Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2015) 23: 789 – 803
© TÜBİTAK
doi:10.3906/elk-1212-20

Conceptual design of a low-cost real-time hardware-in-the-loop simulator for satellite attitude control system

Farhad BAYAT*
Department of Engineering, Faculty of Electrical Engineering, University of Zanjan, Zanjan, Iran

Received: 04.12.2012 • **Accepted/Published Online:** 26.05.2013 • **Printed:** 30.04.2015

Abstract: Integration of flight hardware with real-time simulation increases satellite attitude control system (ACS) reliability by providing greater test coverage through end-to-end testing in a realistic test environment. In this paper, a compound hardware and software simulator has been designed for evaluation and testing of the spacecraft ACS, placing emphasis on the real-time hardware-in-loop (RTHIL) architecture. The environment comprises both real-time control and data acquisition applications on a network of ACS hardware, the MATLAB Real-Time Workshop, and a PCI device to join the hardware and software units. Moreover, a graphical 3D simulator has been designed, enabling the designers and researchers to intuitively analyze the quality of spacecraft maneuvers. Finally, an ad hoc attitude-stabilizing control law for magnetic actuated satellites has been proposed and implemented in the proposed environment and the efficiency, correctness, and robustness of this control law have been verified using RTHIL simulation results.

Key words: Satellite attitude control, satellite compound simulator, real-time hardware-in-the-loop simulation

1. Introduction

Clearly, the subsystems used in spacecraft, and especially attitude determination and control systems, are mission-critical real-time embedded systems, which have to be tested to ensure reliable operation in space [1,2]. Numerical and entirely software-based simulations are inadequate for obtaining mal-operations and verifying the high nonlinear complex systems such as spacecraft. Traditionally, spacecraft operation was verified and analyzed using the so-called engineering model [3], which integrates the dedicated electrical systems model and flight model to simulate the spacecraft operation. The importance of such functional modeling, simulation, and validation, especially in critical projects, has been obvious for researchers since the Ariane 5 launcher ended in failure in 1996. Despite the fact that in the Ariane 5 project a series of verification and test analyses were done, including numerous design reviews on both hardware and software units of subsystems, an error in the software design procedure remained undetected and eventually led to failure [4]. The importance of real-time hardware-in-loop (RTHIL) simulations has been reported in many different applications [5–10]. A pure software simulator for satellite attitude motion on the orbit was proposed in [11]. Application of hardware in the loop simulation for microsatellites was proposed in [12]. A satellite attitude simulator was designed in [13] to emulate the attitude motions by using a hydraulic manipulator whose end-effector rigidly grasps the satellite structures. In [14], an application of real-time simulation was reported for a 3-axis stabilized earth observation satellite, in which the functionality of attitude determination and the control system was tested using a real-time C

*Correspondence: bayat.farhad@znu.ac.ir

environment. Recently, a software-level simulator for satellite attitude and orbital simulation was proposed in [15].

To ensure a reliable and robust operation of a satellite's subsystems, e.g., the attitude control system (ACS), it is crucial to perform a series of RTHIL simulations. Although there are numerous commercial satellite simulator packages on the market [16–18], they are expensive and cannot be easily modified for particular missions. In this regard, the main objective of this paper is to design a low-cost, laboratory-scaled test bed that is easy to construct and sufficiently reliable for verification of the satellite ACS in an almost realistic environment on the ground rather than in-orbit, which eventually reduces the expenses and risks. In the proposed RTHIL architecture, the ACS subsystem and all interfaces are fully emulated using hardware subsystems while the environmental models (orbit, magnetic field, and gravity) are simulated in a real-time software environment. This paper provides a laboratory environment including the ACS hardware, satellite attitude motion software, and an interface unit that joins the hardware and software in RTHIL architecture.

2. Equations of satellite attitude motion

The dynamic equation of motion for a satellite in low earth orbit is [19,20]:

$$I\dot{\omega}_{cw} = -\omega_{cw} \times I\omega_{cw} + \mathbf{T}_{ctrl} + \mathbf{T}_{gg} + \mathbf{T}_{dist}, \tag{1}$$

where $I \in R^{3\times3}$ is the tensor of inertia and $\omega_{cw} \in R^3$ denotes the satellite angular rate with respect to the earth-centered inertial (ECI) coordinate system (CS), resolved in body frame. $\mathbf{T}_{ctrl}, \mathbf{T}_d \in R^3$ denote the control and external disturbance torques, respectively. $\mathbf{T}_{gg} = 3\omega_o^2 (k_o \times Ik_o) \in R^3$ is the gravity gradient torque (see [20] for details). ω_o is the orbital rate and k_o denotes the unit vector along the z-axis of orbital CS. The attitude kinematics can be described by means of quaternion parameters as [19]:

$$\dot{\mathbf{q}} = \frac{1}{2}\omega_{co}q_0 + \frac{1}{2}\omega_{co} \times \mathbf{q}, \qquad \dot{q}_0 = -\frac{1}{2}\omega_{co}.\mathbf{q}, \tag{2}$$

The following relation describes the satellite angular rates with respect to the ECI CS (ω_{cw}) and orbital CS (ω_{co}):

$$\omega_{co} = \omega_{cw} - \omega_o \begin{bmatrix} 2(q_1q_2 + q_0q_3) \\ q_0^2 - q_1^2 + q_2^2 - q_3^2 \\ 2(q_2q_3 - q_0q_1) \end{bmatrix}. \tag{3}$$

Here, it has been considered that the satellite is actuated by only a set of orthogonal magnetorquers. Therefore, $\mathbf{T}_{ctrl}$ acting on the satellite body is [21,22]:

$$\mathbf{T}_{ctrl} = \mathbf{M} \times \mathbf{B}, \tag{4}$$

where $\mathbf{M} \in R^3$ is the magnetic moment and $\mathbf{B} \in R^3$ denotes the local earth magnetic field. Eqs. (1)–(4) constitute the satellite model, where $\mathbf{M}$ is the control variable.

3. RTHIL functional architecture

3.1. Main architectures

Basically there are three different kinds of ACS simulators. The first is completely based on numerical simulations, which usually contains two (or more) interconnected digital computers working on a real-time

network framework. The real-time simulation of the environmental models, the ACS subsystems, and data handling are implemented and executed numerically, communicating through a network connection. The second is the most complex and expensive. Mainly, there are two different platforms for this type. The first platform is a 3-axis air-bearing with sensors, actuators, and control electronics mounted on it [23]. The second contains a 3-axis servo table platform for sensors with control electronics and actuators outside the platform, and the control loop is closed through a digital computer that simulates the satellite dynamics and in turn drives the servo table as shown in Figure 1.

