


ORIGINAL RESEARCH

FPGA in the loop implementation of the PUMA 560 robot based on backstepping control

Arezki Fekik^{1,2} | Hocine Khati³ | Ahmad Taher Azar^{4,5,6}  | Mohamed Lamine Hamida² | Hakim Denoun² | Ibrahim A. Hameed⁷ | Nashwa Ahmad Kamal⁸

¹Department of Electrical Engineering, University AkliMohandOulhadj-Bouria, Rue DrissiYahiaBouira, DrissiYahiaBouira, Algeria

²Electrical Engineering Advanced Technology Laboratory (LATAGE), University MouloudMammeri of Tizi-Ouzou, Tizi-Ouzou, Algeria

³Design and Drive of Production systems Laboratory, Department of Automation Faculty of Electrical and Computing Engineering, University MouloudMammeri of Tizi-Ouzou, Tizi-Ouzou, Algeria

⁴Automated Systems Soft Computing Lab (ASSCL), Prince Sultan University, Riyadh, Saudi Arabia

⁵College of Computer Information Sciences, Prince Sultan University, Riyadh, Saudi Arabia

⁶Faculty of Computers and Artificial Intelligence, Benha University, Benha, Egypt

⁷Department of ICT and Natural Sciences, Norwegian University of Science and Technology, Alesund, Norway

⁸Faculty of Engineering, Cairo University, Giza, Egypt

Correspondence

Ahmad Taher Azar, Automated Systems Soft Computing Lab (ASSCL), Prince Sultan University, Riyadh, Saudi Arabia.

Email: aazar@psu.edu.sa;

Ibrahim A. Hameed, Department of ICT and Natural Sciences, Norwegian University of Science and Technology, Larsgardsvegen, 2, 6009 Alesund, Norway.

Email: ibib@ntnu.no

Norwegian University of Science and Technology, Larsgardsvegen, Norway

Funding information

Norwegian University of Science and Technology, Grant/Award Number: TBA

Abstract

The objective of this article is to present the implementation of a backstepping control regulator on a Xilinx Zedboard Zynq FPGA using the HDL Coder tool through the FPGA in-the-loop option, and to study its effectiveness when applied to a three-dimensional robotic manipulator model. The analysis is based on the application of the backstepping control law for the three degrees of freedom PUMA 560 model, through the development of a dynamic simulation model. The results of practical implementation using the FPGA in the loop technique demonstrate the effectiveness of the proposed method using the Xilinx Zedboard Zynq FPGA, which provides a useful insight into the benefits of using backstepping control laws in robotics applications.

1 | INTRODUCTION

The Puma 560 robot is a computer-controlled industrial robot that was developed in the 1970s by the American company Unimation. Puma stands for “Programmable Universal Machine for Assembly”. The Puma 560 was designed to perform repetitive industrial tasks such as assembly, material handling, welding, and polishing [1–4]. The Puma 560 is a six-axis robot that uses electric actuators to rotate each joint. It is capable of precise and fast movements due to its numerical control system. The robot can also be equipped with special tools, such as grippers or lasers, to perform specific tasks. The Puma 560 was one of the first robots

to be widely used in manufacturing and paved the way for the use of robots to automate production processes [5–7]. Today, the Puma 560 has been replaced by more advanced robots, but it remains an important symbol of the evolution of industrial robotics and its impact on manufacturing production. The Puma 560 has found a new use in education, in part due to the fact that it is the most mathematically described robot [8–11]. Its simple structure allows for the development of new controllers and testing of new control algorithms for educational and scientific purposes. Nowadays, there are many manufacturers on the market, but the robots produced use controllers that are not open for research and education. In the educational process

This is an open access article under the terms of the [Creative Commons Attribution-NonCommercial-NoDerivs](https://creativecommons.org/licenses/by-nc-nd/4.0/) License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2023 The Authors. *IET Control Theory & Applications* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology.

organized for students, it is important to have the opportunity to measure different values (position, error, velocity, etc.) from the control algorithms used on the controller in real-time and compare them with the results of other simulations. Therefore, new control approaches, as well as controllers for the Puma 560 robot, have been developed in institutes and universities around the world. The Puma 560 robot was controlled using different techniques, including position and force control. Position control involves programming the robot to follow a specific path or sequence of movements, while force control allows the robot to detect and respond to forces and torques in its environment. These control techniques were achieved through the use of sophisticated sensors and feedback systems. The Puma 560 robot exhibited nonlinearity due to its mechanical design and control system. This means that its movements did not always follow a linear trajectory due to the complex interaction between its different axes of motion [12–14]. This nonlinearity made programming and controlling the robot more difficult, as factors such as the effect of gravity and mechanical friction had to be taken into account [15]. Engineers had to develop more advanced control techniques to compensate for these factors and ensure precise and repeatable movements of the robot.

In the field of control systems, addressing complex challenges in nonlinear dynamics has led to the development of novel control strategies [16–18]. This paper contributes to this area by proposing two distinct optimal control methods. The first method, presented in [19], focuses on finite-time optimal control for a specific category of nonlinear systems characterized by positive odd rational powers. The approach employs adaptive neural networks, power integrator technique, and a backstepping scheme to ensure semiglobal practical finite-time stability for the closed-loop systems. On the other hand, the second method, discussed in [20], addresses the adaptive fuzzy inverse optimal control design problem for uncertain strict-feedback nonlinear systems. By utilizing fuzzy logic systems, an equivalent system and an auxiliary system are established, and an adaptive fuzzy inverse optimal scheme is developed. This scheme not only guarantees input-to-state stabilizability but also achieves the objective of inverse optimality with respect to the cost functional. Through simulation studies and comparisons, the validity and efficacy of the proposed strategies are verified.

The integration of manipulator arm control can present a significant challenge, as the nonlinearity of the dynamic equations and strong coupling between them can make the process complex and difficult. These problems become increasingly difficult to avoid with traditional control algorithms, namely control with proportional, integral, and derivative components, especially when the dynamic requirements of the control loop are too strict. For this reason, backstepping control stands out as a beneficial method for achieving such behaviour due to its simplification of the system's dynamic model and insensitivity to disturbances introduced in the same direction as the origin of the input [21, 22].

The field-programmable gate array (FPGA) can be considered a valuable tool for implementing control of embedded systems due to its low power consumption, high-speed operations, and considerable data storage capacity, and can be considered for the optimal solution of implementing intelligent

control methods on such systems. It is proposed that the FPGA and real-time modules of LabVIEW software can introduce efficient hardware for implementation of control. The powerful FPGA enables fast execution speed, data processing parallelism, and reconfigurability that allow adaptation of the design and implementation of the proposed controller [23–28].

The majority of controllers in the field of control systems are intricate and demand significant computation time to converge towards desired performance. Consequently, their practical implementation necessitates powerful control boards in terms of computation capability and operating frequency, such as an FPGA. This study presents an FPGA implementation of backstepping controllers using the MATLAB Simulink environment, without resorting to low-level VHDL programming. As a result, the design time is substantially reduced, and the outcomes have demonstrated the efficacy of the adopted methodology. Furthermore, the utilization of fixed-point data types significantly diminishes hardware resource consumption and energy consumption in comparison to implementations reliant on floating-point arithmetic.

This paper focuses on implementing a backstepping control regulator on a Xilinx Zedboard Zynq FPGA using the HDL Workflow Advisor functionality through the FPGA In-the-Loop option applied to an industrial manipulator robot model of the PUMA 560 type with three degrees of freedom. Additionally, it also presents practical implementation results demonstrating the effectiveness and efficiency of the proposed control law. Furthermore, this research will provide a useful overview of the benefits of using the HDL Workflow Advisor functionality through the FPGA in-the-Loop option and backstepping control laws in robotics applications, such as improved tracking performance, robustness to parameter variations, and precise positioning with reduced chatter phenomena.

