

# Real-Time Digital Simulator Design for Differential Drive Mobile Robot using FPGA

Mehmet Rıza Saraç, Omur Aydogmus

**Abstract**—This paper presents a real-time simulation of a differential drive mobile robot (DDMR). The permanent magnet DC motors that drive the robot's left and right wheels were modeled and executed in real-time on a Field Programmable Gate Array (FPGA) based co-simulator platform, interfacing with the Webots robot simulator, which simulates the DDMR on the PC side. The electrical parameters, which are not available in robot simulators, were simulated and measured by the proposed co-simulator system in real-time under various environmental conditions and trajectories of the robot. Parameters such as current, voltage, and torque were measured instantaneously, enabling a more realistic simulation. Additionally, the cycle time of the robot simulator was determined to be 32 ms, and the developed FPGA-based simulation operated at approximately 2000 times the speed of the robot simulator. The results demonstrate the applicability of the developed platform in robotic applications.

**Index Terms**—Co-simulator, Digital Twin, FPGA, Mobile robot, Real-time digital simulator.

## I. INTRODUCTION

OVER time mobile robots have increasingly become integrated into our daily lives, taking on different forms and playing significant roles in various sectors [1]. They have proven their adaptability in industrial production [2], medical applications [3], agricultural processes [4], space exploration [5], and various other fields. As technology advances, it is evident that mobile robots have the potential to overcome challenges and stimulate innovation in different areas [6].

Mobile robots are typically divided into two main categories: wheeled robots [7] and legged robots [8]. Additionally, there are various approaches to combining wheeled and legged components to create different types of robots [9]. Wheeled mobile robots exhibit variations in wheel type, quantity, and assembly configuration. Designs range from standard types to mecanum, omni, and spherical wheels, as well as differential and steering wheels, depending on the connection configuration [10].

However, constraints such as time and cost make it impractical to work directly with mobile robots. Many applications require preliminary work to be done in a simulation environment. Various robot simulation platforms, such as Gazebo, Webots, CoppeliaSim, and Isaac Sim, have been developed for

this purpose. These simulators are considered a practical and effective approach for developing control strategies, but not all simulators are suitable for simulating electrical variables like current and voltage. These simulators typically focus on kinematic and dynamic modeling using physics engines [11].

As an alternative, simulation techniques, especially Hardware-in-the-Loop (HIL) simulations, are becoming increasingly popular in various systems, including robotics [12], automotive [13], power systems [14], and aircraft systems [15]. For instance, a backstepping control regulator has been implemented for PUMA 560 using the FPGA-in-the-loop option on the Xilinx Zedboard Zynq FPGA with the Hardware Description Language (HDL) Coder tool [16]. Another study focused on optimizing solution times for real-time use of inverse kinematics in articulated robots, developing and testing an optimization algorithm compatible with FPGA architecture [17]. Furthermore, a discussion on the application of FPGA-based HIL simulation to computer vision in an unmanned aerial vehicle is presented in [18].

A review of the existing literature shows that there is a significant gap in robot simulators that do not include electrical models. This paper proposes a novel solution using an FPGA architecture to simulate models of electric motors with high-speed processing. In this study, FPGA-based models for the drive motors of a DDMR were designed and executed in real-time with an FPGA-based co-simulator. In the proposed system, the angular velocities and currents of the motors were transmitted to a PC-based robot simulator platform, while voltage and torque information was received as feedback to the FPGA environment. As a result, the motors were operated in the FPGA environment and seamlessly integrated with the PC-operated robot. This innovative approach successfully addresses the gap in traditional robot simulators and improves the existing system by incorporating the previously ignored electrical model component.

This paper is organized as follows: It begins with an introduction that provides information about mobile robots and FPGA-based HIL systems. The next section presents the hardware side of the study, which includes FPGA. In the third section, the robot simulator and control concept are discussed. The experimental results are analyzed in the fourth section. Finally, the paper concludes with a summary section that recaps the main findings and explores their implications for future research.

