International Journal of Automation and Control

ISSN online: 1740-7524 - ISSN print: 1740-7516
https://www.inderscience.com/ijaac

FPGA-based performance evaluation of backstepping control and computed torque control for industrial robots

Arezki Fekik, Hocine Khati, Ahmad Taher Azar, Mohamed Lamine Hamida, Hakim Denoun, Nashwa Ahmad Kamal

DOI: 10.1504/IJAAC.2025.10062960

Article History:

Received:	29 December 2023
Last revised:	06 February 2024
Accepted:	08 February 2024
Published online:	02 December 2024

FPGA-based performance evaluation of backstepping control and computed torque control for industrial robots

Arezki Fekik*

Department of Electrical Engineering,
University Akli Mohand Oulhadj-Bouria,
Rue Drissi Yahia, Bouira – 10000, Algeria
and
Nantes Université,
École Centrale Nantes,
CNRS,
LS2N, UMR 6004,
F-44000 Nantes, France
Email: a.fekik@univ-bouira.dz
Email: Arezki.Fekik@univ-nantes.fr
*Corresponding author

Hocine Khati

Design and Drive of Production Systems Laboratory,
Department of Automation,
Faculty of Electrical and Computing Engineering,
University Mouloud Mammeri of Tizi-Ouzou,
Tizi-Ouzou, Algeria
Email: hocine.khati@ummto.dz

Ahmad Taher Azar

College of Computer and Information Sciences,
Prince Sultan University,
Riyadh, 11586, Saudi Arabia
and
Automated Systems and Soft Computing Lab (ASSCL),
Prince Sultan University,
Riyadh, Saudi Arabia
Email: aazar@psu.edu.sa
and
Faculty of Computers and Artificial Intelligence,
Benha University,
Benha, 13518, Egypt
Email: ahmad.azar@fci.bu.edu.eg
Email: ahmadtazar@ieee.org

Mohamed Lamine Hamida and Hakim Denoun

Electrical Engineering Advanced Technology Laboratory (LATAGE),
Tizi-Ouzou, Algeria
Email: ml_hamida@yahoo.com
Email: akim_danoun2002dz@yahoo.fr

Nashwa Ahmad Kamal

Faculty of Engineering,
Cairo University,
Giza, 12613, Egypt
Email: nashwa.ahmad.kamal@gmail.com

Abstract: In this paper, a comparative study is conducted on two nonlinear control techniques: state feedback control through backstepping and computed torque control. The study focuses on their application to the industrial robot PUMA 560. The primary goal is to assess the trajectory tracking accuracy and speed achieved by these methods. To achieve this objective, both control techniques are employed on the Zed board Zynq FPGA platform, encompassing both simulation and hardware systems. Subsequently, the experimental results are thoroughly analysed and compared, aiming to accentuate the unique advantages and constraints associated with each method.

Keywords: field-programmable gate array; FPGA; backstepping control; computed torque control; CTC; Zed board Zynq; PUMA 560.

Reference to this paper should be made as follows: Fekik, A., Khati, H., Azar, A.T., Hamida, M.L., Denoun, H. and Kamal, N.A. (2025) 'FPGA-based performance evaluation of backstepping control and computed torque control for industrial robots', *Int. J. Automation and Control*, Vol. 19, No. 1, pp.101–132.

Biographical notes: Arezki Fekik is a Senior Lecturer at the Akli Mohand Oulhadj University-Bouira, Algeria. He is a member of the International Group of System Control (IGCS), a member of the Springer conference committee and a member of the IEEE SMARTTECH conference. His current research interests include power electronics and its applications such as wind turbines, photovoltaic systems, reliability, harmonics, microgrids and variable speed drives. Currently, he performs the functions of design engineer supporting experimental tests and research in power systems control at the Laboratoire des Sciences Numérique de Nantes (LS2N) De l'école Centrale de Nantes. He has published more than 65 journal and conference articles and book chapters in the fields of power electronics and its applications.

Hocine Khati obtained his Bachelor's and Master's in Automatic Control from the Mouloud Mammeri University of Tizi-Ouzou (UMMTO) in Algeria in 2013 and 2015 respectively. In 2020, he obtained his Dotoral diploma in Automatics at the same institution. He is currently a teacher-researcher in the Automation Department at the UMMTO. His research focuses on numerical control, intelligent control and robotics.

Ahmad Taher Azar is a Full Professor at the Prince Sultan University, Riyadh, Saudi Arabia. He is also a Full Professor at the Faculty of Computers and Artificial Intelligence, Benha University, Egypt. He is also a leader of Automated Systems and Soft Computing Lab (ASSCL), Prince Sultan University, Saudi Arabia. He is currently an an editor for *IEEE Systems Journal*, *IEEE Transactions on Neural Networks and Learning Systems*, Springer's *Human-centric Computing and Information Sciences*, and Elsevier's *Engineering Applications of Artificial Intelligence*. He has expertise in artificial intelligence, control theory and applications, robotics, machine learning, computational intelligence and dynamic system modelling. He has authored/co-authored over 450 research papers in prestigious peer-reviewed journals, book chapters, and conference proceedings.

Mohamed Lamine Hamida received his BSc, MSc and PhD in Electrical Engineering from the Mouloud Mammeri University, Tizi-Ouzou, Algeria, in 20 3, 2015 and 2019, respectively. He is a member of the Electrical Engineering Advanced Technology Laboratory (LATAGE) since 2015 and he became a Lecturer in 2020 at the same university. His current research interests include power electronics and its applications such as the application of PWM, fuzzy logic control, sliding mode control and Petri Nets modelling to DC/D and DC/AC multi-cell converters. He has published many journal and conference articles in the field of power electronics and its applications.

Hakim Denoun received his BSc in Electrical Engineering from the Mouloud Mammeri University, Algeria, and DEA from Paris 6, France and the Magister from the Polytechnic School, Algiers, Algeria. He received his PhD in Electrical Engineering from the Mouloud Mammeri University, and currently a Professor at the same university. His research interests include electrical machines and drives, power electronics and control systems.