2 | RELATED WORK

Control of manipulator robots is challenging due to their high non-linearities, coupling effects, and external uncertainties and disturbances [29]. Various efforts have been made to address these issues, including PID control and fuzzy logic control [30]. Sliding mode control is an effective robust nonlinear control method as it provides the system dynamics with robust behavior against external disturbances and uncertainties once the system dynamics are driven onto the sliding surface [31]. In [32], a novel adaptive fractional-order non-singular terminal sliding mode (FO-NTSM) control strategy is proposed for an omnidirectional mobile robot manipulator (OMRM) with unknown parameters and external disturbances. The proposed method uses fuzzy wavelet neural networks (FWNNs) to estimate the dynamic uncertainty of the OMRM and design the adaptive NTSM controller to attenuate external disturbances by adjusting the weights of the FWNNs online. Additionally, the FO control criterion is employed to speed up the convergence of the algorithm, and a designed Lyapunov function is used to prove the globally robust stability of the OMRM control system. Simulation and experimental results are presented to show the feasibility and validity of the proposed method. This study

[33] focuses on designing robust control for nonlinear systems with full-state restrictions using sliding-mode theory. The chosen nonlinear system follows a standard Lagrangian structure and has nonparametric uncertainties. A barrier Lyapunov function and a time-varying gain are used to ensure the predefined state constraints, even under external perturbations. The proposed controller ensures convergence of the sliding surface in finite time to the origin, leading to asymptotic convergence of the states to the corresponding equilibrium point. The controller's finite-time stability is demonstrated using the second Lyapunov stability method. The proposed controller is applied to a two-link robotic manipulator, demonstrating better stabilization and tracking performance compared to traditional controllers. In [34], an adaptive recursive terminal sliding-mode (ARTSM) controller is proposed to improve the high-speed and high-precision performance of linear motor (LM) positioners, which are affected by payload variations, friction, and external disturbances. The controller employs a fast nonsingular terminal sliding function and a recursive integral terminal sliding function to ensure finite-time convergence of the sliding surfaces and zero tracking error. By setting an appropriate initial value for the integral element, the control system starts on the sliding surface at the initial time, reducing the reaching time. Stability analysis is presented to prove the effectiveness of the proposed ARTSM controller, and experimental results show significantly reduced tracking errors and faster disturbance rejection compared to a recently reported fast nonsingular terminal sliding-mode (FNTSM) controller for the LM positioner. A novel controller for rehabilitation robots using a combination of second-order fast nonsingular terminal sliding mode control (SOFNTSMC) and adaptive neural networks (ANNs) is presented in [35]. The controller is designed for a two-link single-arm robot with uncertainties and employs SOFNTSMC to achieve faster convergence and reduce chattering. ANNs are used to handle model uncertainties and disturbances without any prior knowledge. The proposed controller is verified through comparative simulations on a two-joint single-arm robot, demonstrating its effectiveness in achieving tracking control while maintaining stability in the closed-loop system. A backstepping holonomic tracking control method for embedded wheeled robots using an evolutionary fuzzy system with qualified ant colony optimization (ACO) is proposed in [36]. The method employs the Taguchi method to design an optimal ACO, which is then used to develop an evolutionary fuzzy system called FS-TACO. The proposed system is implemented on a field-programmable gate array (FPGA) for dynamic tracking control of three-wheeled holonomic mobile robots. The approach offers better population diversity, self-adaptive holonomic control, and avoids premature convergence compared to conventional control systems. The FPGA implementation using the system-on-a-programmable chip methodology of the proposed FS-TACO controller is more effective for real-world embedded applications. Experimental results and comparative studies demonstrate the effectiveness of the proposed FPGA-based FS-TACO controller for three-wheeled holonomic mobile robots. In this research [37], a PID-like fuzzy controller is constructed using a parallel structure consisting

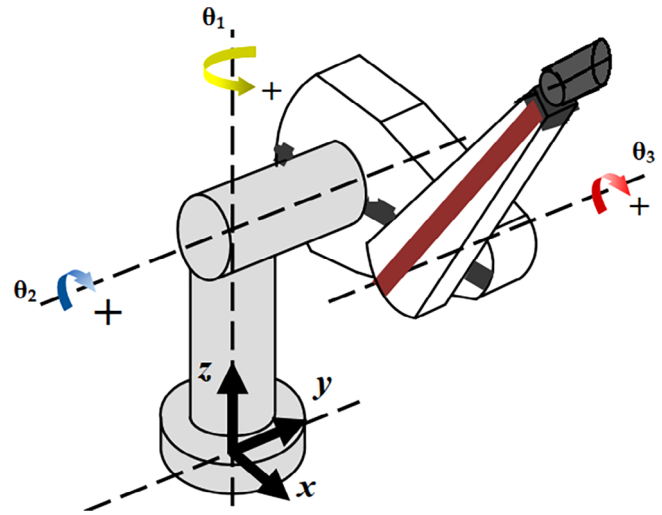


FIGURE 1 3DOF PUMA 560 Robot [39].

of a PD-like fuzzy controller and a PI-like controller to minimize the rule base. However, the backstepping controller works by canceling decoupling and nonlinear terms of the dynamic parameters of each link. This controller relies on the manipulator dynamic model and is highly sensitive to knowledge of all the parameters in the nonlinear robot manipulator's dynamic equation. The paper presented in [38] describes the real-time implementation of sliding mode control for a two-degree-of-freedom robot. The identification process uses a high-order recurrent neural network trained with a modified extended Kalman filter algorithm to enhance performance. Decentralized SMC technique is used to minimize the motion error. A Virtex 7 FPGA chip is configured to validate the proposed controller in a hardware-in-the-loop architecture. Experiments have demonstrated the algorithm's robustness against external disturbances and noise.

3 | DYNAMIC MODEL OF THE PUMA 560 ROBOT

This study is concerned with the development of a dynamic model for controlling the PUMA 560 industrial manipulator robot, as shown in Figure 1. The dynamic equations of motion form a set of mathematical expressions that describe the dynamic behaviour of the robot arm. These equations are essential for simulating the motion of the arm and designing effective control strategies. Different formulations can be used to derive the dynamic model of a manipulator, such as the Lagrange–Euler, Newton–Euler, and d'Alembert equations. The Lagrange–Euler formulation, which is based on an “energy-based” approach, is widely employed due to its simplicity and systematic nature. To apply the Lagrange–Euler formulation, the manipulator is assumed to be a serial chain of rigid links. The Lagrange–Euler equation relates the torque applied at joint i to drive link i of the manipulator to the kinetic energies, potential energies, and generalized coordinates of the

manipulator.

$$\tau_i = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} \quad (1)$$

where, $i = 1, 2, \dots, n$; L : Lagrangian function = $K - P$; K : total kinetic energy of the robot arm; P : total potential energy of the arm; q_i : generalized coordinates of the robot arm; \dot{q}_i : first derivative of the generalized coordinates of the robot arm; τ_i : generalized torque applied to the system at joint i to drive link i .

The Lagrange–Euler equation is employed to calculate the torque vector applied to the joints of each link in the manipulator. This equation yields the general form of the dynamic model of the manipulator, taking into account the kinetic and potential energies as well as the generalized coordinates of the system [40]

$$M(q)\ddot{q} + V(q, \dot{q})\dot{q} + G(q) = \Gamma \quad (2)$$

With

q : $n \times 1$ position vector, 25

$M(q)$: $n \times n$ inertia matrix of the manipulator,

$V(q, \dot{q})$: $n \times 1$ vector of Centrifugal and Coriolis terms

$G(q)$: $n \times 1$ vector of gravity terms Γ : $n \times 1$ vector of torques

By rewriting the term $V(q, \dot{q})$ dependent on velocity in a different way, all the matrices become functions of only the manipulator position. In this case, the dynamic equation is referred to as the configuration space equation and has the following expression:

$$\Gamma = M(q)\ddot{q} + B(q) \cdot [\dot{q} \cdot \dot{q}] + C(q) \cdot [\dot{q}^2] + G(q) \quad (3)$$

$B(q)$: $n \times n(n-1)/2$ matrix of Coriolis torques

$C(q)$: $n \times n$ matrix of Centrifugal torques

$[\dot{q} \cdot \dot{q}]$: $n(n-1)/2 \times 1$ vector of joint velocity products given by:

$$[\dot{q}_1 \cdot \dot{q}_2, \dot{q}_1 \cdot \dot{q}_3, \dots, \dot{q}_1 \cdot \dot{q}_n, \dot{q}_2 \cdot \dot{q}_3, \dot{q}_2 \cdot \dot{q}_4, \dots, \dot{q}_{n-2} \cdot \dot{q}_n, \dot{q}_{n-1} \cdot \dot{q}_n]^T$$

$[\dot{q}^2]$: $n \times 1$ vector given by: $[\dot{q}_1^2, \dot{q}_2^2, \dots, \dot{q}_n^2]$

It should be noted that this thesis only considers three links of the PUMA robot, with joint angles $q_4 = q_5 = q_6 = 0$. As a result, the dynamic equation can be expressed in the configuration space form.

$$\Gamma = A(q)\ddot{q} + B(q) \cdot [\dot{q} \cdot \dot{q}] + C(q) \cdot [\dot{q}^2] + G(q) \quad (4)$$

With,

Matrix A is a symmetric 6×6 matrix:

$$A(q) = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 & 0 & 0 \\ a_{21} & a_{23} & a_{23} & 0 & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 & a_{35} & 0 \\ 0 & 0 & 0 & a_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{66} \end{bmatrix} \quad (5)$$