## II. FPGA HARDWARE DEVELOPMENT

Real-time simulators use fixed time step algorithms for calculations. This approach requires Dommel's method to be

 **Mehmet Rıza Saraç** is with the Department of Mechatronics Engineering, Technology Faculty, Firat University, Elazig, TURKEY e-mail: mrsarac@firat.edu.tr

 **Omur Aydogmus** is with the Department of Mechatronics Engineering, Technology Faculty, Firat University, Elazig, TURKEY e-mail: oaydogmus@firat.edu.tr

Manuscript received Jan 04, 2024; accepted Feb 7, 2024.  
DOI: 10.17694/bajece.1414730

TABLE I  
INTEL CYCLONE V SE 5CSEBA6U23I7NDK FPGA RESOURCES

Resource	Description	Value
LEs (K)	Logic Elements	110
ALMs	Adaptive Logic Modules	41,910
Reg	Registers	116,036
MLAB (Kb)	Memory	621
M10K (Kb)	Memory	5,570
DSP	DSP Blocks	112
18x18 Mult.	18x18 Multipliers	224
PLL	Phase Locked Loop (Fractional)	6
DLL	Delay Locked Loop	4
Global Clock	Clock Networks	16
GPIO	General Purpose I/O	288

used for systems including nonlinear components like inductors and capacitors [19]. Simulator input signal is interpreted as a sequential arrays step functions. Every simulation step requires previous calculation results as boundary conditions. This requires previous simulation step to be completed and the results saved before the evaluation of next step. To compete with required calculation work load, high speed and high performance equipments need to be used. This performance can be achieved by software using high speed  $\mu P$  with combination of real-time operation systems or by hardware approach. Hardware development is mainly used for Very large-scale integration (VLSI) electronic components. However, VLSI manufacturing is a hard and painful process permitting very limited changes after tape-out process.

FPGA-based systems has great advantages for hardware development and permitting hardware changes even after end product is sold. In this study DE10-Nano development board was selected for its popularity. DE10-Nano development board is a Solution on Chip (SoC) development platform produced by Terasic company. Platform designed around Intel Cyclone V 5CSEBA6U23I7 SoC device which contains dual core ARM Cortex-A9 hardware processor and FPGA fabric with resources listed in the Table I [20].

The development board has an onboard Analog-Digital Converter (ADC) system based on Linear Technology LTC2308. The ADC has a resolution of 12 bits, operates at a speed of 500 KSPS, and 8 analog input channels. It supports a 4-wire Serial Peripheral Interface (SPI) connection interface with a maximum operating frequency of 40 MHz. Proposed FPGA-based real-time digital simulator system also includes an interface board for analog signal inputs and outputs. The analog interface board also contains Digital-Analog Converter (DAC) systems which the development board does not include. The DDMR simulated for this study has two DYNAMIXEL XL430-W250-T servo motors with gear ratio of 258.5:1. The proposed real-time simulator was designed to simulate Permanent Magnet Direct-Current (PMDC) motors. To simulate gear box mechanical loading of servo motors used by robot, additional mechanical load was configured with parameter inputs of real-time simulator. The load torque on the shaft of servo motors are simulated by Webots simulator and received as  $T_i$  signal by real-time simulator.

The real-time simulator was designed to accept eight input signals which four of them are supply voltage and torque

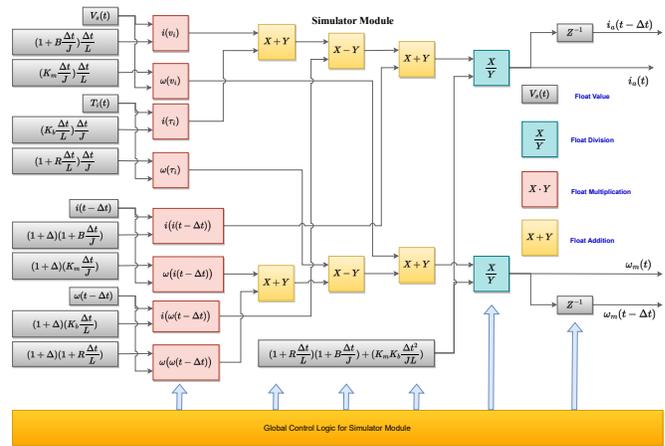


Fig. 1. PMDC motor simulator module of digital twin system

inputs for both left and right motors. Other input signals are parameter inputs for simulator assumed to be same for both motors which are  $B_L$ ,  $J_L$ ,  $K_M$ , and  $K_B$ . ADC inputs for supply voltages and torque inputs were privileged and configured to be read at every ADC read sequence.

