUSION

This work was developed within the SIL simulation technique for the verification of the proposed robot manipulator driven by Brushless DC Motor at an earlier stage. In such a complex system, the SIL simulation can guarantee the verification of software component i.e. C code without any hardware and benefit from its flexibility to validate different controller algorithms. So, the manipulators including their controllers have been presented. Then, the SIL technique has been defined. Also, SIL implementation method has been detailed. Finally, simulation results have shown that the SIL simulation technique leads to high performance at low cost and time and allows detecting errors code before implementation.

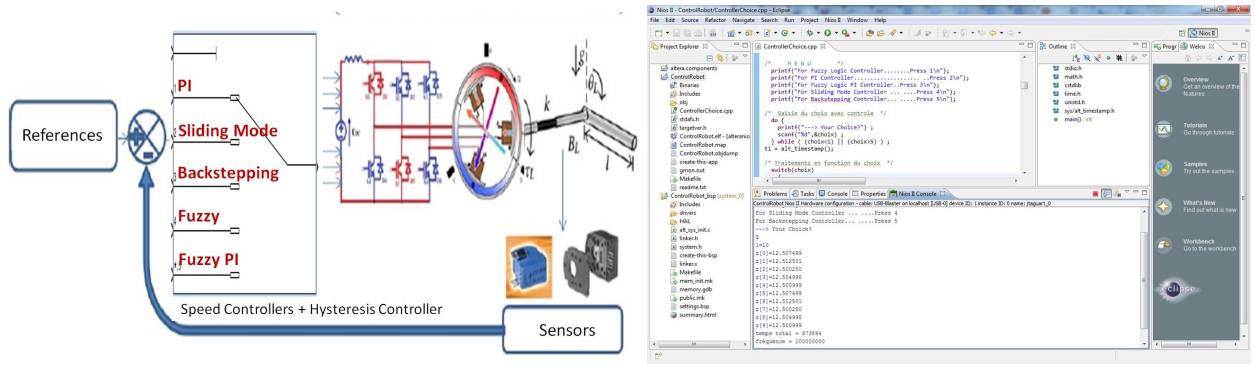

Fig. 3. Different speed and torque controllers opted for the proposed manipulator (left) and the code area (right)

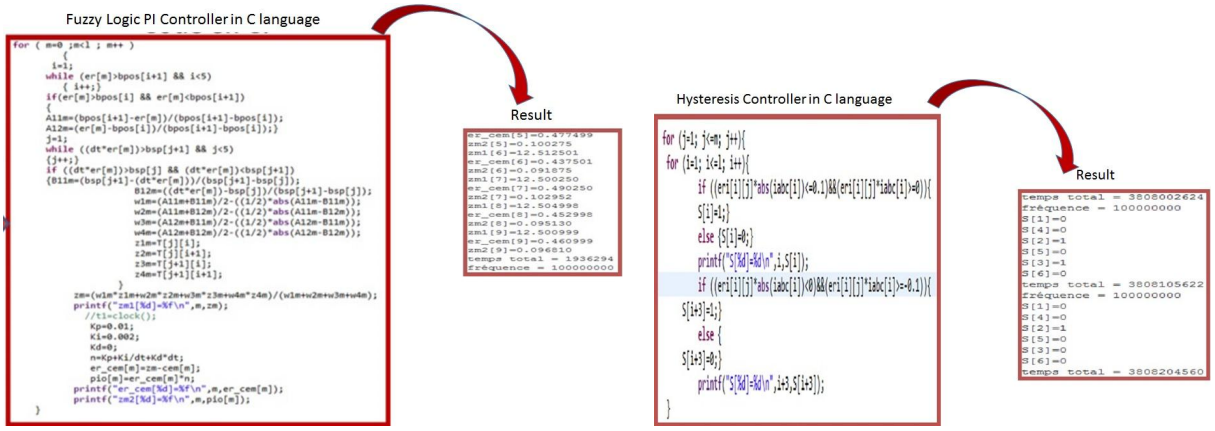

Fig. 4. Fuzzy logic PI controller code (left) and hysterisis controller in C code (right)