Nashwa Ahmad Kamal received her MSc in 2014 and PhD in 2019 from the Faculty of Engineering, Cairo University, Egypt. She is a managing editor of some international journals like *International Journal of System Dynamics Applications*, *International Journal of Service Science, Management, Engineering, and Technology* and *International Journal of Sociotechnology and Knowledge Development* published by IGI Global, USA. She also worked in the areas of control theory and applications, process control, nonlinear control, robotics and renewable energy and has authored/coauthored some research publications in peer-reviewed reputed journals book chapters and conference proceedings. She is a member of international group of control systems and International Society of Automation (ISA), USA.

This paper is a revised and expanded version of a paper entitled [title] presented at [name, location and date of conference].

1 Introduction

The rapid expansion of industrial robots across a wide array of industrial applications has underscored the need for effective control techniques to optimise their performance

(Gadaleta et al., 2009; Luan et al., 2022; Maraveas et al., 2023; Jain et al., 2023; Wang et al., 2021; Hanna et al., 2021; Khamis et al., 2021; Ibraheem et al., 2020). Among these robots, the PUMA 560 has emerged as a prominent choice in the industry, primarily due to its streamlined mechanical structure and remarkable flexibility (Taylor et al., 2022; Marcus et al., 2013; Jokić et al., 2022; Ibrahim, 1996). As a result, extensive research efforts are currently underway to enhance the control systems governing these robots, aiming to address the diverse challenges encountered in real-world applications (Najm et al., 2020).

Efficient control of industrial robots plays a critical role in achieving precise and reliable operation, maximising productivi (Moreno-Valenzuela et al., 2022; Salinashh et al., 2016). By regulating the movements and actions of these robots, control systems contribute significantly to the successful execution of various tasks in industries such as manufacturing, logistics, and healthcare. With the continuous advancement of robotic technology, there is an increasing demand for control methodologies capable of handling the inherent complexities and nonlinearities associated with robotic systems. Researchers and engineers are actively exploring innovative control methods that can offer stability, robustness, and adaptability to meet the dynamic requirements of industrial environments.

Formally, the motivation for this research stems from the identified gap in the existing literature. There is a lack of a comprehensive comparative analysis of two well-established nonlinear control methods – backstepping state feedback control and computed torque control (CTC) – applied to the PUMA 560 robot. This research aims to bridge this gap and contribute valuable insights to the field of industrial robot control by conducting a detailed assessment of trajectory tracking accuracy and speed achieved by each control method.

In this study, our primary objective is to compare the performance of two well-established nonlinear control methods: backstepping state feedback control and CTC, with a specific focus on their application to the PUMA 560 robot (Merheb, 2008; Slotine and Li, 1991; Spong et al., 2020). To accomplish this, we implemented both control methods on a simulation system and a hardware system utilising the Zed board Zynq FPGA platform. Through comprehensive evaluations, we assessed the trajectory tracking accuracy and speed achieved by each control method, shedding light on their individual strengths and limitations. To provide a solid foundation for our research, the ‘related work’ section of this paper introduces the fundamental concepts of industrial robotics and industrial robot control. We explore the underlying principles of backstepping state feedback control and CTC, highlighting their significance and applicability in the context of industrial robot control. In the subsequent section, ‘design and implementation of the two control methods’, we delve into the intricacies of designing and implementing each control method. We outline the specific steps and considerations involved in deploying the backstepping state feedback control and CTC approaches, elucidating the technical aspects of their implementation. The ‘experimental results and discussion’ section presents the outcomes of our experiments conducted with both control methods. We meticulously analyse and compare the experimental results, examining the trajectory tracking accuracy, responsiveness, and other relevant performance metrics. This empirical assessment aims to provide valuable insights into the relative merits of the two control methods in controlling the PUMA 560 robot. Finally, the ‘conclusions’ section summarises the findings obtained from our study, emphasising the key takeaways and implications of the research. We also acknowledge

the limitations of our study and present future prospects for further advancements in the field of industrial robot control. By contributing to the ongoing discourse on control methodologies for industrial robots, we aspire to foster continued innovation and progress in this pivotal domain.