The real-time simulator has eight channels to be read and with the read speed of 500 Kilo samples per Second (KSPS) each channel will have a 62.5 KSPS of speed. To increase sampling ratio a privilege system was used. The system read sequence has four privileged inputs and the only one of the four parameter input channels in ADC read sequence. This asymmetric read sequence enables a speed rate of 100 KSPS for privileged channels and only 25 KSPS speed for parameter inputs. All required ADC channels were read in 10  $\mu s$  and after this read sequence was completed and a signal asserted within simulator as simulation step signal.

FPGA hardware development was made by using Intel Quartus Prime Lite edition software freely available from manufacturers web page. For hardware development Verilog hardware description language (HDL) is used and Intel floating point IP library was used for floating calculations. The development board generate a clock signal of 50 MHz and fed FPGA at three different pins. A hardware Phase Locked Loop (PLL) unit was used to generate required clock signals required by proposed real-time simulator. To configure and read from ADC system an SPI module was developed to communicate with LTC2308 ic. This module requires 40 MHz clock signal to operate which was generated by PLL module. A module was also designed to control reset signal. The analog interface board contain DAC channels which use parallel interface to communicate with development board. A module was developed with in FPGA for PI interface used by DAC channels.

The real-time simulator contains three main modules which operate at 50 MHz required for simulation. There was two motor simulator modules as depicted in Fig. 1 operating in parallel to simulate right and left motors of robot. The real-time simulator also contains a module as depicted in Fig. 2 as parameter module which is used to calculate required coefficients for motor simulator modules from parameter input

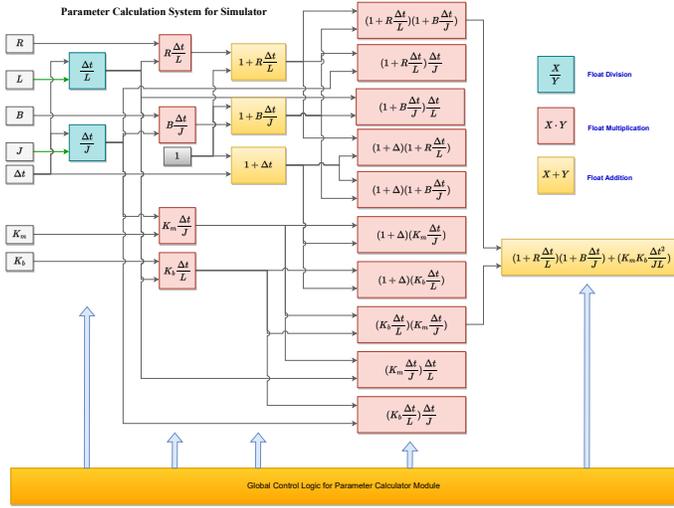


Fig. 2. Parameter calculation module of digital twin system

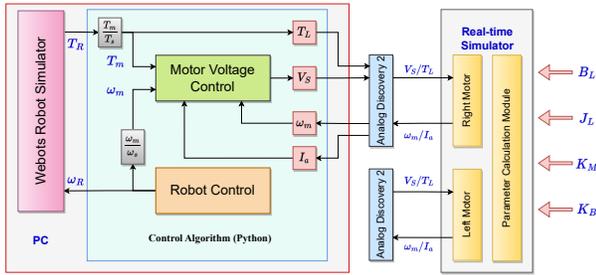


Fig. 3. RTDS and Webots robot simulator connection block diagram

signals.

The motor simulator modules get main input signals and calculate output signals  $I_a$  and  $\omega_m$  by using coefficients calculated by *Parameter Calculation* module. Both modules were designed to perform calculations with in  $10 \mu s$  as it is also one ADC read cycle. However the changes in main input signals can be seen after one simulation cycle, the change in parameter inputs were using an additional simulation cycle to be ready for use by main simulator module which means at least two cycles were required for the the change of parameters to effect the output signals.