With

$$\begin{aligned} a_{11} &= I_{m1} + I_1 + I_3 \cdot CC_2 + I_7 \cdot SS_{23} + I_{10} \cdot SC_{23} \\ &+ I_{11} \cdot SC_2 + I_{21} \cdot SS_{23} + 2[I_5 \cdot C_2 \cdot S_{23} + I_{21} \cdot C_2 \cdot C_{23} + I_{15} \cdot SS_{23} \\ &+ I_{16} \cdot C_2 \cdot S_{23} + I_{22} \cdot SC_{23}] \end{aligned}$$

$$a_{12} = I_4 \cdot S_2 + I_8 \cdot C_{23} + I_9 \cdot C_2 + I_{13} \cdot S_{23} + I_{18} \cdot C_{23}$$

$$a_{13} = I_8 \cdot C_{23} + I_{13} \cdot S_{23} - I_{18} \cdot C_{23}$$

$$a_{22} = I_{m2} + I_2 + I_6 + 2[I_5 \cdot S_3 + I_{12} \cdot C_2 + I_{15} + I_{16} \cdot S_3]$$

$$a_{23} = I_5 \cdot S_3 + I_6 + I_{12} \cdot C_3 + I_{16} \cdot S_3 + 2I_{15}$$

$$a_{33} = I_{m3} + I_6 + 2I_{15}$$

$$a_{35} = I_{15} + I_{17}$$

$$a_{44} = I_{m4} + I_{14}$$

$$a_{55} = I_{m5} + I_{17}$$

$$a_{66} = I_{m6} + I_{23}$$

$$a_{21} = a_{12}, a_{31} = a_{13}, \text{ and } a_{32} = a_{23}$$

matrix B is:

$$B(q) = \begin{bmatrix} b_{112} & b_{113} & 0 & b_{115} & 0 & b_{123} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & b_{214} & 0 & 0 & b_{223} & 0 & b_{225} & 0 & 0 & b_{235} & 0 & 0 & 0 & 0 \\ 0 & 0 & b_{314} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ b_{412} & b_{413} & 0 & b_{415} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & b_{514} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (6)$$

where

$$b_{112} = -2[-I_3 \cdot SC_2 + I_5 \cdot C_{223} + I_7 \cdot SC_{23} - I_{12} \cdot S_{223} \\ + I_{15} \cdot 2 \cdot SC_{23} + I_{16} \cdot C_{223} + I_{21} \cdot SC_{23} + I_5 \cdot (1 - SS_{23})] \\ + I_{10} \cdot (1 - 2 \cdot SS_{23}) + I_{11} \cdot (1 - 2 \cdot SS_2)$$

$$b_{113} = 2[I_5 \cdot C_2 \cdot C_{23} + I_7 \cdot SC_{23} - I_7 \cdot SC_{23} - I_{12} \cdot C_2 \cdot S_{23} \\ + I_{15} \cdot 2 \cdot SC_{23} + I_{16} \cdot C_2 \cdot C_{23} + I_{21} \cdot SC_{23} + I_{22} \cdot (1 - SS_{23})] \\ + I_{10} \cdot (1 - 2 \cdot SS_{23})$$

$$b_{115} = 2[-SC_{23} + I_{15} \cdot SC_{23} + I_{16} \cdot C_2 \cdot C_{23} + I_{22} \cdot CC_{23}]$$

$$b_{123} = 2[-I_8 \cdot S_{23} + I_{13} \cdot C_{23} + I_{18} \cdot S_{23}]$$

$$b_{214} = I_{14} \cdot S_{23} + I_{19} \cdot S_{23} + 2 \cdot I_{20} \cdot S_{23} \cdot (1 - 0.5)$$

$$b_{223} = 2[-I_{12} \cdot S_3 + I_5 \cdot C_3 + I_{16} \cdot C_3]$$

$$b_{225} = 2[I_{16} \cdot C_3 + I_{22}]$$

$$b_{314} = 2[I_{20} \cdot S_{23} \cdot (1 - 0.5)] + I_{14} \cdot S_{23} + I_{19} \cdot S_{23}$$

$$b_{412} = -b_{214} = -[I_{14} \cdot S_{23} + I_{19} \cdot S_{23} + 2I_{20} \cdot S_{23} \cdot (1 - 0.5)]$$

$$b_{413} = -b_{314} = -2[I_{20} \cdot S_{23} \cdot (1 - 0.5)] + I_{14} \cdot S_{23} + I_{19} \cdot S_{23}$$

$$b_{415} = -I_{20} \cdot S_{23} - I_{17} \cdot S_{23}$$

$$b_{514} = -b_{415} = I_{20} \cdot S_{23} + I_{17} \cdot S_{23}$$

matrix C is:

$$C(q) = \begin{bmatrix} 0 & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{21} & 0 & C_{23} & 0 & 0 & 0 \\ C_{31} & C_{32} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ C_{51} & C_{52} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (7)$$

Where:

$$C_{12} = I_4 \cdot C_2 - I_8 \cdot S_{23} - I_9 \cdot S_2 + I_{13} \cdot C_{23} + I_{18} \cdot S_{23}$$

$$C_{13} = 0.5b_{123} = -I_8 \cdot S_{23} + I_{13} \cdot C_{23} + I_{18} \cdot S_{23}$$

$$C_{21} = 0.5b_{112} = I_3 \cdot SC_2 - I_5 \cdot C_{223} - I_7 \cdot SC_{23} \\ + I_{12} \cdot S_{223} - I_{15} \cdot 2 \cdot SC_{23} - I_{16} \cdot C_{223} - I_{21} \cdot SC_{23} \\ - I_{22} \cdot (1 - 2 \cdot SS_{23}) - 0.5I_{10} \cdot (1 - 2 \cdot SS_{23})$$

$$- 0.5I_{11} \cdot (1 - 2 \cdot SS_2)$$

$$C_{23} = 0.5b_{223} = -I_{12} \cdot S_3 + I_5 \cdot C_3 + I_{16} \cdot C_3$$

$$C_{31} = 0.5b_{113} = -I_5 \cdot C_2 \cdot C_{23} - I_7 \cdot SC_{23} \\ + I_{12} \cdot C_2 \cdot S_{23} - I_{15} \cdot 2 \cdot SC_{23} - I_{16} \cdot C_2 \cdot C_{23} - I_{21} \cdot SC_{23} \\ - I_{22} \cdot (1 - 2 \cdot SS_{23}) - 0.5I_{10} \cdot (1 - 2 \cdot SS_{23})$$

$$C_{32} = -C_{23} = I_{12} \cdot S_3 - I_5 \cdot C_3 - I_{16} \cdot C_3$$

$$C_{51} = 0.5b_{115} = SC_{23} - I_{15} \cdot SC_{23} - I_{16} \cdot C_2 \cdot C_{23} - I_{22} \cdot CC_{23}$$

$$C_{52} = -0.5b_{225} = -I_{16} \cdot C_3 - I_{22}$$

And the matrix G is :

$$G(q) = \begin{bmatrix} 0 \\ G_2 \\ G_3 \\ 0 \\ G_5 \\ 0 \end{bmatrix} \quad (8)$$

With :

$$G_2 = G_1 \cdot C_2 + G_2 \cdot S_{23} + G_3 \cdot S_2 + G_4 \cdot C_{23} + G_5 \cdot S_{23}$$

$$G_3 = G_2 \cdot S_{23} + G_4 \cdot C_{23} + G_5 \cdot S_{23}$$

$$G_5 = G_5 \cdot S_{23}$$

And :

$$S_i = \sin \theta_i ; C_i = \cos \theta_i ; C_{ij} = \cos(\theta_i + \theta_j) ; S_{ijk} = \sin(\theta_i + \theta_j + \theta_k) ;$$

$$CC_i = \cos(\theta_i) \cdot \cos(\theta_j) ; CS_i = \cos(\theta_i) \cdot \sin(\theta_j)$$

4 | BACKSTEPPING CONTROL OF ROBOT PUMA 560

Backstepping control is a recursive design procedure used to construct feedback control laws and systematically select associated Lyapunov functions. The idea behind backstepping is to consider certain states as “pseudo-controls” for other states and to use Lyapunov functions to ensure the stability of controlled states. When implementing backstepping control, a virtual control law is assigned to the first state, and a Lyapunov candidate function for control is defined. The virtual control expression is then selected to ensure a negative derivative of the Lyapunov function, which guarantees system stability. The designer continues this process by assigning virtual control laws and choosing Lyapunov functions for subsequent states until all

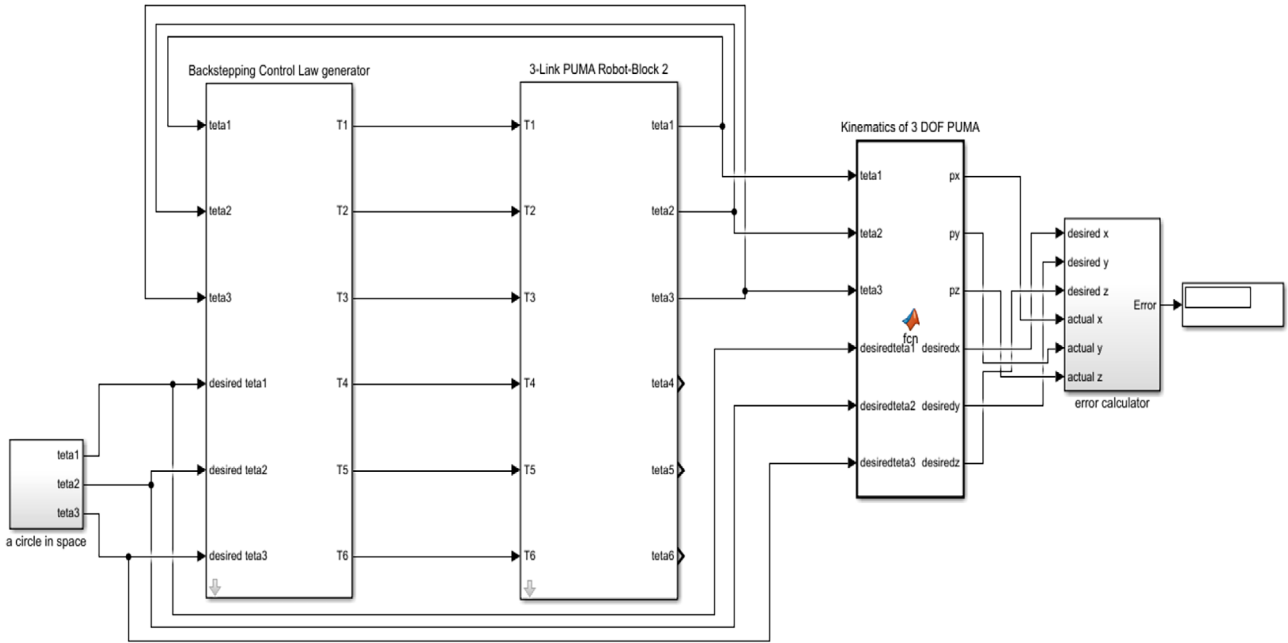


FIGURE 3 PUMA 560 simulink model.