2 Related work

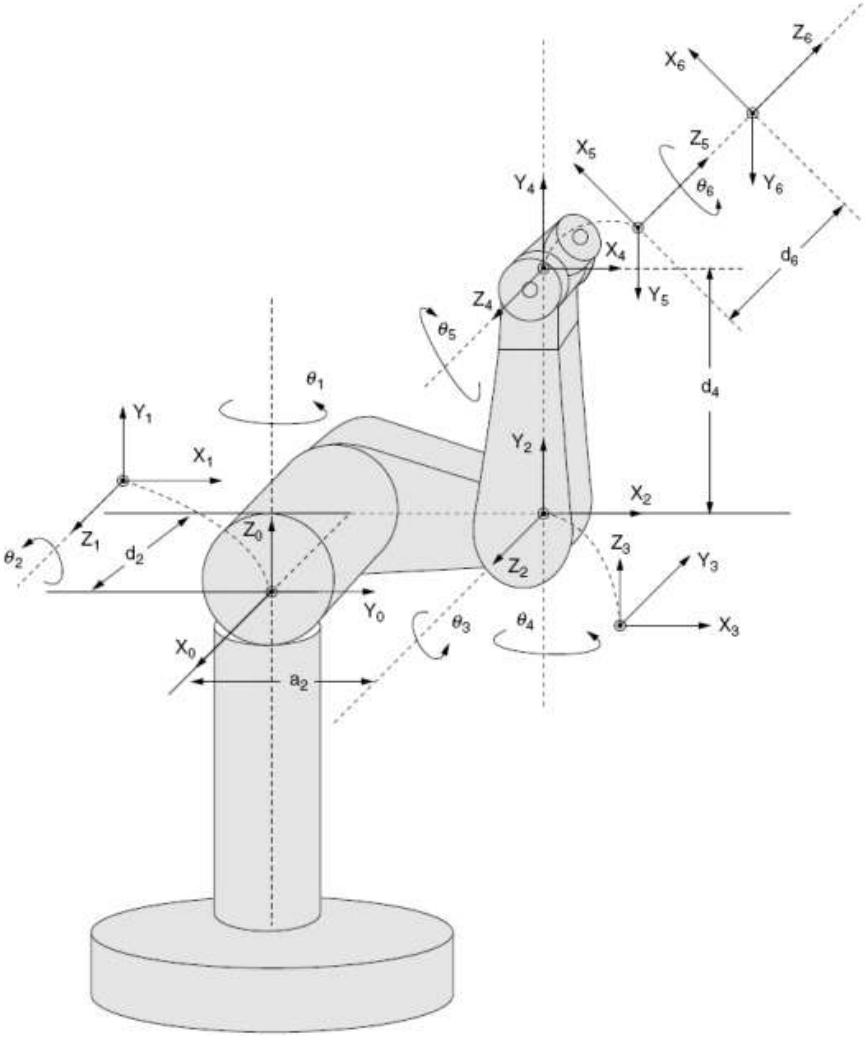
Industrial robotics is a field of robotics that aims to design and develop robots for industrial applications. Industrial robots are automated mechanical systems capable of performing repetitive and monotonous tasks faster, more accurately, and more efficiently than humans (Soliman et al., 2020; Khan, 2020; Azar and Serrano, 2015; Mohamed et al., 2023). The PUMA 560 robot is a six-axis industrial robot designed by Unimation in the 1980s. It is widely used in industries for applications such as welding, assembly, painting, and material handling (Bhattacharya et al., 2020; Lewis et al., 2003). Industrial robot control is a field of robotics that focuses on designing control algorithms for industrial robots. Control algorithms are used to define the robot's trajectory and control its movements to reach the reference position (Wang et al., 2024; Hady et al., 2023; Larbi et al., 2023; Zhu et al., 2022; Rani and Kumar, 2022). Robot control algorithms can be classified into two categories: position control algorithms and force control algorithms. Position control algorithms are used to control the robot's position, while force control algorithms are used to control the force exerted by the robot (Siciliano et al., 2009). Backstepping state feedback control is a nonlinear control method that has been extensively studied in the literature. It enables stable and robust control of nonlinear systems. The backstepping state feedback control method involves dividing the system into multiple subsystems, each controlled by a specific control function (Spong et al., 2020). The backstepping state feedback control method has been widely used in industrial robot control due to its robustness and stability. It has been successfully applied to the control of industrial robots on numerous occasions (Van et al., 2018; Anjum et al., 2021; Rahali et al., 2021; Meng et al., 2021). CTC is another nonlinear control method that is also used in industrial robot control. The CTC method involves calculating the force and torque required to achieve the reference trajectory and controlling the robot based on these values (Chen et al., 2012; Shang et al., 2012; Chen et al., 2013). The CTC method has also been widely used in industrial robot control. It has been successfully applied to the control of industrial robots on multiple occasions. Industrial robot control is an active research area aimed at improving the performance of these systems in various applications. Backstepping state feedback control and CTC are two widely used nonlinear control methods in industrial robot control, and they have been successfully applied to the control of the PUMA 560 robot. The objective of this work is to implement both control methods on a Zed board Zynq FPGA using the HDL WorkflowAdvisor functionality through the FPGA in-the-loop option, and to compare the performance of the two techniques.

3 Dynamic modelling of the PUMA 560 robot

This research focuses on the advancement of a dynamic model to control the PUMA 560 industrial manipulator robot, as depicted in Figure 1. The dynamic equations of

motion represent a collection of mathematical equations that characterise the dynamic characteristics of the robot arm (Ajeil et al., 2020). These equations play a crucial role in simulating the arm's motion and devising efficient control strategies (Fekik et al., 2023; Ahmed et al., 2022; Singh et al., 2022).

Figure 1 The 6 degree-of-freedom (6DOF) PUMA 560 robot



Source: Merheb (2008)

Various formulations exist for deriving the dynamic model of a manipulator, including the Lagrange-Euler, Newton-Euler, and d'Alembert equations. Among these, the Lagrange-Euler formulation is commonly used due to its simplicity and systematic approach, as it is based on an 'energy-based' principle. In this formulation, the manipulator is treated as a serial chain of rigid links. The Lagrange-Euler equation establishes a relationship between the torque exerted at joint i, which drives link i of the

manipulator, and the system's kinetic and potential energies, as well as its generalised coordinates (Merheb, 2008)

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

where

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

The Lagrange-Euler equation is utilised to determine the torque vector that is applied to the joints of each link within the manipulator. By considering the kinetic and potential energies, as well as the generalised coordinates of the system, this equation provides the overall form of the dynamic model of the manipulator (Merheb, 2008).

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

with

- q: $n \times 1$ position vector
- $M(q)$: $n \times n$ 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 redefining the term $V(q,\dot{q})$ to be solely dependent on the manipulator's position, the matrices involved in the dynamic equation are expressed as functions of the manipulator position alone.

This formulation is known as the configuration space equation, and it can be written as follows (Merheb, 2008):

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

- $B(q)$: $n \times n(n-1)/2$ matrix of Coriolis torques
- $C(q)$: $n \times n$ matrix of Centrifugal torques
- $[\dot{q}.\dot{q}]$: $n(n-1)/2 \times 1$ vector of joint velocity products given by:

$$\left[\dot{q}_1.\dot{q}_2, \dot{q}_1.\dot{l}\dot{q}_3, \ldots, \dot{q}_1.\dot{q}_n, \dot{q}_2.\dot{q}_3, \dot{q}_2.\dot{q}_4, \ldots, \dot{q}_{n-2}.\dot{q}_n, \dot{q}_{n-\mathbf{1}}.\dot{q}_n\right]^T$$

- $\left[\dot{q}^2\right]$: $n \times 1$ vector given by: $\left[\dot{q}_1^2, \dot{q}_2^2, \ldots, \dot{q}_n^2\right]$.