### III. ROBOT SIMULATOR AN CONTROL CONCEPT

In this study, the Webots robot simulator was employed for the physical calculations of the robot. Open Dynamics Engine (ODE) was employed as the default physics engine in WEBOTS to perform physics calculations. Additionally, the definitions of the Turtlebot3 were utilized in the pre-defined PROTO file, created from WEBOTS components. The connection between this simulator (PC-side) and the Real Time Digital Simulator (RTDS) (FPGA-side) co-simulator was established using Python, which utilized the *pydfw* library to facilitate communication with Analog Discovery 2 devices. Within the Python program, a subprogram generated angular velocity information to be applied to the motors for robot movement. As illustrated in Fig. 3, Analog Discovery 2

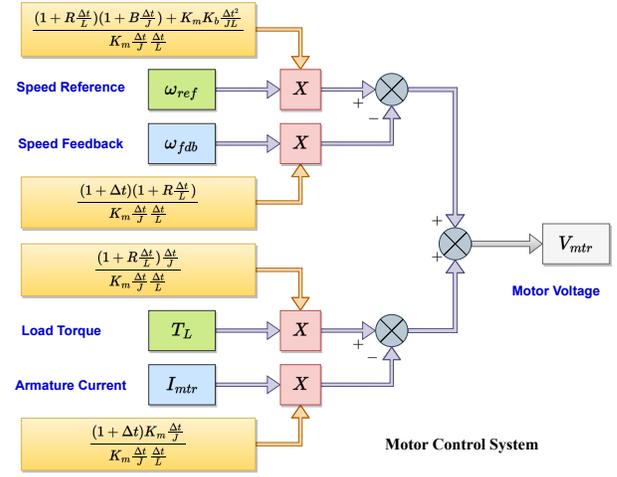


Fig. 4. Control system of motor voltage

devices served as analog input and output modules for data exchange.

The angular velocity values generated by the robot control algorithm were transmitted to the Webots robot simulator. Torque and angular velocity references from Webots were combined with armature current and angular velocity feedback information from the real-time co-simulator. These inputs were then used to calculate the supply voltage value for the digital motors. To derive this voltage value, in Eq. 2. is formulated by substituting the load torque,  $T_L(t)$ , for the input torque,  $T_i(t)$ , in Eq. 1. The calculation of the supply voltage value, crucial for motor application, involves a fixed value of 128 ms for  $\Delta t$ , as specified in the equation. However, given that  $\Delta t$  can vary during the simulation, it is treated as a variable and measured for use in the calculations.

$$\omega(t) = \frac{(1 + \Delta t) \left[ (1 + \Delta t \frac{R}{L}) \omega(t - \Delta t) + \frac{\Delta t K_m}{J} i(t - \Delta t) \right]}{(1 + \Delta t \frac{R}{L})(1 + \Delta t \frac{B}{J}) + \Delta t^2 \frac{K_m K_b}{JL}} + \frac{(1 + \frac{R \Delta t}{L}) \frac{\Delta t}{J} T_i(t) + \frac{\Delta t^2 K_m}{JL} V(t)}{(1 + \Delta t \frac{R}{L})(1 + \Delta t \frac{B}{J}) + \Delta t^2 \frac{K_m K_b}{JL}} \quad (1)$$

$$V(t) = \frac{(1 + R \frac{\Delta t}{L})(1 + B \frac{\Delta t}{J}) + K_m K_b \frac{\Delta t^2}{JL}}{K_m \frac{\Delta t^2}{JL}} \omega(t) + \frac{(1 + R \frac{\Delta t}{L}) \frac{\Delta t}{J}}{K_m \frac{\Delta t^2}{JL}} T_i(t) + \frac{(1 + \Delta t)}{K_m \frac{\Delta t^2}{JL}} K_m \frac{\Delta t}{J} i(t - \Delta t) - \frac{(1 + \Delta t)}{K_m \frac{\Delta t^2}{JL}} (1 + R \frac{\Delta t}{L}) \omega(t - \Delta t) \quad (2)$$

The calculations, as depicted in Eq. 2, are executed within the motor voltage control system illustrated in Fig. 4. These computations were performed to generate the supply voltage values applied to the motors. The obtained voltage and load torque values were transmitted to the real-time simulator through the Analog Discovery 2 device. Utilizing this information, the real-time simulator simulated the motors, generating