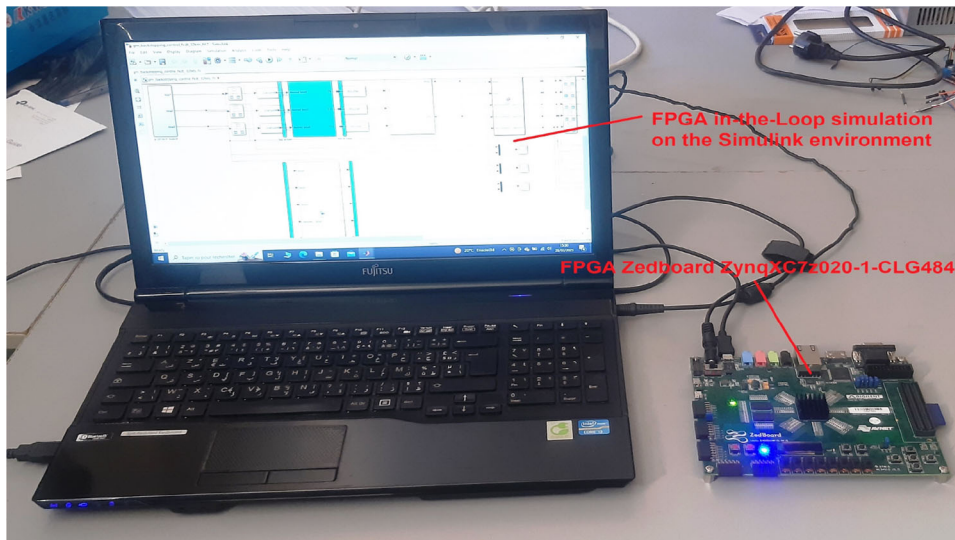


FIGURE 4 Configuration of the FPGA.

By selecting the value of x_3 (\dot{x}_2) to render \dot{V}_2 negative definite, the second virtual control is chosen, which introduces an augmented CLF.

$$\begin{aligned} \alpha_2 &= (x_3)_d = (k_1^2 - 1) \cdot e_1 - (k_1 + k_2) \cdot e_2 + \ddot{j}_{\text{desired}} = \ddot{x} \\ &= f(x) + g(x) \cdot u \end{aligned} \quad (19)$$

And u is found as to get

$$g^{-1}(x) \cdot [(k_1^2 - 1) \cdot e_1 - (k_1 + k_2) \cdot e_2 + \ddot{j}_{\text{desired}} - f(x)] \quad (20)$$

Figure 2 depicts the proposed implementation of the controller intended for use on a ZedBoard FPGA card. This implementation employs the “FPGA in the Loop” technique for robotic manipulators, specifically the PUMA 560, to ensure stability.

5 | EXPERIMENTAL WORK

This section will focus on presenting the principle of implementing the Simulink model using the HDL Coder tool, with the “HDL WorkflowAdvisor” functionality and the “FPGA in-the-Loop” option. This involves creating a Simulink model that describes the system behaviour, generating VHDL/Verilog

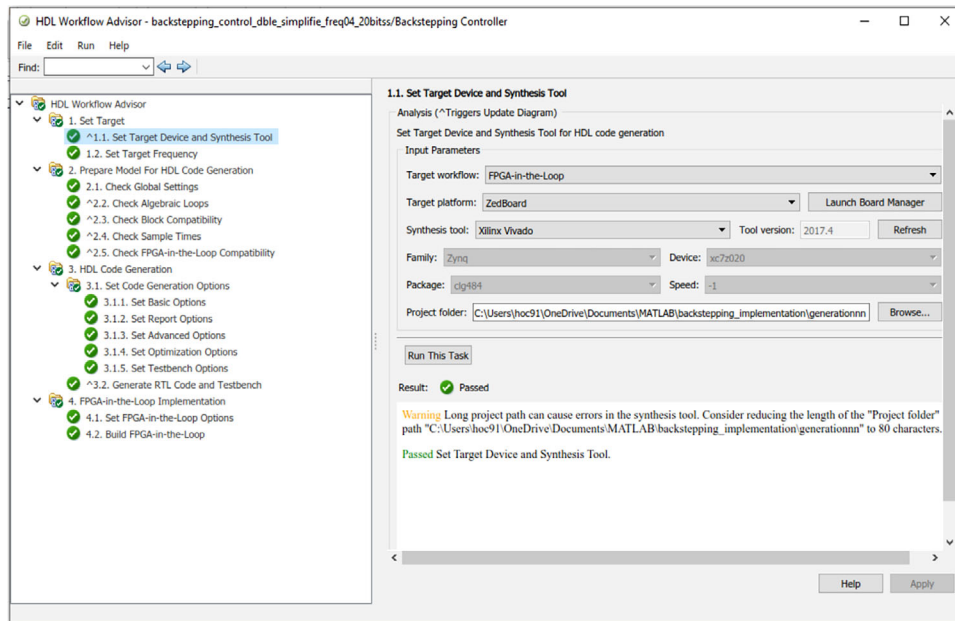


FIGURE 5 VHDL code integration into the FPGA platform.

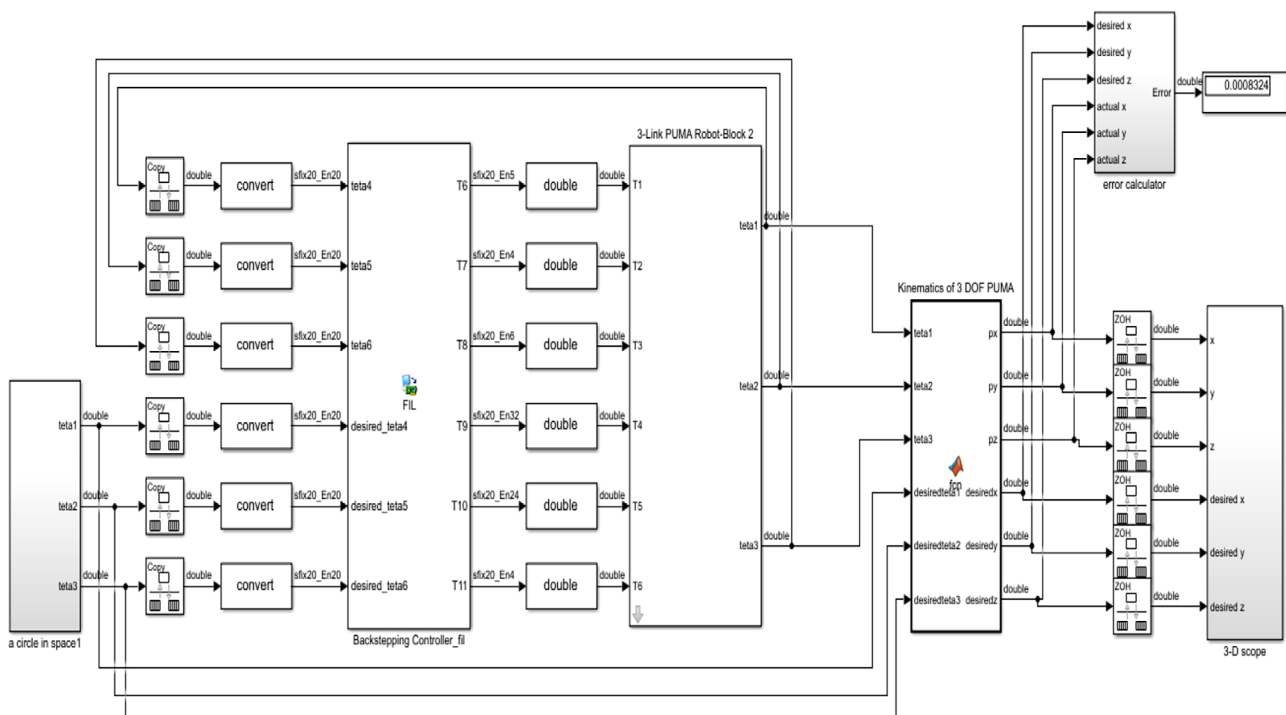


FIGURE 6 Simulation of the FPGA-in-the-Loop.

code from this model, configuring the FPGA platform, integrating the VHDL/Verilog code into the FPGA platform, simulating the Simulink model on the FPGA platform, and physically deploying the system. The “HDL WorkflowAdvisor” functionality guides the user through each step of the implementation process.