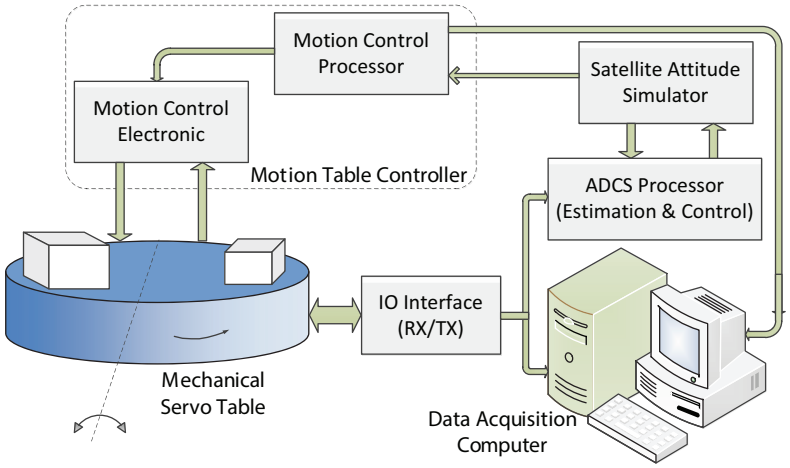

Figure 1. The 3-axis RTHIL servo table platform.

Note that, although the first type of simulator is the simplest, it cannot provide adequate realistic and end-to-end test requirements in some applications. On the other hand, the second type of simulator is the most realistic, but it is complex and expensive because of using mechanical platforms. This motivates us to choose a different kind of RTHIL simulator, the so-called mixed software and hardware (MS&H) simulator. This kind of simulator moderately enjoys the advantages of both previous types, i.e. it is sufficiently simple to construct and accurate for RTHIL verification in many applications. The block diagram and main subsystems of the MS&H simulator for the satellite ACS are shown in Figure 2.

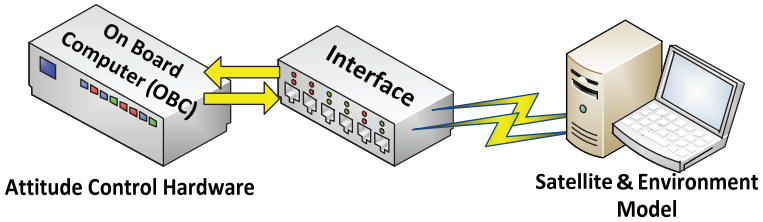

Figure 2. Block diagram and main subsystems of the MS&H simulator.

The main components of the MS&H simulator are as follows:

a. Digital computer for modeling the satellite's attitude motion and environmental effects (orbit, gravity, magnetic field, etc.).

b. Attitude determination and control hardware, for implementation of attitude control laws, command and data handling (C&DH) procedures, and synchronizing of the software and hardware units.

c. Peripheral component interconnect (PCI) interface, which interconnects the ACS (hardware) and digital computer (software) in a real-time closed loop structure.

Generally, the above three parts together provide a flexible environment for modeling, analyzing, and developing the flight hardware such as the satellite ACS.

3.2. RTHIL subsystems design

RTHIL simulations increase the reliability of the control system design, verification, and development procedures. The typical closed-loop control system model consists of the plant, sensor, and controller (or compensator). This structure was presented and discussed in [24]. The same building blocks are used in this paper to implement the ACS.

Real-time simulation and hardware in the loop testing have previously been used for spacecraft ACS design and also for subsystem verification and validation [25,26]. As mentioned, the main goal of this work is to provide a low-cost, laboratory-scale simulator that is easy to construct and sufficiently accurate for performance analysis and robustness verification of the ACS unit. To this aim, all dynamic equations including satellite attitude motion, environmental effects, actuators, and sensors have been modeled in the digital computer, while the ACS unit is fabricated enabling the designer to implement and test any appropriate attitude control law. This feature decreases the complexity of design and development procedures to a large extent, while keeping the accuracy of validation within an acceptable limit for several applications. Here we use the MATLAB Real-Time Workshop (RTW) toolset to implement the satellite equations of attitude motions, environmental effects, actuators, and sensors.

MATLAB RTW provides an integrated environment to specify, analyze, and simulate dynamic models using generated code in a real-time network with the capability of hardware detection. It is possible to simulate the entire system, the controller only (for rapid prototyping before the real system), or the plant only [27]. The hierarchical block diagram of the satellite's attitude motion equations and the environmental effects are shown in Figure 3. This model contains three main parts: satellite and environment model, input and output interfaces, and C&DH units including input decoder and output encoder. These blocks together represent the software unit of the MS&H real-time simulator.

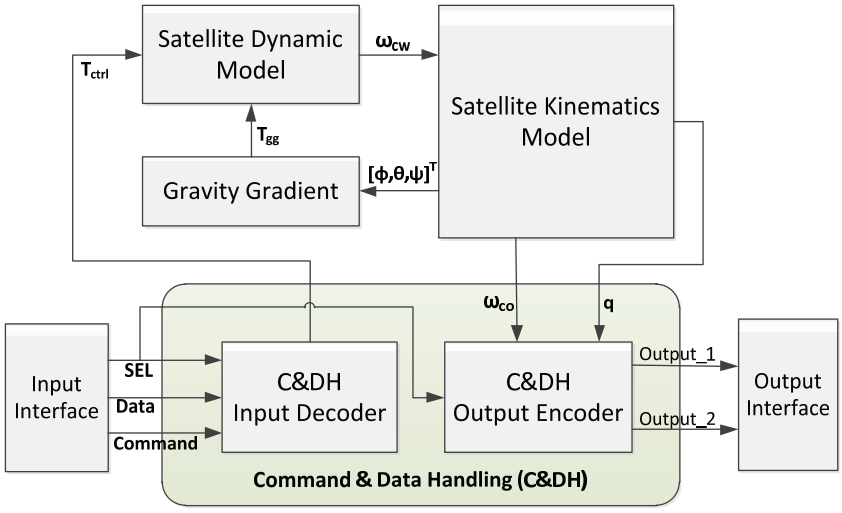

Figure 3. Hierarchical model for environment and satellite equations of attitude motion.

For real-time simulation purposes, the fixed-step Bogacki–Shampine method [28] is chosen as a solver, which computes the model's state at the next time step using an explicit Runge–Kutta (2, 3) formula for numerical integration (see [28] and [29] for more details). The sample time properties of all blocks in the RTW are set to 1 ms and the sampling times of the data acquisition and input/output components are set to 10 ms. The implemented control law in the ACS hardware should then be executed within 10 ms. Naturally, for verification of more sophisticated control laws, one needs to use much expensive high-frequency hardware platforms. The block diagram of the ACS hardware and the signal flow graph between its C&DH unit and the corresponding software unit are shown in Figure 4. The ACS hardware has an Atmel low-power CMOS 8-bit microcontroller (i.e. ATmega128) as its central processor, which is based on the AVR enhanced RISC architecture working with 16 MIPS throughput at 16 MHz [30]. Note that the hardware unit of C&DH is selected as the master part and the corresponding software unit as the slave. The whole closed-loop MS&H simulator is shown in Figure 5.