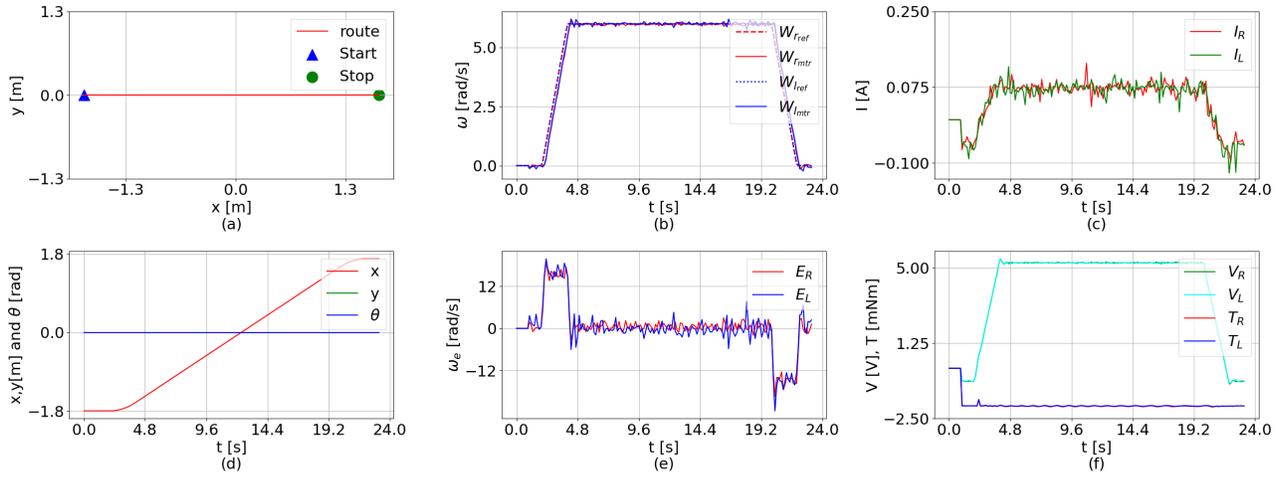


Fig. 5. Results for robot simulation for linear trajectory on a surface with inclination of 0.1 for  $x = 0$  axis rad and 2 kg of payload; a) trajectory followed by robot, b) speed references and simulated speed value for motors, c) armature currents for both motors, d) position and orientation of robot, e) error values for both motors angular speed values, f) simulated load torque and supply voltages for motors

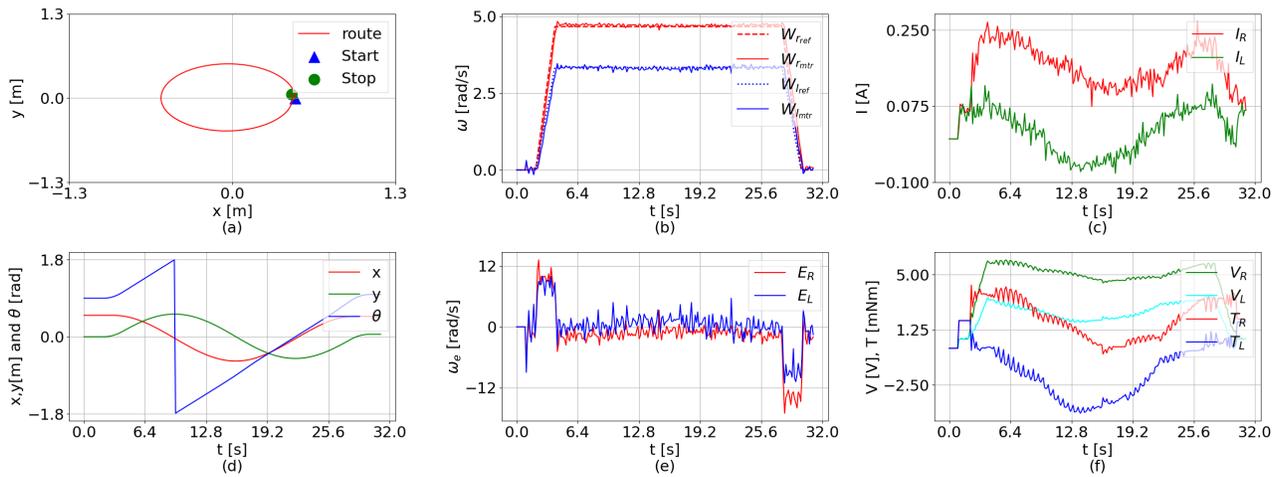


Fig. 6. Results for robot simulation for circular trajectory on a surface with slope of 0.1 rad for  $y = 0$  axis and 2 kg of payload; a) trajectory followed by robot, b) speed references and simulated speed value for motors, c) armature currents for both motors, d) position and orientation of robot, e) error values for both motors angular speed values, f) simulated load torque and supply voltages for motors

angular velocity and armature current signals. Subsequently, these calculated values were relayed back to the Python-based control program. The program also undertakes the task of saving these values to show the experimental results.