To achieve this, the following steps must be followed:

- 1) Creation of the Simulink model: This step involves creating a simulation model in Matlab/Simulink of the PUMA 560 robot controlled by the Backstepping technique with double precision floating-point data type, as shown in Figure 3.
- 2) Configuration of the FPGA platform involves setting up the FPGA board, in this case, the Xilinx Zedboard Zynq

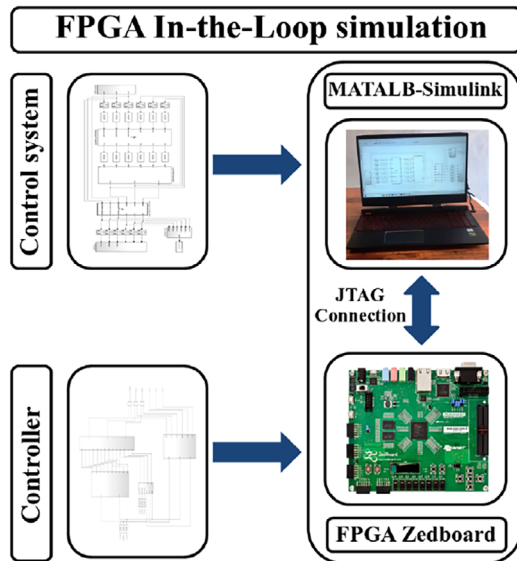


FIGURE 7 FPGA in-the-Loop simulation diagram.

XC7z020-1-CLG484, with a sampling period of 0.0001 s, as shown in Figure 4. The fixed-point data type should be used with a data size of 20 bits. The synthesized model from Simulink is converted from double-precision floating-point to a fixed-point model with 20 bits using the “fixed-point tool”. This is necessary because the Zedboard does not support the double-precision floating-point data type. Moreover, using fixed-point allows generating optimal VHDL code in terms of hardware resources consumed during the implementation of the controller.

- 3) Integrating VHDL code into the FPGA platform involves the user integrating the VHDL code generated by the HDL Coder tool into the FPGA platform [43]. This step includes compiling the code, placing, and routing the components on the FPGA board, as shown in Figure 5.
- 4) Simulation using FPGA in-the-Loop: The user can utilize the “FPGA in-the-Loop” feature of the HDL Coder tool to simulate the Simulink model in the context of the FPGA platform. The simulation, as shown in Figure 6, allows for the verification of the system’s behavior and the detection of design errors. In this case, a FIL Simulink block is generated based on the VHDL language, which allows the controller to run on the Zedboard while the rest of the system is executed in the Simulink environment.

6 | EXPERIMENTAL RESULTS AND DISCUSSION

The developed algorithm is simulated in the MATLAB-Simulink environment with a robot model using the FPGA in-the-Loop (FIL) technique to test their behaviors before they are applied to the real system. The controller is developed using

Simulink’s HDL Coder blocks with a double-precision floating-point data type. Subsequently, these models are converted into 20-bit fixed-point models, and their VHDL code is generated using HDL Coder’s “HDL workflow advisor” functionality. The generated VHDL code is ultimately implemented on the ZedBoard FPGA card, and then the entire control system is simulated in the Simulink environment using the FIL feature. The generated VHDL codes are executed on the ZedBoard to control the Puma robot model. Data exchange between the FPGA card and the Simulink environment occurs at each sampling period through the JTAG communication protocol. The concept of FIL simulation is illustrated in the following Figure 7.

The parameters that are taken into account to test the proposed controller include varying the reference trajectory and system dynamics, as well as analyzing the resources consumed by the FPGA board.

The results presented in Figures 8–12 effectively validate the real-time efficiency of the control technique in guiding the motion of the PUMA 560 robot.

Illustrated in Figure 8, the real-time angular positions of the robot’s three joints are showcased. Employing the “HDL workflow advisor” tool and harnessing the FPGA in-the-Loop capability, these angular positions demonstrate an asymptotic convergence towards the desired positions, achieving a remarkable level of accuracy. This convergence is achieved through the backstepping control strategy, which effectively guides the system to its target positions. Importantly, the implementation of backstepping control using FPGA in-the-Loop offers notable advantages. The utilization of FPGA hardware provides rapid and efficient execution of control algorithms, ensuring timely responses to dynamic changes. This is particularly evident in the observed reaction times for each joint: the first joint exhibits a reaction time of 0.3 s, the second joint responds in 0.35 s, and the third joint achieves a reaction time of approximately 0.37 s. These swift response times underscore the ability of the FPGA-accelerated backstepping control to promptly adapt to varying conditions, contributing to enhanced control precision and overall system stability.

Depicted in Figure 10, the trajectory tracking error of Cartesian positions, governed by the backstepping control technique, is showcased in the context of controlling the motion of the PUMA 560 robot. The presented results unveil noteworthy insights. At the initial stages of robot movement, representing the transient period, the average trajectory tracking error stands at approximately 0.0016. As the robot settles into steady-state operation, this error margin increases slightly to around 0.002. These minute error values underscore the impressive precision achieved in trajectory tracking through the application of the backstepping control method for the PUMA 560 robot. Additionally, it is worth highlighting the instrumental role played by the FPGA board, facilitated by the “HDL workflow advisor” tool and the FPGA In-the-Loop functionality. This combination has likely contributed significantly to the observed precision levels. By enabling rapid and accurate real-time execution of control commands, the FPGA-based approach enhances the overall precision of the control system. This aspect is particularly beneficial for real-time applications, where swift

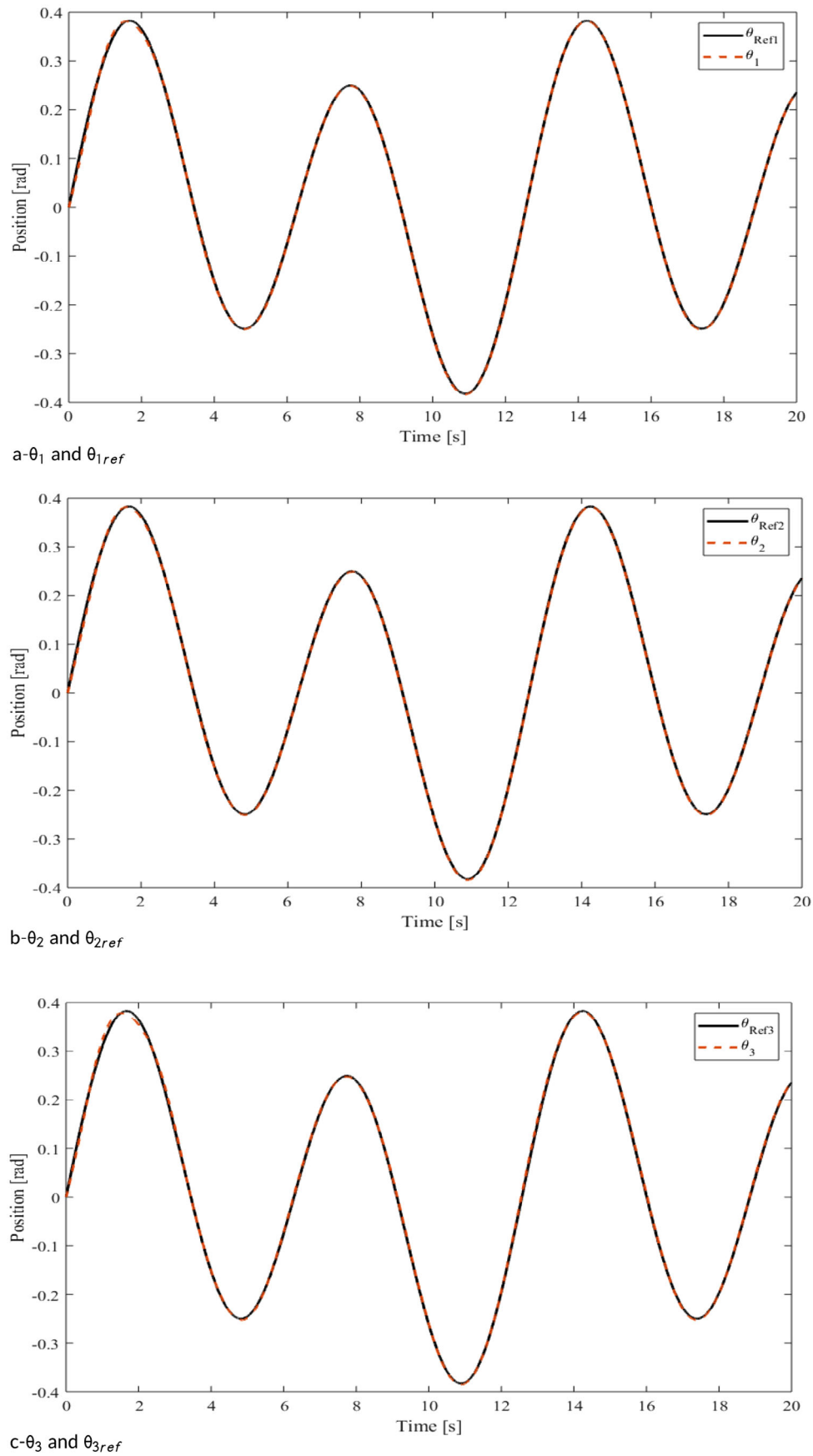


FIGURE 8 Angular position for the three degrees of freedom.