It is important to mention that this thesis focuses exclusively on three links of the PUMA robot, where the joint angles q_4, q_5, and q_6 are set to zero. Consequently, the dynamic equation can be simplified and expressed in the configuration space representation (Merheb, 2008).

$$\Gamma = A\left(q\right)\ddot{q} + B\left(q\right).\left[\dot{q}.\dot{q}\right] + C\left(q\right).\left[\dot{q}^2\right] + G(q) \tag{4}$$

with, matrix A is a symmetric 6x6 matrix represents the inertia terms of the robot (Merheb, 2008):

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

with

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

$$\begin{aligned} a_{12} &= I_4.S_2 + I_8.C_{23} + I_9.C_2 + I_{13}.S_{23} + I_{18}.C_{23} \\ a_{13} &= I_8.C_{23} + I_{13}.S_{23} - I_{18}.C_{23} \\ a_{22} &= I_{m2} + I_2 + I_6 + 2[I_5.S_3 + I_{12}.C_2 + I_{15} + I_{16}.S_3] \\ a_{23} &= I_5.S_3 + I_6 + I_{12}.C_3 + I_{16}.S_3 + 2I_{15} \\ a_{33} &= I_{m3} + I_6 + 2I_{15} \\ a_{35} &= I_{15} + I_{17} \\ a_{44} &= I_{m4} + I_{14} \\ a_{55} &= I_{m5} + I_{17} \\ a_{66} &= I_{m6} + I_{23} \\ a_{21} &= a_{12},\ a_{31} = a_{13} \text{ and } a_{32} = a_{23} \end{aligned}$$

Matrix B represents the coriolis and centripetal terms of the robot (Merheb, 2008):

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

where

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

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

$$b_{115} = 2[-SC_{23} + I_{15}.SC_{23} + I_{16}.C_2.C_{23} + I_{22}.CC_{23}]$$
$$b_{123} = 2[-I_8.S_{23} + I_{13}.C_{23} + I_{18}.S_{23}]$$
$$b_{214} = I_{14}.S_{23} + I_{19}.S_{23} + 2.I_{20}.S_{23}.(1 - 0.5)$$
$$b_{223} = 2[-I_{12}.S_3 + I_5.C_3 + I_{16}.C_3]$$
$$b_{225} = 2[I_{16}.C_3 + I_{22}]$$
$$b_{314} = 2[I_{20}.S_{23}.(1 - 0.5)] + I_{14}.S_{23} + I_{19}.S_{23}$$
$$b_{412} = -b_{214} = -[I_{14}.S_{23} + I_{19}.S_{23} + 2I_{20}.S_{23}.(1 - 0.5)]$$
$$b_{413} = -b_{314} = -2[I_{20}.S_{23}.(1 - 0.5)] + I_{14}.S_{23} + I_{19}.S_{23}$$
$$b_{415} = -I_{20}.S_{23} - I_{17}.S_{23}$$
$$b_{514} = -b_{415} = I_{20}.S_{23} + I_{17}.S_{23}$$

Matrix C represents the viscous friction terms of the robot (Merheb, 2008):

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

where

$$C_{12}=I_4.C_2-I_8.S_{23}-I_9.S_2+I_{13}.C_{23}+I_{18}.S_{23}$$
$$C_{13}= 0.5b_{123}= -I_8.S_{23}+I_{13}.C_{23}+I_{18}.S_{23}$$

$$C_{21}=0.5b_{112}=I_3.SC_2-I_5.C_{223}-I_7.SC_{23}+I_{12}.S_{223}-I_{15}.2.SC_{23} -I_{16}.C_{223}-I_{21}.SC_{23}-I_{22}.(1 - 2.SS_{23}) -0.5I_{10}.(1 - 2SS_{23})-0.5I_{11}.(1 - 2.SS_2)$$

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

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

$$C_{32}= -C_{23}=I_{12}.S_3-I_5.C_3-I_{16}.C_3$$
$$C_{51}= 0.5b_{115}=SC_{23}-I_{15}.SC_{23}-I_{16}.C_2.C_{23}-I_{22}.CC_{23}$$
$$C_{52}= -0.5b_{225}= -I_{16}.C_3-I_{22}$$

Matrix G is (Merheb, 2008):

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

with

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

and

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

4 Backstepping control

4.1 Concept

The backstepping control technique is a recursive design approach used to develop feedback control laws and systematically choose associated Lyapunov functions (Vaidyanathan et al., 2018). The fundamental idea behind backstepping is to treat certain states as 'pseudo-controls' for other states and utilise Lyapunov functions to ensure the stability of these controlled states. In the implementation of backstepping control, an artificial control law is assigned to the first state, and a Lyapunov candidate function for control is defined. The expression for the artificial control is then selected to guarantee a negative derivative of the Lyapunov function, thus ensuring system stability. This process is repeated by assigning artificial control laws and selecting Lyapunov functions for subsequent states until all states of the system are addressed. Ultimately, the designer achieves guaranteed asymptotic stability of the system (Fekik et al., 2016, 2021).

Let's consider a system in the state space form that is feedback linearisable, represented by the following equation (Merheb, 2008):

$$\ddot{x}=f(x)+g(x).u \tag{9}$$

Assuming that the system can be stabilised using a continuous state feedback control law (Merheb, 2008):

$$u=\alpha(x) \tag{10}$$

e_1 can be defined as the displacement error, which is presented by the following equation (Merheb, 2008):

$$e_1 = y_{reference} - y_{mesured} = x_{reference} - x_{mesured} \tag{11}$$

Driven by the assumption of system stability, a control Lyapunov function (CLF), denoted as $V(x)$, can be identified, where its derivative is negatively definite. The selected candidate Lyapunov function satisfies the specified condition (Merheb, 2008):

$$\dot{V}=\frac{dV}{dx}\left(f\left(x\right)+\alpha.g(x)\right)\leq -W(x)\leq 0 \tag{12}$$

Choose the first CLF as indicate in the equation (Merheb, 2008):

$$V_1=\frac{1}{2}.e_1^2 \tag{13}$$

The derivation of the first CLF is given by (Merheb, 2008):

$$\dot{V_1}=e_1.\dot{e_1}=e_1.[x_2-\dot{y_d}] \tag{14}$$

To ensure stability of the system, it is necessary for $\dot{V_1}$ value to have a negative (Merheb, 2008):

$$[x_2]_{reference}=k_1.e_1.x_2+\dot{y_d} \tag{15}$$

The virtual control, represented by α_1, is a suitable option, let's take a step back for the second state (Merheb, 2008):

$$e_1=x_{2mesured}-x_{2reference}=x_{2mesured}-\alpha_1=x_{2mesured}+k_1.e_1-\dot{y_d} \tag{16}$$

We can introduce an extended of CLF (Merheb, 2008):

$$V_2=\frac{1}{2}.e_1^2+\frac{1}{2}.e_2^2 \tag{17}$$

With $e_1=x_2-\dot{y_d}=e_2-k_1.e_1$.