#### IV. EXPERIMENTAL RESULTS

##### A. Linear trajectory

The results shown in Fig. 5 are obtained for a robot carrying a 2 kg payload on a plane with a positive slope of 0.1 rad in the  $x$ -axis. As depicted in Fig. 5a, the robot travels approximately 4 meters linearly along the  $x$ -axis. The robot covered the distance between the start point and the end point in approximately 24 seconds. Fig. 5b demonstrates that the left and right motors accurately follow the reference velocity signals. The armature currents of the motors are illustrated in Fig. 5c. The robot's  $x$ ,  $y$  position, and rotation information

on the  $z$ -axis  $\theta$  are presented in Fig. 5d. Since the robot moves linearly, its rotation on the  $z$ -axis is zero. The error signals at angular velocities are shown in Fig. 5e. The high values in the error signals in the acceleration and deceleration regions are attributed to the control step time of the Webots simulator, which is 130 ms. The voltage signals obtained at the controller output and the torque values measured from the robot wheels are analyzed in Fig. 5f. As seen, the torque values are negative as the robot descends the ramp. In this case, both motors should operate in forward-braking (regenerating) mode. However, the total load torque of motor is the sum of both servo systems load torque and the load of the gearbox inside servo system. For lower speed of robot, the motors were operating in forward-braking mode and the armature current is reversed. But for higher speeds of robot, the required torque for the gearbox increased and motors need to compensate additional torque by working at forward-motor quarter. The

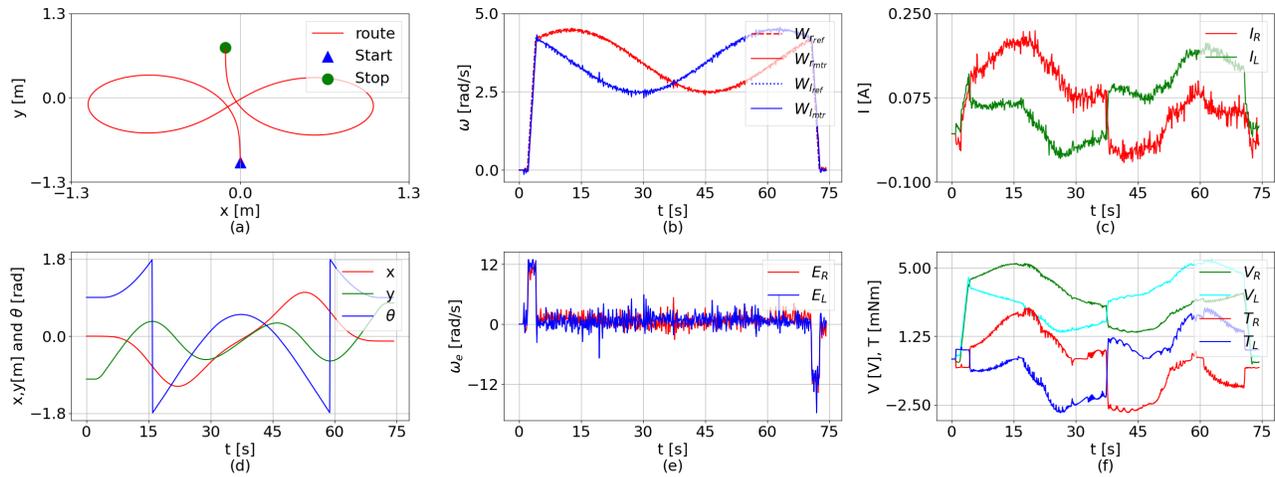


Fig. 7. Results for robot simulation for infinite symbol trajectory on a surface with inclination of 0.1 rad for  $x = 0$  axis; a) trajectory followed by robot, b) speed references and simulated speed value for motors, c) armature currents for both motors, d) position and orientation of robot, e) error values for both motors angular speed values, f) simulated load torque and supply voltages for motors

motors driving the robot are connected to the wheels through a reducer. Thus, the torque at the output of the reducer and the armature current patterns of the motors are dissimilar as shown in Fig. 5.