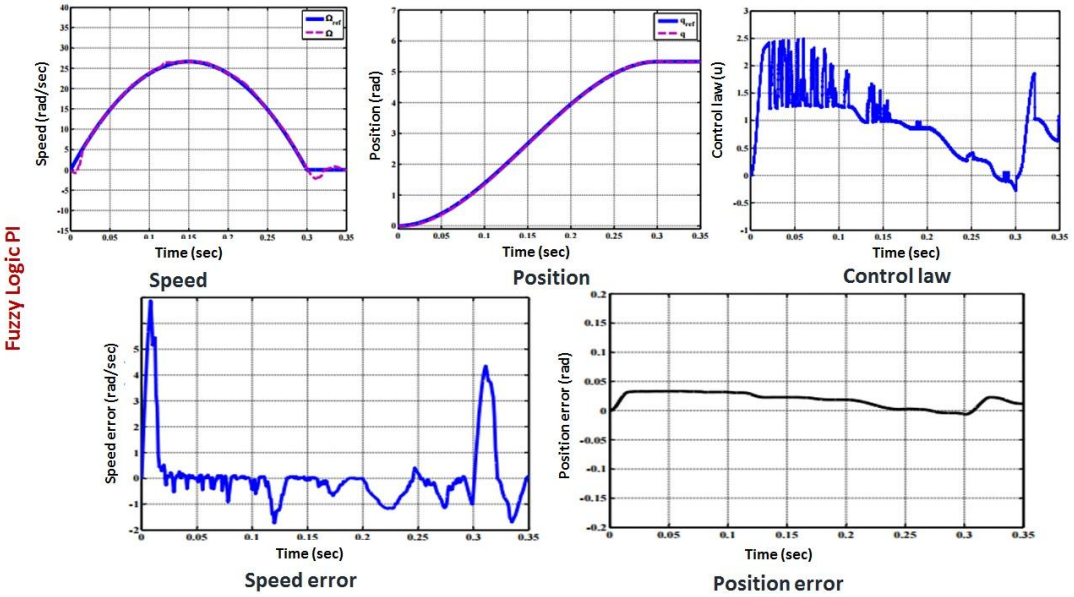

Fig. 5. Simulation results for fuzzy logic PI controller

REFERENCES

[1] L. Zouari, M. B. Ayed, M. Abid, "Embedded control of robot arm driven by brushless dc motor on fpga", Complex Systems (WCCS), Second World Conference, pp. 722–727, 2014

[2] J. de J Lozoya-Santos, R. Morales-Menendez, R. A Ramirez-Mendoza, "Evaluation of onoff semi-active vehicle suspension systems by using the hardware-in-the-loop approach and the software-in-the-loop approach", Journal of Automobile Engineering, Vol. 229, No. 1, 2013

[3] H. Belhadaoui, Safe design of intelligent mechatronic systems for critical applications, PhD Thesis, Lorraine Polytechnic National Institute, 2011

[4] D. Meola, L. Iannelli, L. Glielmo, "Flight control system for small-size unmanned aerial vehicles: Design and software-in-the-loop validation", Control & Automation (MED) 21st Mediterranean Conference, pp. 357–362, 2013

[5] L. Zouari, M. Ben Ayed, M. Abid, "Improved performance of a brushless dc motor using hardware in the loop control technique", 12th International Multi-Conference on Systems, Signals Devices (SSD), Tunisia, pp. 1–6, March 16-19, 2015

[6] H. Vardhan, B. Akin, H. Jin, "A low-cost, high-fidelity processor-in-the loop platform: For rapid prototyping of power electronics circuits and motor drives" IEEE Power Electronics Magazine, Vol. 3, No. 2, pp. 18–28, 2016

[7] S. Jeong, Y. Kwak, W. Jin Lee, "Software-in-the-loop simulation for early-stage testing of autosar software component", 8th International Conference on Ubiquitous and Future Networks (ICUFN), pp. 59–63, Austria, July 5-8, 2016

[8] J. F. Broenink, Y. Ni, "Model-driven robot-software design using integrated models and co-simulation", International Conference on Embedded Computer Systems (SAMOS), pp. 339–344, Greece, July 16-19, 2012

[9] J. F. Broenink, Y. Ni, A. Marcel, "Groothuis. On model-driven design of robot software using co-simulation", Proceedings of SIMPAR 2010 Workshops, International Conference on Simulation, Modeling and Programming for Autonomous Robots, Germany, pp. 659–668, 2010

[10] X. Chen, M. Salem, T. Das, X. Chen, "Real time software-in-the-loop simulation for control performance validation", Simulation, Vol. 84, No. 8-9, pp. 457–471, 2008

[11] A. S. Damstra, Virtual prototyping through co-simulation in hardware/software and mechatronics co-design, MSc Thesis, University of Twente, 2008

[12] D. Garlan, Software architecture: a roadmap, in: The Future of Software Engineering, A. Finkelstein, ed., ACM Press, 2000

[13] J. C. V. S. Junior, A. V. Brito, 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

[14] A. Bittar, H. V. Figuereido, P. A. Guimaraes, A. C. Mendes, "Guidance software-in-the-loop simulation using x-plane and simulink for uavs", International Conference on Unmanned Aircraft Systems (ICUAS), pp. 993–1002, USA, May 27-30, 2014

[15] A. Kalavade, E. A Lee, "A hardware-software codesign methodology for dsp applications", IEEE Design Test of Computers, Vol. 10, No. 3, pp. 16–28, 1993