The derivative of the extended of CLF becomes (Merheb, 2008):

$$\dot{V_2}=-k_1e_1^2+e_2.\left[e_1+\dot{x_2}-\alpha_1\right]=-k_1e_1^2+e_2.\left[\left(1-k_1^2\right)e_1+k_1e_2+x_3-\ddot{y}_{desired}\right. \tag{18}$$

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 (Merheb, 2008):

$$\alpha_2=(x_3)_d=\left(k_1^2-1\right).e_1-\left(k_1+k_2\right).e_2+\ddot{y}_{reference}=\ddot{x}=f\left(x\right)+g\left(x\right).u \tag{19}$$

And *u* is found as to get (Merheb, 2008):

$$g^{-1}\left(x\right).\left[\left(k_1^2-1\right).e_1-\left(k_1+k_2\right).e_2+\ddot{y}_{reference}-f\left(x\right)\right] \tag{20}$$

4.2 Backstepping control of three-link PUMA 560 Robot

The equation for the robot's is configuration space (Merheb, 2008):

$$M(q)\ddot{q} + V(q,\dot{q})\dot{q} + G(q) = \Gamma \tag{21}$$

We can define a new variable, $x_1 = q => \dot{x_1} = \dot{q} = x_2$, which implies that $\dot{x_2} = \ddot{q}$, and set $x_1 = y$.

Therefore, the system can be expressed in the following form (Merheb, 2008):

$$\dot{x}_1 = x_2 \tag{22}$$

$$\dot{x}_2 = A^{-1}(q).[\Gamma - B(q).(\dot{q}\dot{q}) - C(q).(\dot{q}^2) - G(q)] \tag{23}$$

$$y = x_1 = h(x) \tag{24}$$

With $e_1 = q_d - q => \dot{e_1} = \dot{q_d} - \dot{q} = \dot{e_2}$ where g_d is the reference joint position vector, and q is the actual joint position.

The first CLF is given by (Merheb, 2008):

$$V_1 = \frac{1}{2}.e_1^2 \tag{25}$$

The derivation of the first CLF is (Merheb, 2008):

$$\dot{V_1} = e_1.\dot{e_1} = e_1.[x_2 - \dot{q_d}] \tag{26}$$

To ensure system stability, x_2 is selected as the virtual control input (Merheb, 2008):

$$[x_2]_{reference} = -k_1.e_1 + \dot{q_d} \tag{27}$$

The control law of the speed using the backstepping control is given by (Merheb, 2008):

$$e_1 = x_{2mesured} - x_{2reference} = x_{2mesured} + k_1.e_1 - \dot{q_d} \tag{28}$$

The extended of CLF is given by the following equation (Merheb, 2008):

$$V_2 = \frac{1}{2}.e_1^2 + \frac{1}{2}.e_2^2 \tag{29}$$

The derivative of the extended of CLF becomes as show in (Merheb, 2008):

$$\begin{aligned}\dot{V_2} &= -k_1 e_1^2 + e_2.[e_1 + \dot{x_2} - \alpha_1] \\ &= -k_e^{2_1} + e_2.[(1 - k^{2_1})]e_1 + k_1 e_2 + \dot{x}_2 - \ddot{q}_d\end{aligned} \tag{30}$$

By selecting $\dot{x}_2$ as the second virtual control and setting its value to make $\dot{V_2}$ negative definite, we obtain (Merheb, 2008):

$$\begin{aligned}\alpha_2 = \dot{x}_2 = \ddot{q} &= (k_1^2 - 1).e_1 - (k_1 + k_2).e_2 \\ &+ \ddot{q}_d = A^{-1}(q).\Gamma - B(q).(\dot{q}\dot{q}) - C(q).(\dot{q}^2) - G(q)\end{aligned} \tag{31}$$

Solving for the Torque we find (Merheb, 2008):

$$\begin{aligned}\Gamma = A(q).[(k_1^2 - 1).e_1 - (k_1 + k_2).e_2 + \ddot{q}_d] + B(q).(\dot{q}\dot{q}) \\ + C(q).(\dot{q}^2) + G(q)\end{aligned} \tag{32}$$

5 Calculated torque control

5.1 Concept

Calculated torque control is an advanced control approach employed in multi-joint manipulator control (Mahfoud et al., 2022; El Ouanjli et al., 2022; Ramu et al., 2020). This method combines both feedforward and feedback control elements to enhance the accuracy and performance of the robot. Its principle is based on utilising an inverse model of the robot to compensate for the nonlinearities in its dynamics, thereby reducing deviations from the reference trajectory. Concurrently, a proportional-derivative (P-D) controller is used to regulate the joint motors, ensuring precise and smooth movement of the manipulator. This approach represents a cutting-edge method for optimising the performance of multi-joint manipulators.

The first component of the control law relies on a model that simplifies the system to a unit mass system: $\Gamma = \alpha\Gamma' + \beta$. Here, α and β are functions or constants chosen such that Γ' becomes the new input to the system. This first part of the control law contributes to establishing a simplified representation of the system (Merheb, 2008). The second part of the control law corresponds to the P-D controller, also known as the 'servo part'. This controller utilises both proportional (P) and derivative (D) actions to adjust and regulate the system. The proportional action takes into account the difference between the reference value and the actual value, while the derivative action evaluates the rate of change of this difference over time. Together, these two components enable precise and rapid response of the system, thus contributing to its servoing (Merheb, 2008).

$$\Gamma = A(q)\,\ddot{q} + B(q).[\dot{q}.\dot{q}] + C(q).\left[\dot{q}^2\right] + G(q) = \alpha\Gamma' + \beta \tag{33}$$

By selecting $\alpha = A(q)$ and $\beta = B(q).[\dot{q}\ddot{q}] + C(q)[\dot{q}\wedge 2] + G(q)$ and $\Gamma' = \ddot{q}$.

The servo component consists of a P-D controller, where (Merheb, 2008):

$$\Gamma' = \ddot{q}_d + k_v\dot{e} + k_{pe} \tag{34}$$

with $e = q_d - q \Rightarrow \dot{e} = \dot{q}_d - \dot{q}$.

In this equation, q_d represents the reference joint position vector, q represents the actual joint position, and K_v and K_p are positive-definite matrix gains for the derivative and proportional terms, respectively.