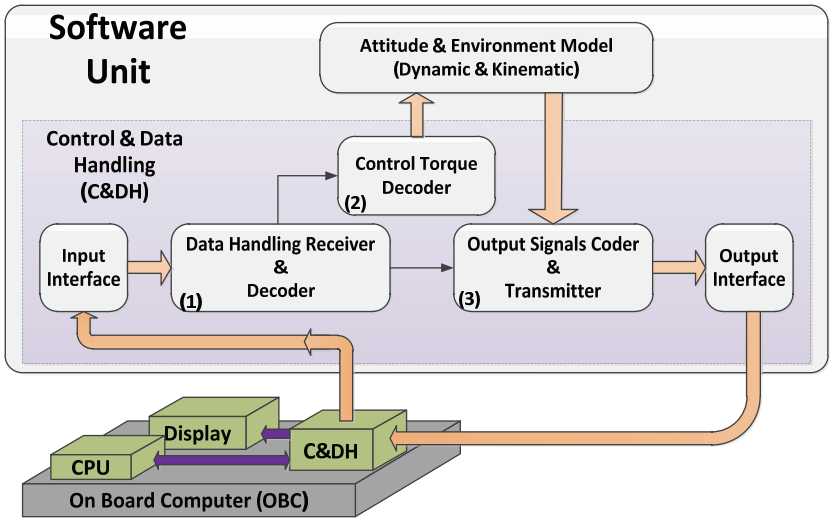

Figure 4. Block diagram of the ACS's C&DH and the corresponding software unit.

Here we use a commonly used PCI-1711 and a parallel port as devices interconnecting the hardware and software units. Considering the mission objectives and control system requirements of the ACS, at least 8 outputs (i.e. 4 quaternion parameters, 3 angular velocities, and 1 synchronization time) and 3 inputs (control torques) are needed to be able to close the control loop of the ACS. However, the selected interface (PCI-1711) provides only one 16 bit digital input and output. To cope with this limitation, the C&DH unit plays a central role. In this regard, the hardware part of C&DH provides command signals to manage data transmission between hardware and software units. When a command signal is received in the software unit, one of the following tasks will be executed associated with the command signal:

1. Sampling and encoding one of the required 8 parameters (as the input for ACS), and sending it to the ACS to generate an appropriate control command.
2. Fetching and decoding the control command (torque) generated in the ACS hardware.
3. Accomplishing one real-time simulation cycle by applying the control torque to the running real-time software model.

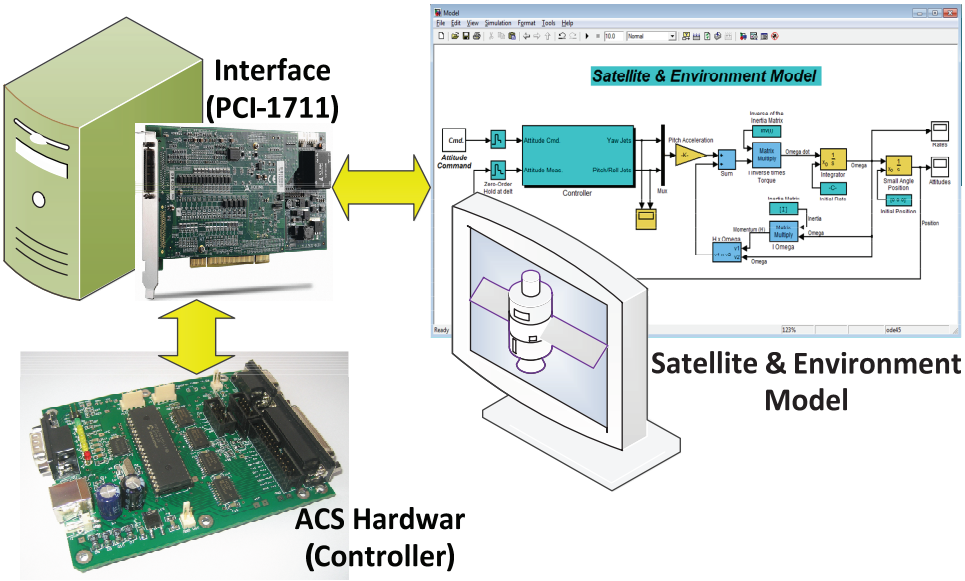

Figure 5. Satellite ACS RTHIL simulator.

Note that the 5 available bits of parallel port are allocated to manage the required command signals. Corresponding to these 5 bits, 32 different commands can be generated. All commands and the associated operations are shown in Table 1. The first 8 codes (SEL0–SEL7) denote sending satellite parameters to the ACS. For future development, 16 codes (SEL8–SEL15) have been reserved for receive/transmit procedures, 15 codes (SEL16–SEL30) stand for receiving control inputs from the ACS, and the last code (SEL31) is used as a flag for fulfillment of one real-time simulation cycle.

Table 1. Control signals and corresponding operations.

Operation	B4	B3	B2	B1	B0	SEL	Operation	B4	B3	B2	B1	B0	SEL
$q_0 \rightarrow ACS$	0	0	0	0	0	0	$t \rightarrow ACS$	0	0	1	1	1	7
$q_1 \rightarrow ACS$	0	0	0	0	1	1	$X \leftrightarrow ACS$ (reserved)	0	1	...	...	...	8-15
$q_2 \rightarrow ACS$	0	0	0	1	0	2	$T_x \leftarrow ACS$	1	0	0	0	0	16
$q_3 \rightarrow ACS$	0	0	0	1	1	3	$T_y \leftarrow ACS$	1	0	0	0	1	17
$\omega_x \rightarrow ACS$	0	0	1	0	0	4	$T_z \leftarrow ACS$	1	0	0	1	0	18
$\omega_y \rightarrow ACS$	0	0	1	0	1	5	$PC \leftarrow ACS$ (reserved)	1	...	...	...	...	19-30
$\omega_z \rightarrow ACS$	0	0	1	1	0	6	One cycle fulfillment	1	1	1	1	1	31

As mentioned, two standard ports are used to transfer attitude parameters and control signals. The attitude parameters are transferred via the PCI channel using the PCI-1711 interface. The control signals are transferred using a standard parallel port. Moreover, we use the standard serial communication port to interconnect the hardware unit and the designed graphical simulator (ZNU-SatSim) to evaluate ACS performance. Details and characteristics of the graphical simulator are described in Section 3.4.