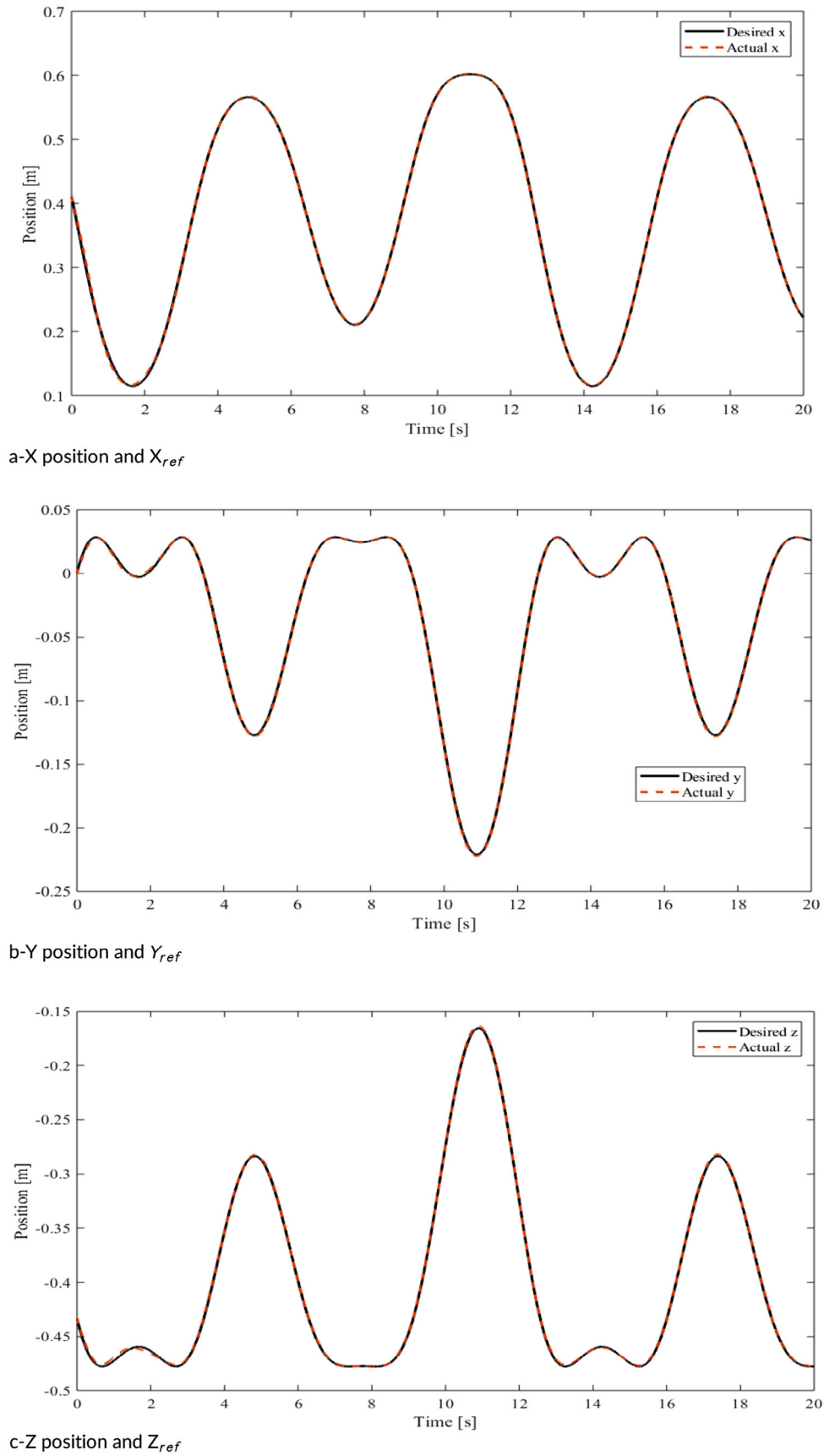


FIGURE 9 Joint trajectory and their references of robot PUMA 560.

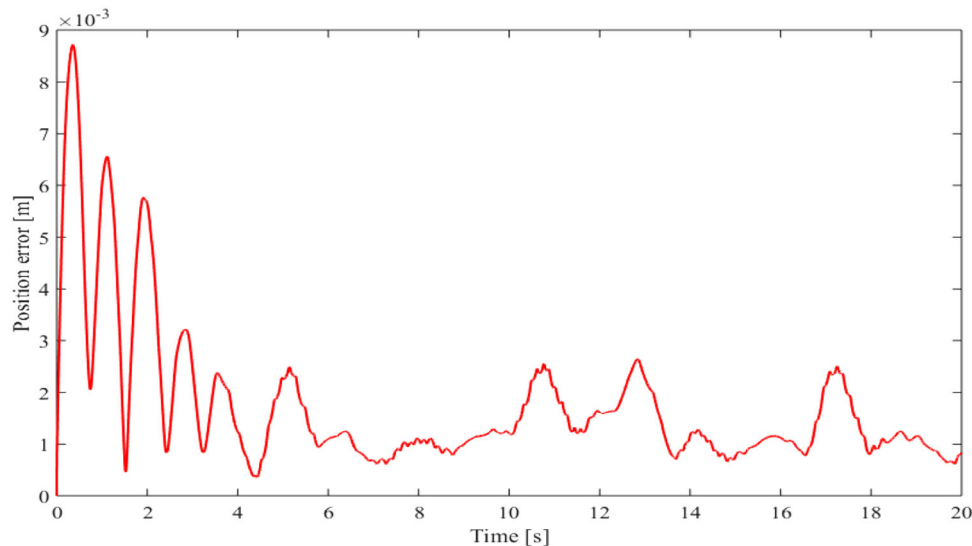


FIGURE 10 Trajectory tracking error.

and precise control responses are essential for achieving both stability and accuracy in the control of robotic systems.

In Figures 9 and 11, the position and velocity were compared in the three Cartesian directions (X , Y , and Z). These figures provide information on the performance of the backstepping control technique in terms of its ability to accurately follow desired trajectories in three-dimensional space. Furthermore, the results concern the implementation of the backstepping control on an FPGA board using the “HDL workflow advisor” feature through the FPGA In-the-Loop option. This means that the control has been implemented in FPGA hardware, which offers higher processing speed compared to a software implementation. This can partly explain the fast reaction times observed for the robot’s joints, as the control is executed more quickly in FPGA hardware. Additionally, the use of the FPGA in-the-Loop option allows for real-time verification of the control using FPGA hardware inputs and outputs, ensuring increased accuracy for the control. Overall, the use of the FPGA in-the-Loop option may have contributed to the satisfactory performance of the backstepping control results for the PUMA 560 robot.

The insights derived from Figure 12 shed light on the resource utilization pattern of the ZED-BOARD ZYNQ FPGA board during the implementation of backstepping control for the PUMA 560 robot. Specifically, these results delineate a compelling picture. The consumption of DSP resources accounts for 76% of the available 220 DSPs on the FPGA board, while the utilization of lookup tables (LUTs) stands at 61% of the available 53,200 LUTs. These figures offer a clear perspective on the hardware resources necessary for the successful deployment of the robot controller. An essential facet to highlight is that the consumption of other resources on the FPGA board remains notably low, remaining under 30%. This observation underscores the FPGA board’s competence in processing robot control commands with a prudent allocation of resources. This bodes well for a range of real-time robot control

applications, as the FPGA’s efficient resource utilization could potentially be harnessed for diverse control scenarios. Furthermore, the strategic use of 20-bit fixed-point data types emerges as a key factor in achieving this optimization. Despite the algorithm incorporating several non-linear functions like cosine and sine, the choice of fixed-point representation has enabled the generation of VHDL code that maximizes hardware resource efficiency. This judicious approach in data representation aligns well with the FPGA’s capacity, contributing to an effective balance between precision and resource utilization in the real-time control of the PUMA 560 robot.