The control law can be expressed as (Merheb, 2008):

$$\Gamma = A(q).[\ddot{q}_d + k_v\dot{e} + k_p e] + B(q).[\dot{q}\dot{q}] + C(q).\left[\dot{q}^2\right] + G(q) \tag{35}$$

Applying this control law to regulate the manipulator (Merheb, 2008):

$$\begin{aligned}\Gamma &= A(q).[\ddot{q}_d + k_v\dot{e} + k_p e] + B(q).[\dot{q}\dot{q}] \\ &\quad + C(q).\left[\dot{q}^2\right] + G(q) \\ &= A(q).\ddot{q} + B(q).[\dot{q}\dot{q}] + C(q).\left[\dot{q}^2\right] \\ &\quad + G(q) \\ &= A(q).[\ddot{e}(t) + k_v\dot{e}(t) + k_p e(t)] = 0\end{aligned} \tag{36}$$

By appropriately selecting K_v and K_p, such that the characteristic roots of the controller function have negative real parts, the position error vector can asymptotically approach zero. This scenario is referred to as the critically damped case, where errors are efficiently suppressed in a second-order linear manner without causing any overshoot. It is worth noting that $A(q)$ is always nonsingular in this context.

The characteristic equation of the system can be expressed as follows (Merheb, 2008):

$$S^2 + K_v S + K_p = 0 \tag{37}$$

The roots of a second-order equation are given

$$\begin{aligned} S_1 &= -\frac{K_v}{2} - \frac{\sqrt{K_v^2 - 4K_p}}{2} \\ S_2 &= -\frac{K_v}{2} + \frac{\sqrt{K_v^2 - 4K_p}}{2} \end{aligned} \tag{38}$$

In the aforementioned approximation, it is assumed that the dynamics of the controlled system are completely known. However, in situations where the knowledge about the system is not exact, estimated values of the system parameters are utilised. It is important to note that in such cases, the feedback component of the controller will no longer perfectly cancel out the nonlinearities of the system (Merheb, 2008).

$$\begin{aligned} \Gamma &= A(q) \cdot [\ddot{q}_d + k_v \dot{e} + k_p e] + B(q) \cdot [\dot{q}\dot{q}] + C(q) \cdot \left[\dot{q}^2\right] + G(q) \\ &= A(q) \cdot \ddot{q} + B(q) \cdot [\dot{q}\dot{q}] + C(q) \cdot \left[\dot{q}^2\right] + G(q) \end{aligned} \tag{39}$$

The estimated values of the robot dynamics, denoted as $A_a(q)$, $B_a(q) \cdot [\dot{q}\dot{q}]$, $C_a(q)$, and $G_a(q)$ are used when the knowledge about the system is not exact. As the estimations become more accurate, they help in canceling out the nonlinear effects caused by gravity, manipulator inertia tensor, and the Coriolis and Centrifugal forces of the robot. These effects are treated as disturbances in the single-joint controller.

5.2 CTC of three-link PUMA 560 robot

Applying the CTC method to the configuration space equation of the PUMA robot results in (Merheb, 2008):

$$\begin{aligned} &A(q) \cdot [\ddot{q}_d + k_v \dot{e} + k_p e] + B_a(q) \cdot [\dot{q}\dot{q}] + C_a(q) \cdot \left[\dot{q}^2\right] + G_a(q) \\ &= A(q) \cdot \ddot{q} + B(q) \cdot [\dot{q}\dot{q}] + C(q) \cdot \left[\dot{q}^2\right] + G(q) \Rightarrow \ddot{q} \\ &\times \left(A(q) \cdot [\ddot{e}(t) + k_v \dot{e}(t) + k_p e(t)] + B_a(q) \cdot [\dot{q}\dot{q}] + C_a(q) \cdot \left[\dot{q}^2\right]\right. \\ &\left. + G_a(q) - B(q) \cdot [\dot{q}\dot{q}] - C(q) \cdot \left[\dot{q}^2\right] - G(q)\right) A^{-1} \end{aligned} \tag{40}$$

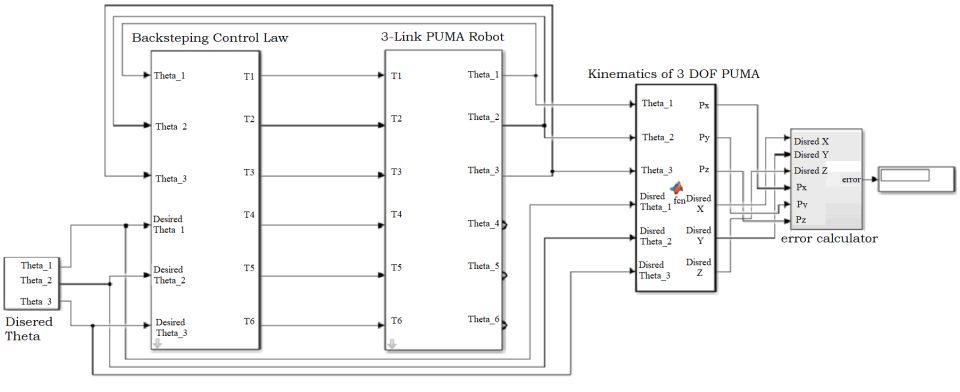
If the estimated values $A_a(q)$, $B_a(q) \cdot [\dot{q}\dot{q}]$, $C_a(q)$, and $G_a(q)$ are equal to the actual values $A(q)$, $B(q)$, $C(q)$, and $G(q)$, respectively, the controlled robot model can be represented as $A(q) \cdot [\ddot{q}_d + k_v \dot{e} + k_p e] = 0$, where $e(t)$ denotes the position error vector (Merheb, 2008).

$$\begin{bmatrix} a_{11}\ a_{12}\ a_{13} \\ a_{21}\ a_{22}\ a_{23} \\ a_{31}\ a_{32}\ a_{33} \end{bmatrix} \cdot \left(\begin{bmatrix} \ddot{e}_1(t) \\ \ddot{e}_2(t) \\ \ddot{e}_3(t) \end{bmatrix} + \begin{bmatrix} K_{v1} & 0 & 0 \\ 0 & K_{v2} & 0 \\ 0 & 0 & K_{v3} \end{bmatrix} \cdot \begin{bmatrix} \dot{e}_1(t) \\ \dot{e}_2(t) \\ \dot{e}_3(t) \end{bmatrix} \right) + \begin{bmatrix} K_{p1} & 0 & 0 \\ 0 & K_{p2} & 0 \\ 0 & 0 & K_{p3} \end{bmatrix} \cdot \begin{bmatrix} e_1(t) \\ e_2(t) \\ e_3(t) \end{bmatrix} = 0 \tag{41}$$