3.3. Data transmission and quantization error

Since the attitude parameters and control torques are transferred through a 16 bit digital channel, the transferred data are not precise and quantization error will be an issue to be considered. To achieve a safe and proper operation, the quantization error must be limited to an admissible range. A practical way to find admissible bounds is to consider variations of all parameters in normal and worst cases. Accordingly, two constraints must be satisfied for each bound:

1. It must be feasible to describe the system in normal and worst case modes.
2. It must provide sufficient accuracy subject to the prescribed mission objective in normal mode.

It is clear that the above constrains are incompatible. The first one requires expansion of the admissible bounds, while the second requires their reduction. This issue has been considered and analyzed incorporating the numerical simulation results under the enforced quantization error. Finally, taking about 30% safety margin into account, the admissible bound corresponding to the quantization error for each parameter is determined as shown in Table 2. Decoding and encoding procedures for each variable to handle the given bounds are presented in Table 3.

Table 2. Admissible range and corresponding quantization errors of signals.

	Admissible bounds	Quantization errors
1	$T \in \left(\begin{array}{cc} -10^{-2} & +10^{-2} \end{array} \right) N.m$	$R_T = \frac{2\times 10^{-2}}{2^{16}} = 3 \times 10^{7}$
2	$q_i \in \left(\begin{array}{cc} -1 & +1 \end{array} \right)$	$R_q = \frac{2\times 1}{2^{16}} = 3 \times 10^{-5}$
3	$\omega_i \in \left(\begin{array}{cc} -0.1 & +0.1 \end{array} \right) {}^{rad}/_{s}$	$R_\omega = \frac{2\times 0.1}{2^{16}} = 3 \times 10^{-6}$
4	$t \in \left(\begin{array}{cc} 0 & 6500 \approx 1\, orbit \end{array} \right) s$	$R_t = \frac{6500}{2^{16}} \approx 0.1$

Table 3. Signal encoding and decoding formulas.

	Encoding formulas	Decoding formulas
1	$T_{Encode} = floor\left(\frac{T_k + T_{\max}}{2T_{\max}} \times 65535\right)$	$T_{Decode} = \left(\frac{2T_{Encode}}{65535} - 1\right) T_{\max}$
2	$q_{Encode} = floor\left(\frac{q_k + 1}{2} \times 65535\right)$	$q_{Decode} = \left(\frac{2q_{Encode}}{65535} - 1\right)$
3	$\omega_{Encode} = floor\left(\frac{\omega_k + \omega_{\max}}{2\omega_{\max}} \times 65535\right)$	$\omega_{Decode} = \left(\frac{2\omega_{Encode}}{65535} - 1\right) \omega_{\max}$
4	$t_{Encode} = floor\left(\frac{t}{t_{\max}} \times 65535\right)$	$t_{Decode} = \left(\frac{t_{Encode}}{65535}\right) t_{\max}$

3.4. Satellite graphical simulator (ZNU-SatSim)

The proposed low-cost MS&H simulator provides a suitable test bed for end-to-end verification and qualification of satellite attitude control laws. To achieve an intuitive feeling of the ACS performance and attitude maneuvers quality, a 3D graphical simulator has been designed, as shown in Figure 6. Using this graphical simulator one can observe and roughly predict possible mechanical shocks during the maneuvers that may strike the satellite's body. This enables the designer to obtain and prevent critical design conditions. The main window of the 3D graphical simulator (ZNU-SatSim) and its facilities are shown in Figure 6.

4. RTHIL Experimental Results

The performance of the closed-loop system and applied control law is investigated with respect to pointing error, robustness, and capability of simple implementation using RTHIL simulations. In our previous work [31] we

proposed a heuristic globally asymptotically stable attitude controller for a magnetic actuated satellite. In this section, we will implement this controller in order to verify its performance and robustness using the proposed RTHIL simulator.

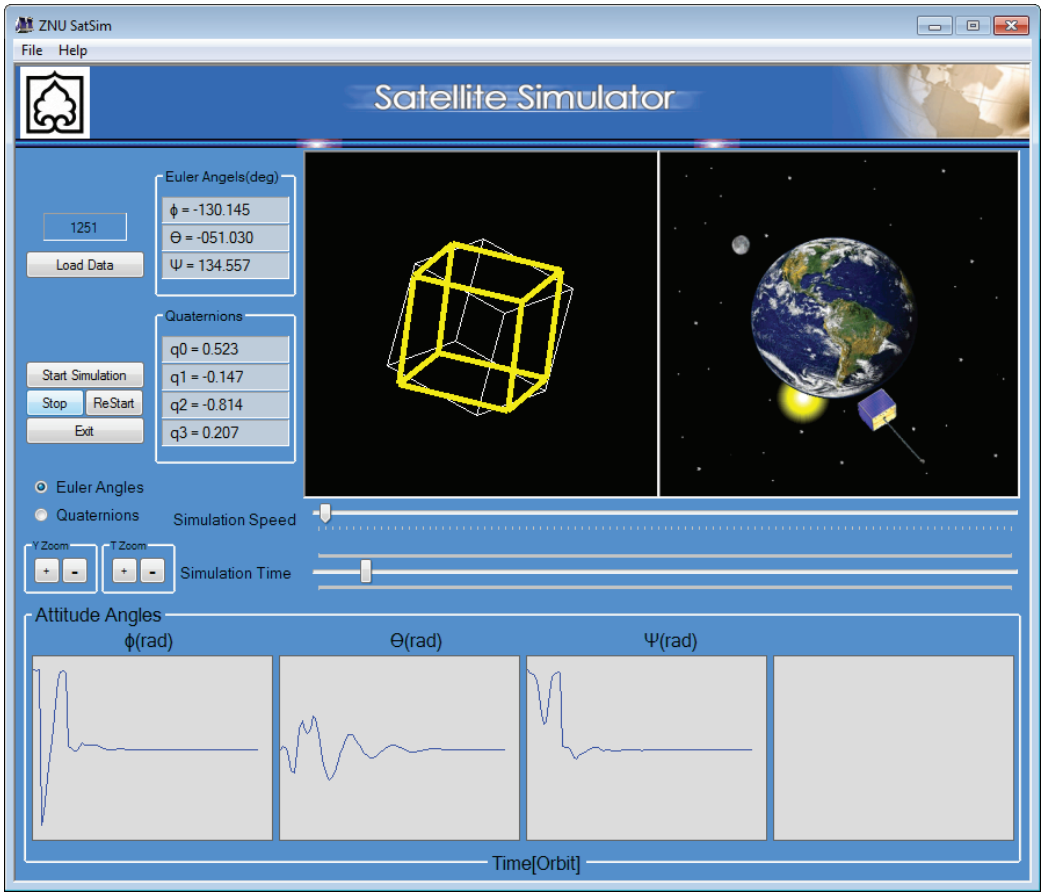

Figure 6. Graphical simulator (ZNU-SatSim).