7 | CONCLUSION

This article presents the implementation of a backstepping controller on the Xilinx Zedboard Zynq FPGA using the HDL Workflow Advisor feature through the FPGA in-the-Loop option, and its application to a three-dimensional robotic manipulator model. The practical implementation results demonstrate the effectiveness of the proposed method and the benefits of using backstepping control laws in robotics applications. The study shows that the backstepping control law is a viable solution for controlling the three degrees of freedom PUMA 560 model and can effectively track reference trajectories while ensuring system stability and convergence of errors to zero. The use of FPGA in the loop simulation provides a realistic and accurate way to test the control law in real-time, and the results show that the proposed method is both efficient and effective. In terms of future work, several potential directions can be explored. One possible avenue is to extend the study to investigate the performance of the backstepping control law on other robotic models with different degrees of freedom. Additionally, the study can be further extended by considering the effect of external disturbances and uncertainties on the performance of the control law.

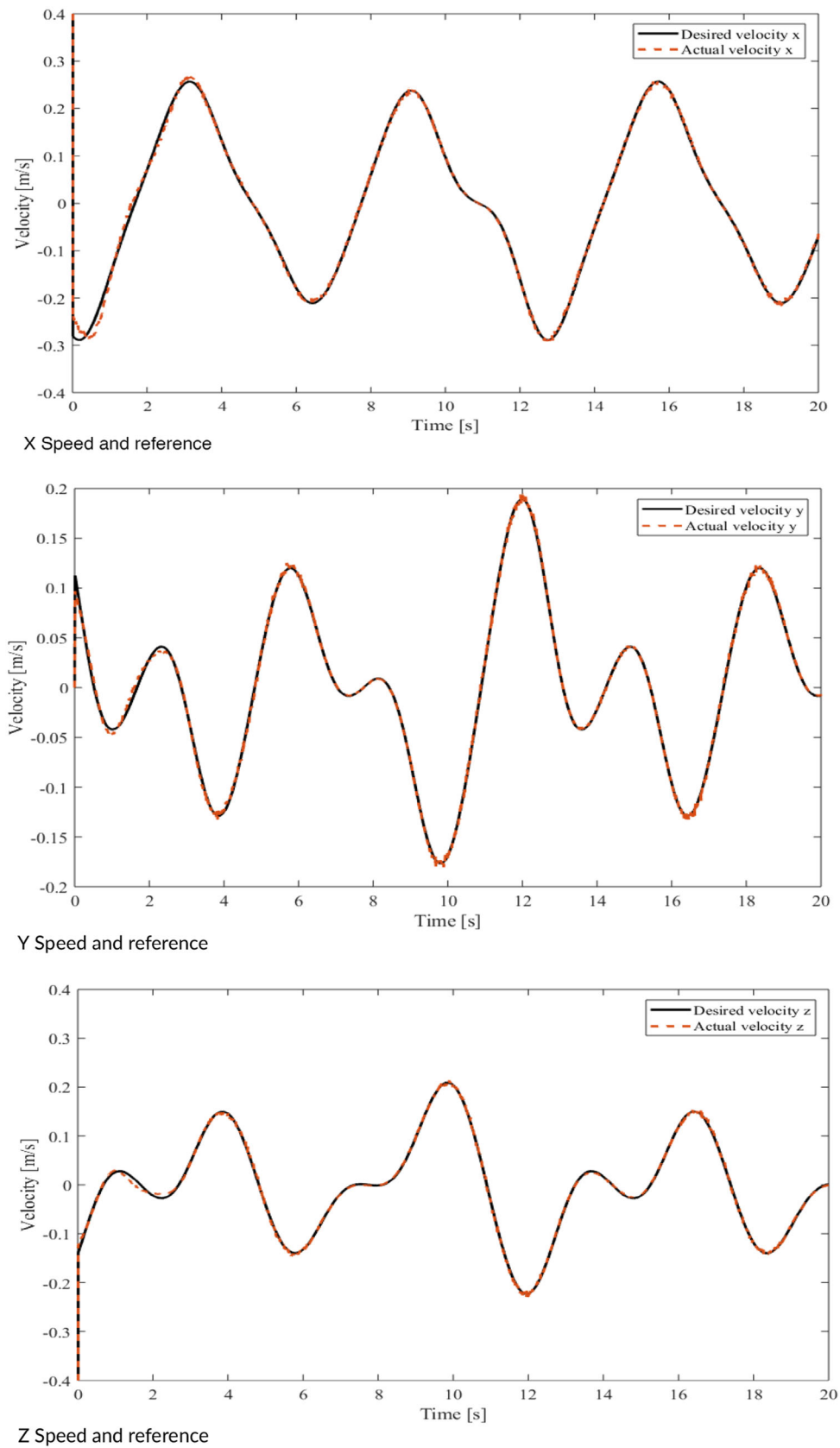


FIGURE 11 Joint speed and references.

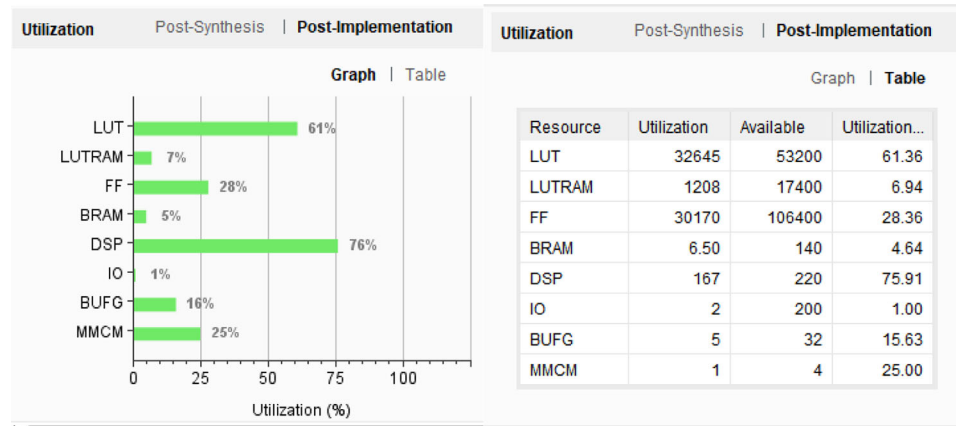


FIGURE 12 Resources consumed by the FPGA board.

AUTHOR CONTRIBUTIONS

Arezki Fekik: Conceptualization; formal analysis; methodology; resources; software; writing - original draft; writing - review and editing. **Hocine Khati:** Conceptualization; data curation; formal analysis; methodology; resources; visualization; writing - review and editing. **Ahmad Taher Azar:** Conceptualization; formal analysis; investigation; methodology; supervision; validation; visualization; writing - original draft; writing - review and editing. **Mohamed Lamine Hamida:** Conceptualization; data curation; formal analysis; methodology; resources; software; supervision; writing - review and editing. **Hakim Denoun:** Conceptualization; data curation; formal analysis; methodology; resources; software; supervision; writing - review and editing. **Ibrahim Hameed:** Data curation; formal analysis; funding acquisition; methodology; resources; software; validation; writing - review and editing. **Nashwa Ahmad Kamal:** Conceptualization; formal analysis; investigation; methodology; resources; software; validation; visualization; writing - review and editing.

ACKNOWLEDGMENTS

The authors would like to acknowledge the support of the Norwegian University of Science and Technology for paying the Article Processing Charges (APC) of this publication. The authors would like to thank Prince Sultan University, Riyadh, Saudi Arabia for its support. Special acknowledgement to Automated Systems & Soft Computing Lab (ASSCL), Prince Sultan University, Riyadh, Saudi Arabia. In addition, the authors wish to acknowledge the editor and anonymous reviewers for their insightful comments, which have improved the quality of this publication.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT

Data sharing not applicable - no new data generated, or the article describes entirely theoretical research.

ORCID

Ahmad Taher Azar  <https://orcid.org/0000-0002-7869-6373>

REFERENCES

- Singh, B., Sellappan, N., Kumaradhas, P.: Evolution of industrial robots and their applications. *Int. J. Emerg. Technol. Adv. Eng.* 3(5), 763–768 (2013)
- Hong, T.S., Ghobakhloo, M., Khaksar, W.: Robotic welding technology. *Compr. Mater. Process.* 6, 77–99 (2014)
- Harmon, L.D.: Automated tactile sensing. *Int. J. Rob. Res.* 1(2), 3–32 (1982)
- Cho, H.S.: R & d at the laboratory for control systems and automation of the Korea advanced institute of science and technology. *Robotica* 8(4), 333–338 (1990)
- Siepel, F.J., Maris, B., Welleweerd, M.K., Groenhuis, V., Fiorini, P., Stramigioli, S.: Needle and biopsy robots: a review. *Curr. Rob. Rep.* 2, 73–84 (2021)
- Troccaz, J., Dagnino, G., Yang, G.Z.: Frontiers of medical robotics: from concept to systems to clinical translation. *Annu. Rev. Biomed. Eng.* 21, 193–218 (2019)
- Abdi, E., Kulić, D., Croft, E.: Haptics in teleoperated medical interventions: force measurement, haptic interfaces and their influence on user's performance. *IEEE Trans. Biomed. Eng.* 67(12), 3438–3451 (2020)
- Dahmane, S.A., Slimane, A., Chaib, M., Kadem, M., Nehari, L., Slimane, S.A., et al.: Analysis and compensation of positioning errors of robotic systems by an interactive method. *J. Braz. Soc. Mech. Sci. Eng.* 45(2), 119 (2023)
- Llopis-Albert, C., Rubio, F., Valero, F.: Modelling an industrial robot and its impact on productivity. *Mathematics* 9(7), 769 (2021)
- Bilancia, P., Schmidt, J., Raffaelli, R., Peruzzini, M., Pellicciari, M.: An overview of industrial robots control and programming approaches. *Appl. Sci.* 13(4), 2582 (2023)
- Nguyen, V., Melkote, S.: Hybrid statistical modelling of the frequency response function of industrial robots. *Rob. Comput. Integr. Manuf.* 70, 102134 (2021)
- Tan, N., Yu, P.: Robust model-free control for redundant robotic manipulators based on zeroing neural networks activated by nonlinear functions. *Neurocomputing* 438, 44–54 (2021)
- Vo, A.T., Truong, T.N., Kang, H.J.: A novel prescribed-performance-tracking control system with finite-time convergence stability for uncertain robotic manipulators. *Sensors* 22(7), 2615 (2022)
- Vo, A.T., Truong, T.N., Kang, H.J.: A novel tracking control algorithm with finite-time disturbance observer for a class of second-order nonlinear systems and its applications. *IEEE Access* 9, 31373–31389 (2021)
- Ajeil, F.H., Ibraheem, I.K., Azar, A.T., Humaidi, A.J.: Autonomous navigation and obstacle avoidance of an omnidirectional mobile robot using