6 Experimental work

This section explains the fundamental steps involved in implementing the Simulink model using the HDL Coder tool, specifically utilising the 'HDL WorkflowAdvisor' functionality and the 'FPGA in-the-loop' option. However, in the cited research works (Mehendale et al., 2022; Qasimh et al., 2009; Ibrohh and Marinova, 2022; Sahu et al., 2017; Pandit and Shet, 2017; Al-Safi et al., 2020), this technique is not used. The process includes developing a Simulink model to represent the system's behaviour, generating VHDL/Verilog code based on the model, configuring the FPGA platform, integrating the generated code into the FPGA platform, simulating the Simulink model on the FPGA platform, and finally deploying the system physically. The 'HDL WorkflowAdvisor' function provides guidance at each step of the implementation process. The following steps should be followed to accomplish this: in this step, a simulation model of the PUMA 560 robot controlled by both the CTC and backstepping techniques will be created using the MATLAB/Simulink software. The model will utilise the double precision floating-point data type for accurate representation and computation. Figure 2 illustrates the structure of the simulation model.

Figure 2 The Simulink model of the PUMA 560 robot (see online version for colours)



Figures 3 and 4 provide a visual representation of the configuration parameters used to solve the equations of the model being studied. These figures also illustrate the parameterisation process involved in generating the hardware description language (HDL) code. By examining these figures, one can gain a better understanding of the specific settings and values used in the configuration and code generation processes.

To configure the FPGA platform, the Xilinx Zed board Zynq XC7z020-1-CLG484 is selected. The board is setup with a sampling period of 0.0001s, as depicted in

Figure 5. In order to match the capabilities of the FPGA, the synthesised model from Simulink needs to be converted from double-precision floating-point to a fixed-point model with a data size of 20 bits. This conversion is facilitated using the 'fixed-point tool'. It is important to note that the Zedboard does not support the double-precision floating-point data type, making the use of fixed-point representation necessary. Additionally, employing fixed-point data types enables the generation of VHDL code that optimises hardware resource utilisation during the implementation of the controller.

Figure 3 Simulink configuration setting (see online version for colours)

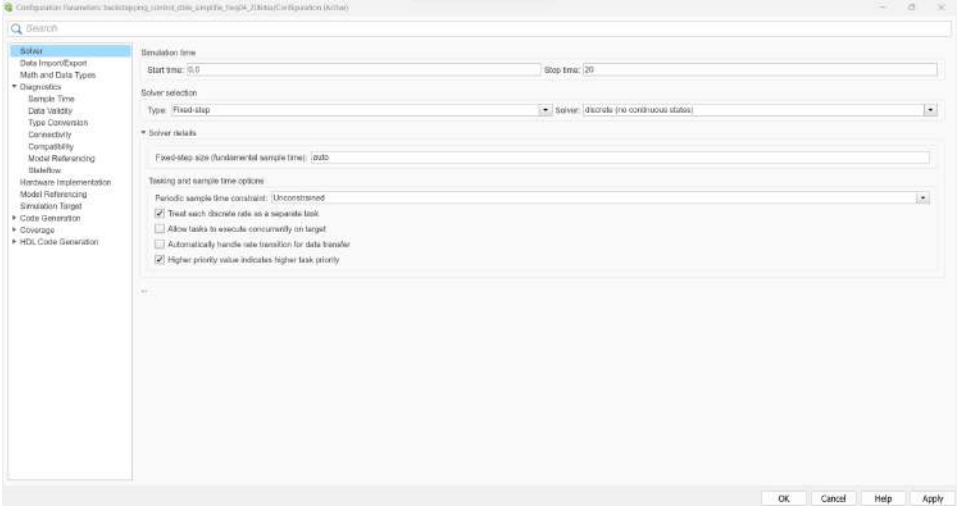
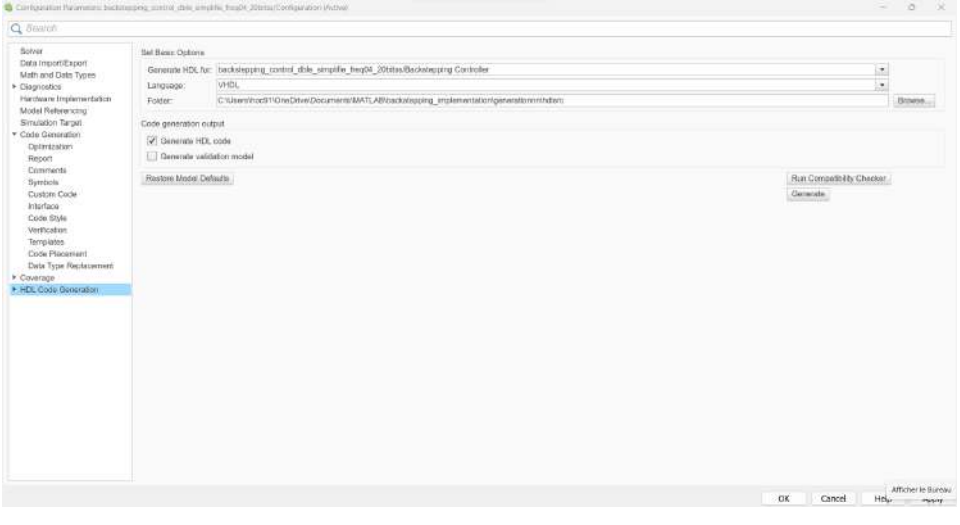


Figure 4 HDL code generation setting (see online version for colours)



To integrate the VHDL code into the FPGA platform, the user needs to incorporate the VHDL code generated by the HDL Coder tool into the FPGA platform. This process

involves compiling the code and performing the necessary tasks of placing and routing the components on the FPGA board. These actions are illustrated in Figure 6.