4.1. Magnetic attitude controller

We assume that the only available actuator for attitude control purpose is a magnetorquer. Attitude control of magnetic actuated satellites was addressed in [22] and [31–34]. Using magnetorquer coils for control, a torque is generated by the interaction of the geomagnetic field with the magnetorquer current giving rise to a magnetic moment:

$$m\left(t\right) = n_{coil} i_{coil}\left(t\right) A_{coil}. \tag{5}$$

The control torque acting on the satellite, provided that the coils are placed orthogonal to the x-, y-, and z-axes of the body CS, is:

$$\mathbf{T}_{ctrl} = \mathbf{M} \times \mathbf{B} = \begin{bmatrix} M_y B_z - M_z B_y \\ M_z B_x - M_x B_z \\ M_x B_y - M_y B_x \end{bmatrix} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}. \tag{6}$$

The design goal is to find $\mathbf{M} = \begin{bmatrix} M_x & M_y & M_z \end{bmatrix}^T$ such that the attitude stabilization from each arbitrary initial condition is guaranteed. In the following, without loss of generality, the design procedure of our proposed

controller in [31] is described briefly for roll channel. For pitch and yaw channels, similar results, mutatis mutandis, can be adapted.

Considering Eq. (6), one obtains the magnetic dipole moment in roll axis (M_x) affecting the pitch and yaw axis control torques (T_y, T_z). Accordingly, one can see that $T_y \propto -M_x B_z$. On the other hand, the positive torque ($T_y > 0$) leads to positive angular velocity ($\omega_{boy} > 0$) and vice versa since the coordinate system is right orthogonal. Therefore, to eliminate this component of angular velocity, a reasonable guess is to choose the control torque proportional to the magnitude of angular velocity and versus it as $[T_y \propto -M_x B_z] \propto -\omega_{boy}$. This condition is selected as the first condition for attitude stabilization. It is easy to verify that this condition is guaranteed if one chooses $M_x \propto B_z \omega_{boy}$. Similarly, from Eq. (6) one gets $T_z \propto M_x B_y$. Thus, the yaw axis control torque (T_z) that can eliminate yaw angular velocity (ω_{boz}) should be $[T_z \propto M_x B_y] \propto -\omega_{boz}$. Similarly, one can easily verify that $M_x \propto -B_y \omega_{boz}$ is the sufficient condition for satisfaction of the last relation. To avoid rehash, similar reasoning can be applied for pitch and yaw channels. Table 4 presents sufficient conditions for three channels.

4.2. Controller design

The block diagram of the closed-loop magnetic actuated satellite is shown in Figure 7. Note that this architecture describes the attitude detumbling architecture, in which the angular velocity feedback is only required. The magnetic controller contains three multiple-input single-output (MISO) subcontrollers associated with three channels, roll, pitch, and yaw. The input vector of channel k corresponding to the control signal M_k, $(k = x, y, z)$ is:

$$\mathbf{U}_k = \begin{bmatrix} \bar{\omega}_i & \bar{\omega}_j & B_i & B_j \end{bmatrix}^T, \quad (\bar{\omega} \equiv \omega_{bo}) \\ (i \neq k, j \neq k, i \neq j), (i, j, k) \in \{x, y, z\} \tag{7}$$

where $\bar{\omega} \equiv \omega_{bo}$ and B_i denotes the ith component of the geomagnetic field. As summarized in Table 4, for attitude stabilization of each channel, two conditions are required to be satisfied simultaneously. A weighted sum of the two conditions is selected for each channel as:

$$M_k = \xi_{k1} (B_j \omega_{boi}) + \xi_{k2} (-B_i \omega_{boj}), \quad (i, j, k) \in \{x, y, z\}. \tag{8}$$

The following theorem characterizes the proposed control law.

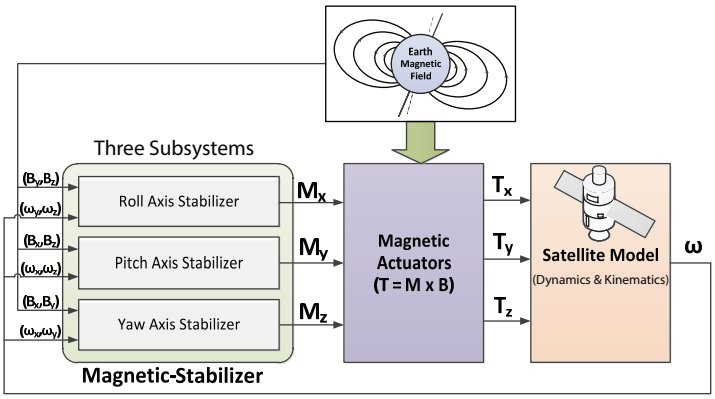

Figure 7. Closed-loop architecture of the satellite attitude detumbling controller.

Theorem 1 *Global asymptotic stability*

Consider the satellite equations of attitude motion, Eqs. (1) and (2), in a nonequatorial orbit. Let $\xi_k = \xi_{k1} = \xi_{k2} > 0$*, and then the control law given in Eq. (8) makes the equilibrium point* $\omega_{bo} = \mathbf{0}$ *globally asymptotically stable.*

Proof See [31] for a detailed proof. □

The effectiveness and performance of the proposed heuristic magnetic controller can be verified using the proposed real-time hardware-in-the-loop simulator. To this aim, assuming different initial conditions given in Table 5, two simulations were performed representing the normal- and worst-case situations. The satellite is symmetric with inertia matrix given by $I_{xx} = 198, I_{yy} = 200, I_{zz} = 18, I_{xy} = 0.3, I_{xz} = 0.5$, and $I_{yz} = 0.4$ $(kg.m^2)$, which operates in a near polar orbit (92° inclination) with an altitude of 800 km and a corresponding orbit period of about 6500 s. In the software part of the RTHIL simulator the earth's magnetic field is modeled based on the well-known International Geomagnetic Reference Field (IGRF) model [35]. Figures 8–15 illustrate the normal- and worst-case simulations for the proposed heuristic controller implemented in the RTHIL simulator.