### B. Circular trajectory

As a different operation, the plane is set to a 0.1 rad slope in the  $y$ -axis. Subsequently, the robot is operated on a circular trajectory with a radius of 0.5m under the same load conditions. The graphs obtained under these conditions are presented in Fig. 6. The circular trajectory along which the robot moves is shown in Fig. 6a. Since the robot moves counter-clockwise, Fig. 6b illustrates that the left wheel rotates slower than the right wheel. In Fig. 6c, between 12-19 seconds, the left motor operates in regenerative mode, drawing negative current. This is because, at that moment, the motor continues to run at a positive speed. Fig. 6d depicts the robot rotating on the  $z$ -axis for a total of one revolution, oriented in the same direction. The error signals are presented in Fig. 6e, and voltage and torque signals are given in Fig. 6f.

### C. Infinite symbol trajectory

Finally, the robot is moved along a trajectory that resembles an infinity symbol. In Fig. 7, the robot is unloaded, and the plane is rotated on the  $x$ -axis by 0.1 rad in the positive direction to obtain a slope. The starting position of the robot is (0,-1), and the end position is (0,0.75). The trajectory of the robot is shown in Fig. 7a. Examining Fig. 7b reveals that the angular velocities of the motors are driven by sine and cosine references. In Fig. 7c, it can be observed that, at some points, the motor operates the generator mode by drawing negative current, performing a braking operation. In this case, the position and angle information of the robot is given in Fig. 7d. Additionally, control error signals are shown in Fig. 7e. Fig. 7f illustrates the voltages applied to the motor and the measured torque signals. Notably, both the left and right motors brake by producing negative torque in different regions.

## V. CONCLUSION

A study is presented to address a missing aspect of robot simulators, namely the electrical modeling of motors. In the proposed system, motor models were developed in real-time within the FPGA environment. This model was integrated with a PC-based robot simulator, introducing a new method to address deficiencies in robot simulation. In the study, the PC side of the robot simulator was responsible for the robot's dynamic features, while the FPGA-based co-simulator side took on the real-time operation of the electrical and mechanical models of the motors.

Upon examining the results of the study, it can be said that satisfactory outcomes were achieved. This work demonstrates the real-time feasibility of electrical analyses through the development of an additional module for robot simulators. As a future study, there are plans to create an FPGA-based robot simulator by establishing a system similar to the physical motor of the robot simulator discussed in this study.

## REFERENCES

- [1] M. U. Farooq, A. Eizad, and H.-K. Bae, "Power solutions for autonomous mobile robots: A survey," *Robotics and Autonomous Systems*, vol. 159, p. 104285, 2023.
- [2] G. Fragapane, D. Ivanov, M. Peron, F. Sgarbossa, and J. O. Strandhagen, "Increasing flexibility and productivity in industry 4.0 production networks with autonomous mobile robots and smart intralogistics," *Annals of operations research*, vol. 308, no. 1-2, pp. 125-143, 2022.
- [3] G. Fragapane, H.-H. Hvolby, F. Sgarbossa, and J. O. Strandhagen, "Autonomous mobile robots in hospital logistics," in *IFIP International Conference on Advances in Production Management Systems*. Springer, 2020, pp. 672-679.
- [4] L. Emmi, E. Le Flécher, V. Cadenat, and M. Devy, "A hybrid representation of the environment to improve autonomous navigation of mobile robots in agriculture," *Precision Agriculture*, vol. 22, pp. 524-549, 2021.
- [5] F. Luo, Q. Zhou, J. Fuentes, W. Ding, and C. Gu, "A soar-based space exploration algorithm for mobile robots," *Entropy*, vol. 24, no. 3, p. 426, 2022.
- [6] P. K. Panigrahi and S. K. Bisoy, "Localization strategies for autonomous mobile robots: A review," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 8, pp. 6019-6039, 2022.