[16] P. Mannar Mannan, Framework for the design and implementation of software defined radio based wireless communication system, MSc Tthesis, University of Akron, 2005

[17] M. Ben Ayed, F. Bouchhima, L. Zouari, M. Abid, "An accelerated hardware software in the loop technique for control units", International Journal of Mechanical & Mechatronics Engineering, Vol. 14, No. 3, pp. 10–21, 2014

[18] M. Muresan, D. Pitica, "Software in the loop environment reliability for testing embedded code", IEEE 18th International Symposium on Design and Technology in Electronic Packaging, pp. 325–328, Romania, October 25-28, 2012

[19] J. Ou, V. K. Prasanna, "Matlab/simulink based hardware/software co-simulation for designing using fpga configured soft processors",19th IEEE International Parallel and Distributed Processing Symposium (IPDPS-05), USA, April 4-8, 2005

[20] E. Stoy, Z. Peng, "Inter-domain movement of functionality as a repartitioning strategy for hardware/software co-design", Journal of Systems Architecture, Vol. 43, No. 1-5, pp. 87–98, 1997

[21] S. M. Shah, M. Irfan, "Embedded hardware/software verification and validation using hardware-in-the-loop simulation", IEEE Symposium on Emerging Technologies, pp. 494–498, Pakistan, September 18, 2005

[22] G. Brambilla, A. Grazioli, M. Picone, F. Zanichelli, M. Amoretti, "A cost-effective approach to software-in-the-loop simulation of pervasive systems and applications", 2014 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), pp. 207–210, Hungary, March 24-28, 2014

[23] M. Muresan, D. Pitica, "Software in the loop environment reliability for testing embedded code", IEEE 18th International Symposium on Design and Technology in Electronic Packaging (SIITME), pp. 325–328, Romania, October 25-28, 2012

[24] National Instruments, "Automating Real-Time Testing for Embedded Software Development", Seminar lecture, presentation available at: ftp://ftp.ni.com/pub/branches/spain/eventos/seminarios/seminario_test/herramientas_para_automatizacion_validacion_software_embebido.pdf

[25] K. Pierce, C. Gamble, Y. Ni, J. F. Broenink, "Collaborative modelling and co-simulation with destecs: A pilot study", IEEE 21st International WETICE, pp. 280–285, Tunisia, June 25-27, 2012

[26] D. Meola, L. Iannelli, L. Glielmo, "Flight control system for small-size unmanned aerial vehicles: Design and software-in-the-loop validation", 21st Mediterranean Conference on Control and Automation, pp. 357–362, Greece, June 25-28, 2013

[27] I. Prado, D. Castro, S. M. G. Filizola, D. Santos, "Software in the loop simulation for multirotor position tracking using a linear constrained model predictive controller", Proceeding Series of the Brazilian Society of Applied and Computational Mathematics, pp. 0100491–0100496, 2013

[28] S. Werner, L. Masing, F. Lesniak, J. Becker, "Software-in-the-loop simulation of embedded control applications based on virtual platforms", 25th International Conference on Field Programmable Logic and Applications (FPL), pp. 1–8, United Kingdom, September 2-4, 2015

[29] M. Hellerer, M. J. Schuster, R. Lichtenheldt, "Software-in-the-loop simulation of a planetary rover", International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS 2016), China, June 19-22, 2016

[30] A. Hosseini, H. Karimi, P. Zarafshan, J. Massah, Y. Parandian, "Modeling and control of an octorotor flying robot using the software in a loop", 4th International Conference on Control, Instrumentation, and Automation (ICCIA), pp. 52–57, Iran, January 27-28, 2016

[31] A. Santana, L. Martins, R. Duarte, G. Arantes, "Processor-in-the-Loop Simulations Applied to the Design and Evaluation of a Satellite Attitude Control", in: Computational and Numerical Simulations, Intech, 2014

[32] G. Stoeppler, T. Menzel, S. Douglas, "Hardware-in-the-loop simulation of machine tools and manufacturing systems", Computing & Control Engineering Journal, Vol. 16, No. 1, pp. 10–15, 2005

[33] O. Gietelink, J. Ploeg, B. De Schutter, M. Verhaegen, "Development of advanced driver assistance systems with vehicle hardware-in-the-loop simulations", Vehicle System Dynamics, Vol. 44, No. 7, pp. 569–590, 2006

[34] L. Zouari, H. Abid, M. Abid, "Sliding mode and pi controllers for uncertain flexible joint manipulator", International Journal of Automation and Computing, Vol. 12, No. 2, pp.117–124, 2015

[35] L. Zouari, H. Abid, M. Abid, "Backstepping controller for electrically driven flexible joint manipulator under uncertainties" International Journal of Applied Engineering Research, Vol. 12, No. 8, pp. 19885–19896, 2015