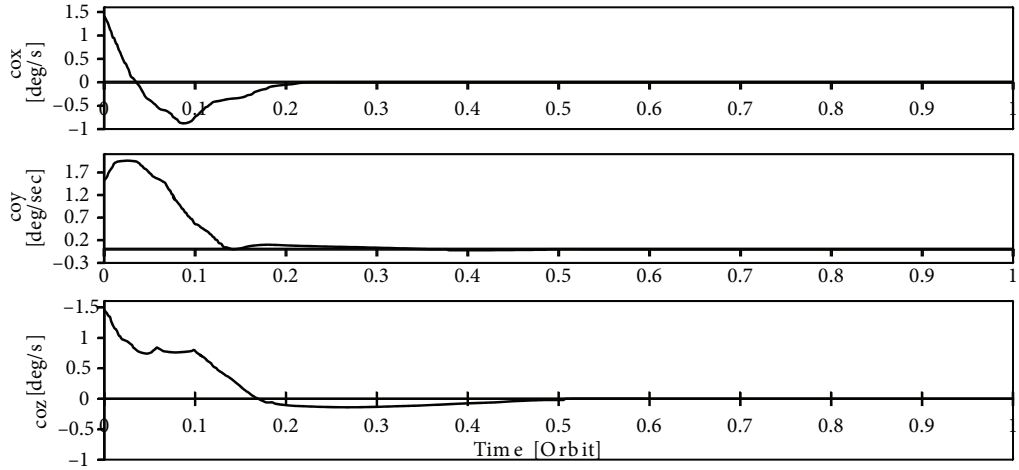

Figure 8. RTHIL simulation results of ω_{co} (normal case).

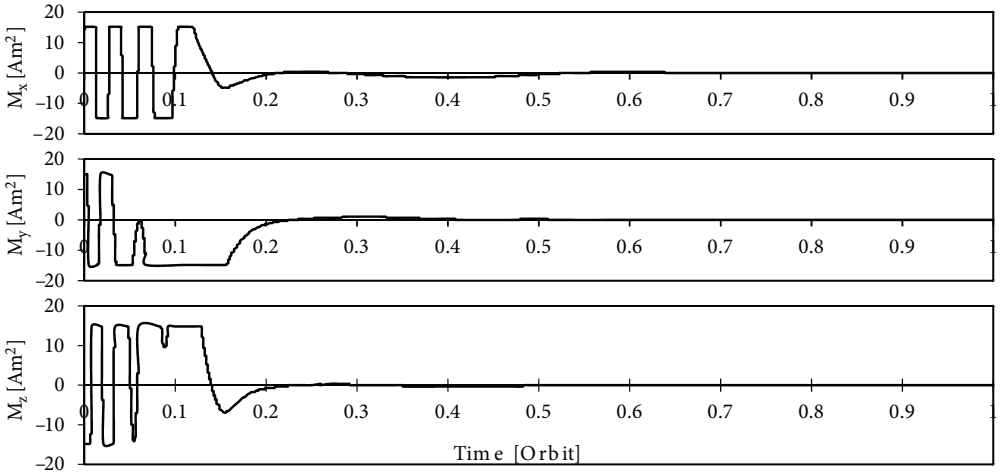

Figure 9. RTHIL simulation results of M (normal case). (saturated at ± 15 Am^2).

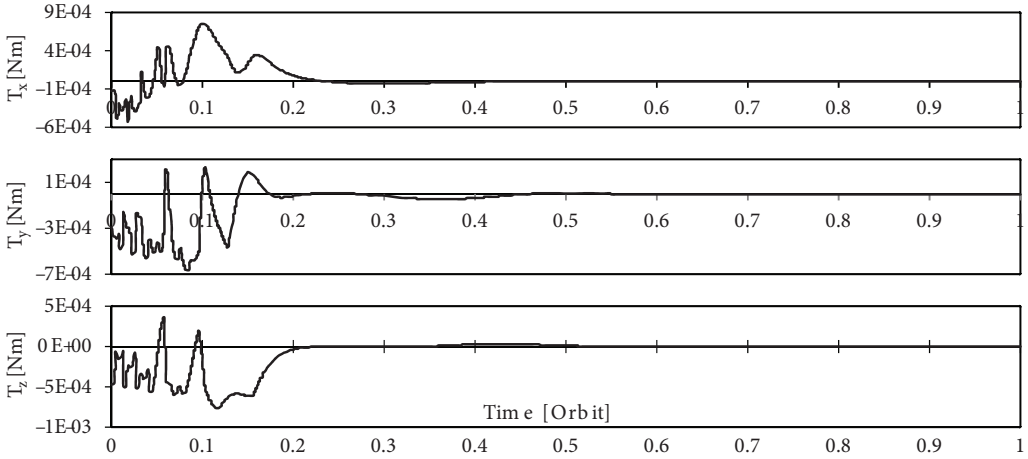

Figure 10. RTHIL simulation results of T (normal case).

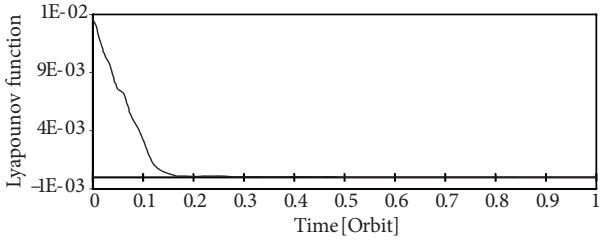

Figure 11. Total energy of the satellite (normal case).

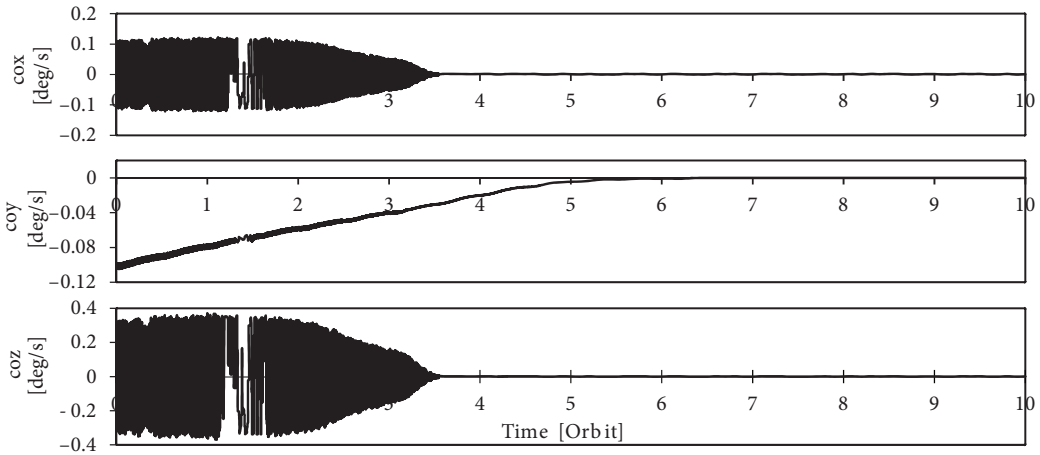

Figure 12. RTHIL simulation results of ω_{co} (worst case).