- swarm optimization and sensors deployment. *Int. J. Adv. Rob. Syst.* 17(3), 1–15 (2020)
16. Ammar, H.H., Azar, A.T., Shalaby, R., Mahmoud, M.I.: Metaheuristic optimization of fractional order incremental conductance (FO-INC) maximum power point tracking (MPPT). *Complexity* 2019, 1–13 (2019)
 17. Meghni, B., Dib, D., Azar, A.T., Saadoun, A.: Effective supervisory controller to extend optimal energy management in hybrid wind turbine under energy and reliability constraints. *Int. J. Dyn. Contr.* 6(1), 369–383 (2018)
 18. Meghni, B., Dib, D., Azar, A.T., Ghoulbourk, S., Saadoun, A.: Robust adaptive supervisory fractional order controller for optimal energy management in wind turbine with battery storage. In: Azar, A.T., Vaidyanathan, S., Ouannas, A. (eds.) *Fractional Order Control and Synchronization of Chaotic Systems*, Studies in Computational Intelligence, vol. 688, pp. 165–202. Springer, Cham (2017)
 19. Li, Y., Yang, T., Tong, S.: Adaptive neural networks finite-time optimal control for a class of nonlinear systems. *IEEE Trans. Neural Networks Learn. Syst.* 31(11), 4451–4460 (2019)
 20. Li, Y.m., Min, X., Tong, S.: Adaptive fuzzy inverse optimal control for uncertain strict-feedback nonlinear systems. *IEEE Trans. Fuzzy Syst.* 28(10), 2363–2374 (2019)
 21. Arefi, M.M., Vafamand, N., Homayoun, B., Davoodi, M.: Command filtered backstepping control of constrained flexible joint robotic manipulator. *IET Control Theory Appl.* (2023) <https://doi.org/10.1049/cth2.12528>
 22. Sehgal, K., Upadhyaya, S., Verma, M., Rayguru, M., et al.: Backstepping based trajectory tracking control of a class of reconfigurable mobile robot. In: 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), pp. 752–756. IEEE, Piscataway, NJ (2021)
 23. Kim, D.: An implementation of fuzzy logic controller on the reconfigurable FPGA system. *IEEE Trans. Ind. Electron.* 47(3), 703–715 (2000)
 24. Jamieson, P., Blank, D., Ghanem, J., McGrew, T., Corti, G.: A methodology for an FPGA implementation of a programmable logic controller to control an atomic layer deposition system. *Int. J. Reconfigurable Comput.* 2022, 8827417 (2022)
 25. Hasegawa, K., Takasaki, K., Nishizawa, M., Ishikawa, R., Kawamura, K., Togawa, N.: Implementation of a ROS-based autonomous vehicle on an FPGA board. In: 2019 International Conference on Field-Programmable Technology (ICFPT), pp. 457–460. IEEE, Piscataway, NJ (2019)
 26. Cañas, J.M., Fernández-Conde, J., Vega, J., Ordóñez, J.: Reconfigurable computing for reactive robotics using open-source FPGAs. *Electronics* 11(1), 8 (2021)
 27. Dirik, M., Kocamaz, A.F., Donmez, E.: Implementation of fuzzy controller for mobile robot navigation on NI's embedded-FPGA robotic platform. *Comput. Sci.* 4(2), 80–87 (2019)
 28. Estrada, L., Vázquez, N., Vaquero, J., de Castro, Á., Arau, J.: Real-time hardware in the loop simulation methodology for power converters using labview FPGA. *Energies* 13(2), 373 (2020)
 29. Chen, B., Zhang, R., Zhou, F., Du, W.: An observer-driven distributed consensus braking control method for urban railway trains with unknown disturbances. *Actuators* 12(3), 111 (2023)
 30. Sarkhel, P., Banerjee, N., Hui, N.B.: Fuzzy logic-based tuning of PID controller to control flexible manipulators. *SN Comput. Sci.* 2, 1–11 (2020)
 31. Alsubaie, H., Yousefpour, A., Alotaibi, A., Alotaibi, N.D., Jahanshahi, H.: Stabilization of nonlinear vibration of a fractional-order arch mems resonator using a new disturbance-observer-based finite-time sliding mode control. *Mathematics* 11(4), 978 (2023)
 32. Wu, X., Huang, Y.: Adaptive fractional-order non-singular terminal sliding mode control based on fuzzy wavelet neural networks for omnidirectional mobile robot manipulator. *ISA Trans.* 121, 258–267 (2022)
 33. Cruz-Ortiz, D., Chairez, I., Poznyak, A.: Sliding-mode control of full-state constraint nonlinear systems: a barrier Lyapunov function approach. *IEEE Trans. Syst. Man Cybern. Syst.* 52(10), 6593–6606 (2022)
 34. Shao, K., Zheng, J., Huang, K., Wang, H., Man, Z., Fu, M.: Finite-time control of a linear motor positioner using adaptive recursive terminal sliding mode. *IEEE Trans. Ind. Electron.* 67(8), 6659–6668 (2019)
 35. Liang, X., Wang, H., Zhang, Y.: Adaptive nonsingular terminal sliding mode control for rehabilitation robots. *Comput. Electr. Eng.* 99, 107718 (2022)
 36. Huang, H.C., Chiang, C.H.: Backstepping holonomic tracking control of wheeled robots using an evolutionary fuzzy system with qualified ant colony optimization. *Int. J. Fuzzy Syst.* 18, 28–40 (2016)
 37. Khajeh, A., Piltan, F., Rashidian, M.R., Salehi, A., et al.: Design new intelligent PID like fuzzy backstepping controller. *Int. J. Mod. Educ. Comput. Sci.* 6(2) (2014)
 38. Davalos-Guzman, U., Castañeda, C.E., Aguilar-Lobo, L.M., Ochoa-Ruiz, G.: Design and implementation of a real time control system for a 2DOF robot based on recurrent high order neural network using a hardware in the loop architecture. *Appl. Sci.* 11(3), 1154 (2021)
 39. Hilal, R.: Commande tolérante aux défauts du robot manipulateur basé sur flou type 2 adaptatif backstepping en présence de la variation du charge utile. *Int. J. Intell. Eng. Syst.* 14(4), 1–14 (2021)
 40. Merheb, A.R.: Nonlinear control algorithms applied to 3 D.O.F PUMA manipulator. Master's Thesis, Middle East Technical University (2008)
 41. Fekik, A., Denoun, H., Azar, A.T., Kamal, N.A., Zaouia, M., Benyahia, N., et al.: Direct power control of three-phase PWM-rectifier with backstepping control. In: *Backstepping Control of Nonlinear Dynamical Systems*, pp. 215–234. Elsevier, New York (2021)
 42. Fekik, A., Azar, A.T., Denoun, H., Kamal, N.A., Hamida, M.L., Kais, D., et al.: A backstepping direct power control of three phase pulse width modulated rectifier. In: *Soft Computing Applications: Proceedings of the 8th International Workshop Soft Computing Applications (SOFA 2018)*, vol. 2, pp. 445–456. Springer, Cham (2021)
 43. Vaidyanathan, S., Azar, A.T., Rajagopal, K., Sambas, A., Kacar, S., Cavusoglu, U.: A new hyperchaotic temperature fluctuations model, its circuit simulation, FPGA implementation and an application to image encryption. *Int. J. Simul. Process. Model.* 13(3), 281–296 (2018)

How to cite this article: Fekik, A., Khati, H., Azar, A.T., Hamida, M.L., Denoun, H., Hameed, I.A., Kamal, N.A.: FPGA in the loop implementation of the PUMA 560 robot based on backstepping control. *IET Control Theory Appl.* 1–15 (2023). <https://doi.org/10.1049/cth2.12589>