- [7] G. Boztas and O. Aydogmus, "Implementation of pure pursuit algorithm for nonholonomic mobile robot using robot operating system," *Balkan Journal of Electrical and Computer Engineering*, vol. 9, no. 4, pp. 337–341, 2021.
- [8] O. Aydogmus and M. Yilmaz, "Comparative analysis of reinforcement learning algorithms for bipedal robot locomotion," *IEEE Access*, pp. 1–1, 2023.
- [9] S. Zhang, J.-t. Yao, Y.-b. Wang, Z.-s. Liu, Y.-d. Xu, and Y.-s. Zhao, "Design and motion analysis of reconfigurable wheel-legged mobile robot," *Defence Technology*, vol. 18, no. 6, pp. 1023–1040, 2022.
- [10] L. Tagliavini, G. Colucci, A. Botta, P. Cavallone, L. Baglieri, and G. Quaglia, "Wheeled mobile robots: state of the art overview and kinematic comparison among three omnidirectional locomotion strategies," *Journal of Intelligent & Robotic Systems*, vol. 106, no. 3, p. 57, 2022.
- [11] J. Yoon, B. Son, and D. Lee, "Comparative study of physics engines for robot simulation with mechanical interaction," *Applied Sciences*, vol. 13, no. 2, p. 680, 2023.
- [12] I. Tejado, J. Serrano, E. Pérez, D. Torres, and B. M. Vinagre, "Low-cost hardware-in-the-loop testbed of a mobile robot to support learning in automatic control and robotics," *IFAC-PapersOnLine*, vol. 49, no. 6, pp. 242–247, 2016.
- [13] Y. Chen, S. Chen, T. Zhang, S. Zhang, and N. Zheng, "Autonomous vehicle testing and validation platform: Integrated simulation system with hardware in the loop," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 949–956.
- [14] A. Hadizadeh, M. Hashemi, M. Labbaf, and M. Parniani, "A matrix-inversion technique for fpga-based real-time emt simulation of power converters," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 2, pp. 1224–1234, 2019.
- [15] C. Qi, F. Gao, X. Zhao, A. Ren, and Q. Wang, "A force compensation approach toward divergence of hardware-in-the-loop contact simulation system for damped elastic contact," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 4, pp. 2933–2943, 2017.
- [16] A. Fekik, H. Khati, A. T. Azar, M. L. Hamida, H. Denoun, I. A. Hameed, and N. A. Kamal, "Fpga in the loop implementation of the puma 560 robot based on backstepping control," *IET Control Theory & Applications*, 2023.
- [17] S. Dereli and R. Köker, "Hardware design of fpga-based embedded heuristic optimization technique for solving a robotic problem: Ic-pso," *Arabian Journal for Science and Engineering*, pp. 1–15, 2023.
- [18] E. Moréac, E. M. Abdali, F. Berry, D. Heller, and J.-P. Diguët, "Hardware-in-the-loop simulation with dynamic partial fpga reconfiguration applied to computer vision in ros-based uav," in *2020 International Workshop on Rapid System Prototyping (RSP)*. IEEE, 2020, pp. 1–7.
- [19] H. W. Dommel, "Digital computer solution of electromagnetic transients in single-and multiphase networks," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-88, no. 4, p. 388–399, April 1969.
- [20] *Cyclone V Device Overview*, May 2018.



hardware development, FPGA-based system design, and embedded system design.

**Mehmet Rıza SARAÇ** is a research assistant in the Department of Mechatronics Engineering at the University of Firat, where he has been a faculty member since 2011. He graduated with B.Sc. degree from the Department of Electrical - Electronics Engineering at Inonu University in 2002 and M.Sc. degrees from the Department of Mechatronics Engineering at the Firat University in 2016. Then he completed his Ph.D. degree at the Department of Mechatronics Engineering at the same university in 2023. His current research areas include computer hardware,



a highly respected researcher in the field of mechatronics engineering, and his work has made contributions to the development of new and innovative technologies in robotics and industrial automation.

**Omur Aydogmus** is a professor at Firat University, Faculty of Technology, specializing in Mechatronics Engineering. He earned his M.Eng. and Ph.D. degrees in electrical and electronics engineering from Firat University, Elazig, TURKIYE. His research primarily centers on collaborative robots, the integration of artificial intelligence in robotics, electric motor drive systems, and industrial automation applications. He has published numerous papers in peer-reviewed journals and conferences, and he has also received several grants for his research. He is