Note that all RTHIL simulations were performed in the presence of constraints such as actuator saturation and quantization and round-off errors. Based on the experimental results, the proposed controller exhibits a satisfactorily robust and reliable performance against the saturation, quantization, and round-off errors. However, it is important to note that a price has to be paid regarding the closed-loop performance of the controller to overcome aforementioned limitations.

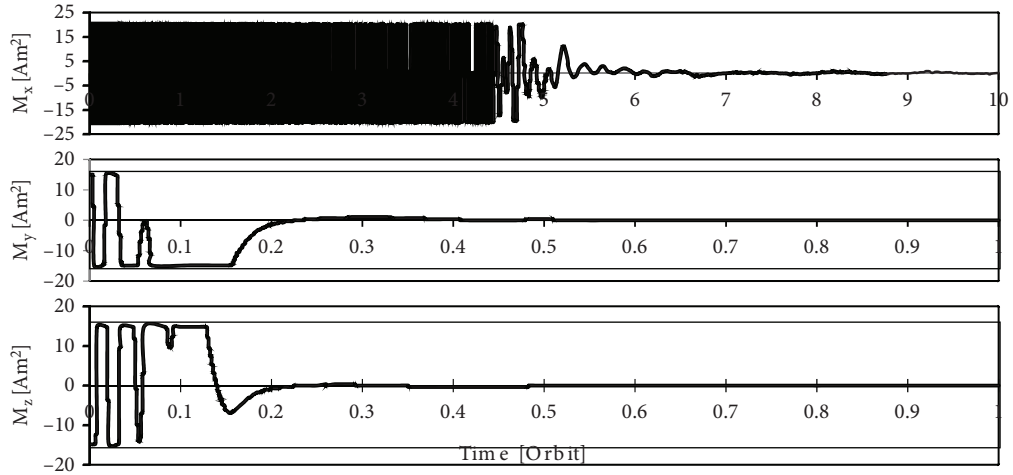

Figure 13. RTHIL simulation results of M (worst case) (saturated at ± 20 Am^2).

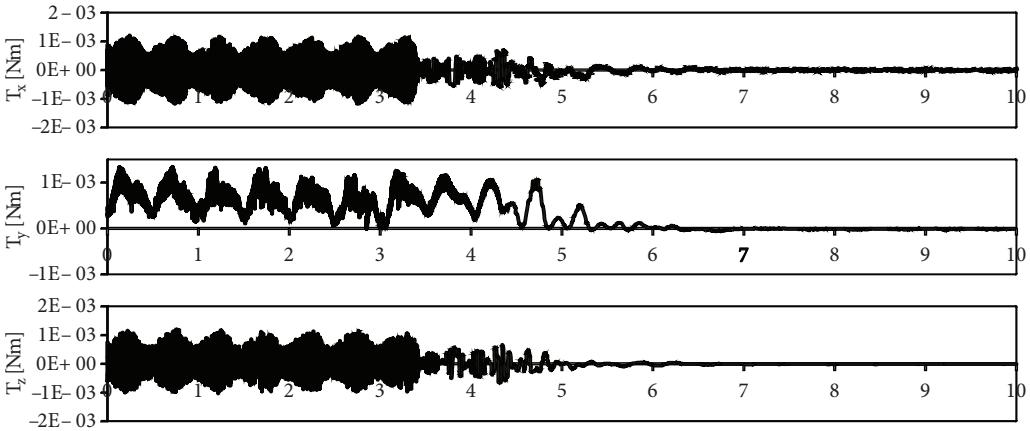

Figure 14. RTHIL simulation results of T (worst case).

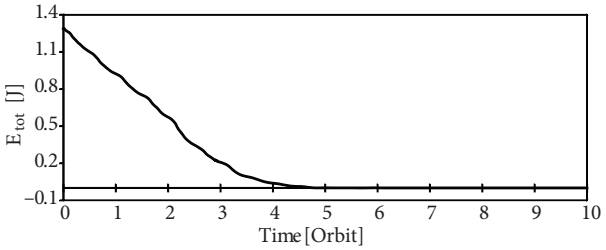

Figure 15. Total energy of the satellite (worst case).

Table 4. Sufficient stabilizing conditions for roll, pitch, and yaw channels.

Roll channel	Pitch channel	Yaw channel
$M_x \propto B_z\omega_{coy}$	$M_z \propto B_y\omega_{cox}$	$M_y \propto B_x\omega_{coz}$
$M_x \propto -B_y\omega_{coz}$	$M_z \propto -B_x\omega_{coy}$	$M_y \propto -B_z\omega_{cox}$

As an example, in the normal-case simulation the attitude stabilizing lasts for about 0.5 orbits, while in the worst-case simulation the attitude is captured after more than 5 orbits (10 times more). This phenomenon

apparently originated from the fact that the magnetic coils are saturated in the worst-case simulation as depicted in Figure 12 for a long time. However, the important point is that the closed-loop asymptotic stability of both cases (i.e. normal and worst cases) is preserved. As shown in Figures 11 and 15, this point is also verified in terms of the system's total energies, which are uniformly decreasing for both cases.

Table 5. Initial conditions of normal- and worst-case for RTHIL simulations.

	Normal case	Worst case
$\omega_{co}(t=0)\left(\frac{\text{deg}}{s}\right)$	$\begin{bmatrix} 1.5 & 1.5 & 1.5 \end{bmatrix}^T$	$\begin{bmatrix} 5.7 & -5.7 & 5.7 \end{bmatrix}^T$

Finally, we emphasize that the proposed RTHIL architecture is not restricted to the attitude detumbling verification. As discussed in Section 3, the proposed RTHIL simulator provides full feedback from both attitude angle and angular velocity. Accordingly, this architecture can be used for other purposes and other control schemes (e.g., attitude maneuver) taking the constraints given in Tables 1–3 into account.

5. Conclusion

The main objective of this paper was designing a low-cost, realistic, and easily constructed test environment for analysis and design of an ACS. Toward this aim, a mixed hardware and software simulator were designed. Using this simulator, real-time end-to-end simulations can be done. As an experimental result, a heuristic globally asymptotically stabilizing control law recently developed by the authors was implemented and analyzed in the proposed RTHIL test-bed. The RTHIL simulation results confirmed the admissible operation of the proposed control law. Finally, the designed 3D graphical simulator (ZNU-SatSim) provides a visual platform for intuitive qualification of the attitude maneuvers.

References

[1] X. Yue, "Robust adaptive control of a 3-Axis motion simulator for instruments testing", IEEE Power Electronic Specialist Conference, Cairns, Australia, pp. 552–557, 2002.

[2] C. Frangos, "Control system analysis of a hardware in-the-loop simulation", IEEE Transactions on Aerospace and Electronic Systems, Vol. 26, pp. 666–669, 1990.

[3] R. Hendricks, J. Eickhoff, "The significant role of simulation in satellite development and verification", International Aerospace Science and Technology Congress, Vol. 9, pp. 273–283, 2005.

[4] J.L. Lions, Ariane 5 Flight 501 Failure: Report of the Inquiry Board, Paris, ESA/CNES, 1997.

[5] M. Montazeri, M. Nasiri, M. Rajabi, M. Jamshidfard, "Actuator based hardware-in-the-loop testing of a jet engine fuel control unit in flight conditions", Simulation Modelling Practice and Theory, Vol. 21, pp. 65–77, 2012.

[6] T. Ersal, M. Brudnak, A. Salvi, J.L. Stein, Z. Filipi, H.K. Fathy, "Development and model-based transparency analysis of an Internet-distributed hardware-in-the-loop simulation platform", Mechatronics, Vol. 21, pp. 22–29, 2011.

[7] N.R. Gans, W.E. Dixon, R. Lind, A. Kurdila, "A hardware in the loop simulation platform for vision-based control of unmanned air vehicles", Mechatronics, Vol. 19, pp. 1043–1056, 2009.

[8] H. Temeltas, S. Bogosyan , M. Gokasan, "Hardware-in-the-loop robot simulators for on-site and remote education in robotics", International Journal of Engineering Education, Vol. 22, pp. 815–821, 2006.

[9] D. Bullock, B. Johnson, R.B. Wells, M. Kyte, Z. Li, "Hardware-in-the-loop simulation", Transportation Research Part C: Emerging Technologies, Vol. 12, pp. 73–89, 2004.

[10] D.M. Vilathgamuwa, X. Yue, K.J. Tseng, "Development and control of a 3-axis motion simulator for satellite ADCS hardware-in-the-loop simulation", 30th Annual Conference of the IEEE Industrial Electronics Society, Vol. 1, pp. 524–529, 2004.

[11] M. Nasirian, H. Bolandi, A. Khaki Sedigh, A.R. Khoogar, "Design of a satellite attitude control simulator", 1st International Symposium on Systems and Control in Aerospace and Astronautics, Vol. 1, pp. 163–166, 2006.

[12] W. Qian, Z. Tao, S. Jingyan, "Study on the control system of a hardware-in-loop micro-satellite simulator", IEEE International Conference on Robotics and Biomimetics, pp. 2427–2432, 2009.

[13] F. Aghili, M. Namvar, G. Vukovich, "Satellite simulator with a hydraulic manipulator", IEEE International Conference on Robotics and Automation, pp. 3886–3892, 2006.

[14] C. Yavuzyilmaz, M. Akbas, Y. Acar, F. Gulmammadov, O. Kahraman, Y. Subasi, N. Ertongur, C.S. Tufekci, "RASAT ADCS flight software testing with dynamic attitude simulator environment", 5th International Conference on Recent Advances in Space Technologies, pp. 974–977, 2011.

[15] Z. Aiwu, Z. Jianping, M. Yongping, Z. Kai-Min, "Orbit Dynamics Simulation of Satellite Simulators", 5th International Conference on Measuring Technology and Mechatronics Automation, pp. 777–780, 2013.

[16] F. Bauer, J. Bristow, D. Folta, K. Hartman, D. Quinn, J. How, "Satellite formation flying using an innovative autonomous control system (AUTOCON) environment", Proceedings of the AIAA Guidance, Navigation, and Control Conference, 1997.

[17] A.I. Solutions, FreeFlyer User's Guide, v.4.0, Lanham, MD, USA, A.I. Solutions, 1999.

[18] J. Woodburn, K. Williams, H.D. Witt, "The customization of satellite tool kit for use on the NEAR mission", Proceedings of the Space Flight Mechanics Conference, 1997.

[19] J.R. Wertz, Spacecraft Attitude Determination and Control, Dordrecht, the Netherlands, Kluwer, 1990.

[20] M.J. Sidi, Spacecraft Dynamics and Control, Cambridge, UK, Cambridge University Press, 1997.

[21] S. Das, M. Sinha, K.D. Kumar, A. Misra, "Reconfigurable magnetic attitude control of Earth-pointing satellites", Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, Vol. 224, pp. 1309–1326, 2010.

[22] R. Wisniewski, M. Blanke, "Fully magnetic attitude control for spacecraft subject to gravity gradient", Automatica, Vol. 35, pp. 1201–1214, 1999.

[23] S. Ramaswamy, "The role of hardware in-loop motion simulation for Indian satellites", IEEE Transactions on Aerospace and Electronic Systems, Vol. 27, pp. 261–267, 1991.

[24] C.L. Phillips, H.T. Nagle, Digital Control System Analysis and Design, Upper Saddle River, NJ, USA, Prentice Hall, 1984.

[25] L.I. Slafer, "The use of real-time, hardware-in-the-loop simulation in the design and development of the new Hughes HS601 spacecraft attitude control system", Proceedings of the 3rd Annual Conference on Aerospace Computational Control, Vol. 2, p. 713, 1989.

[26] D. Zimbelman, "The attitude control system test bed for SWAS and future SMEX missions", Guidance and Control Conference, Vol. 92, pp. 145–163, 1996.

[27] MathWorks-MATLAB, Dutch Space Streamlines Development of Advanced Satellite System Simulators, Natick, MA, USA, MathWorks, 2015.

[28] P. Bogacki, L.F. Shampine, "A 3(2) pair of Runge–Kutta formulas", Applied Mathematics Letters, Vol. 2, pp. 321–325, 1989.

[29] L.F. Shampine, M.W. Reichelt, "The MATLAB ODE Suite", SIAM Journal on Scientific Computing, Vol. 18, pp. 1–22, 1997.

[30] ATMEL Corporation System, http://www.futurlec.com/Atmel/ATMEGA128.shtml.

[31] F. Bayat, H. Bolandi, A.A. Jalali, "A heuristic design method for attitude stabilization of magnetic actuated satellites", Acta Astronautica, Vol. 65, pp. 1813–1825, 2009.

[32] M. Lovera, E.D. Marchi, S. Bittanti, "Periodic attitude control techniques for small satellites with magnetic actuators", IEEE Transactions on Control System Technology, Vol. 10, pp. 90–95, 2002.

[33] E. Silani, M. Lovera, "Magnetic spacecraft attitude control: a survey and some new results", Control Engineering Practice, Vol. 13, pp. 357–371, 2005.

[34] M. Lovera, A. Astolfi, "Spacecraft attitude control using magnetic actuators", Automatica, Vol. 40, pp. 1405–1414, 2004.

[35] C. Roithmayr, Contributions of Spherical Harmonics to Magnetic and Gravitational Fields, Technical Report, Houston, TX, USA, NASA Johnston Space Center, 1992.