Figure 5 Configuration of the FPGA Zynq Zed board platform (see online version for colours)

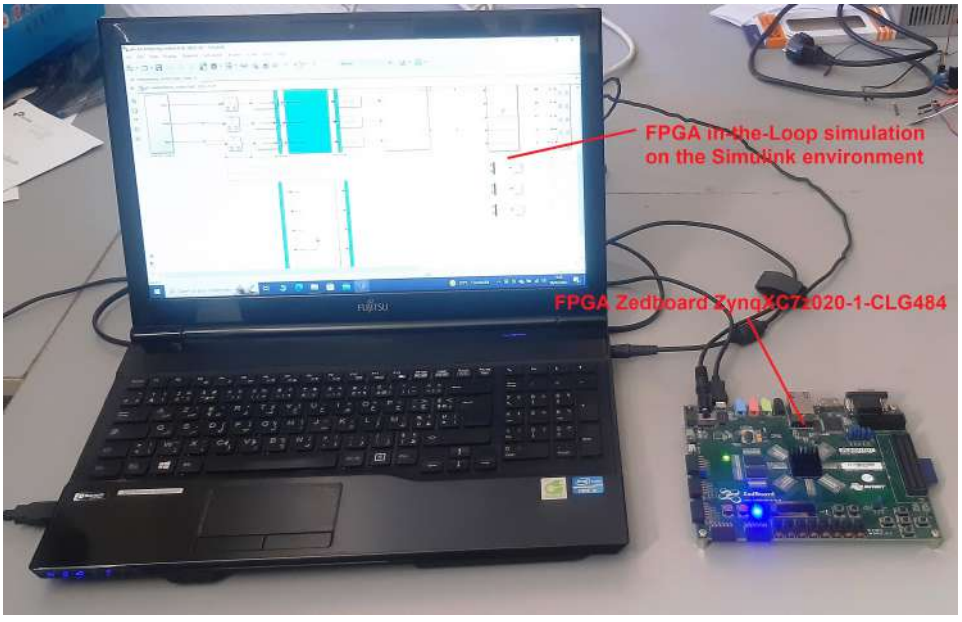


Figure 6 The integration of VHDL code into the FPGA platform (see online version for colours)

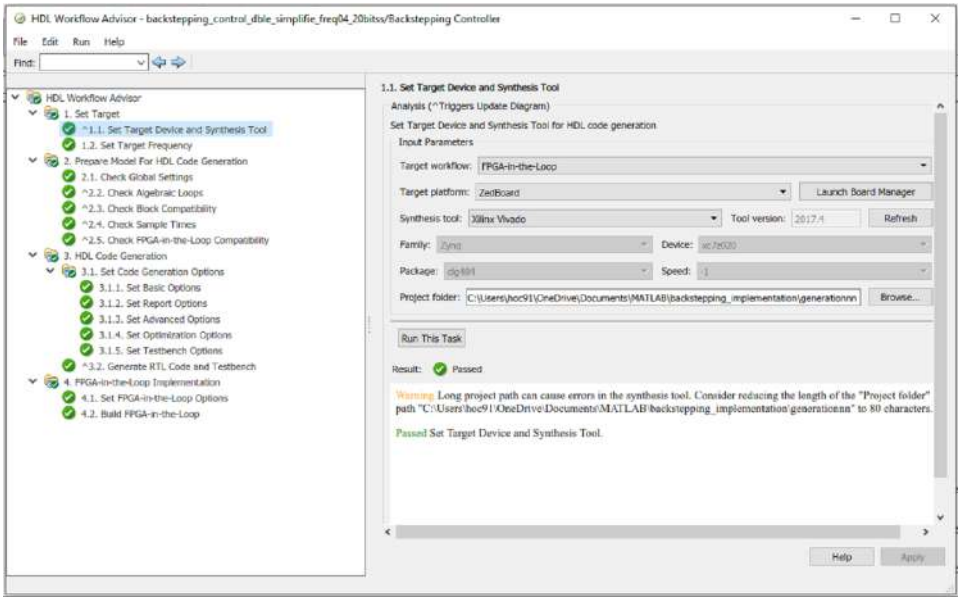


Figure 7 Utilising a backstepping controller for FPGA in-the-loop (see online version for colours)

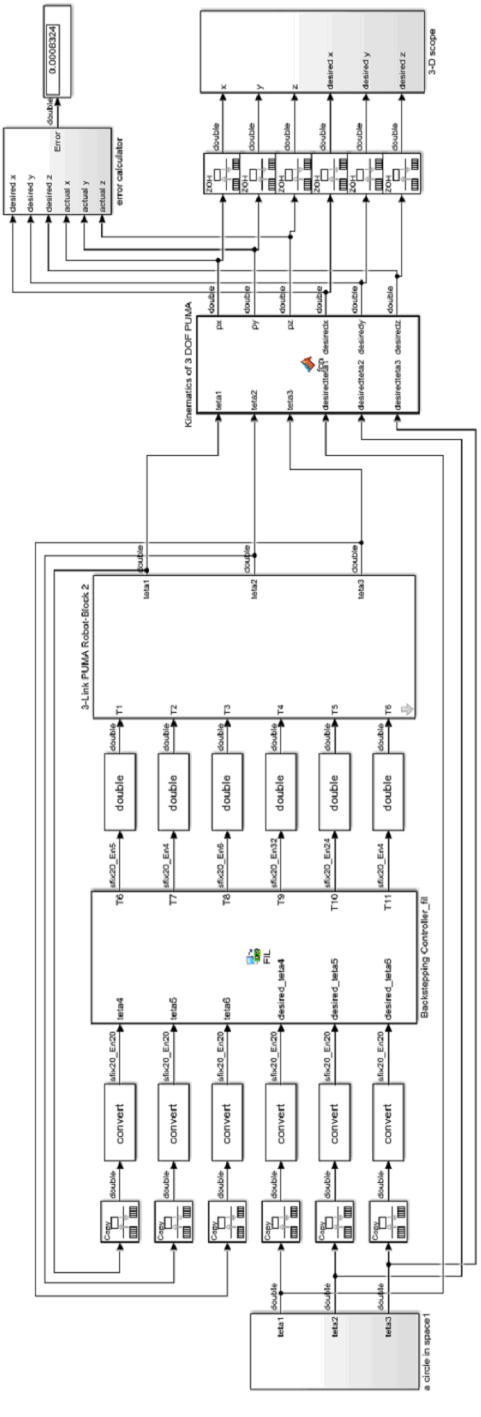


Figure 8 Utilising a CTC for FPGA in-the-loop (see online version for colours)

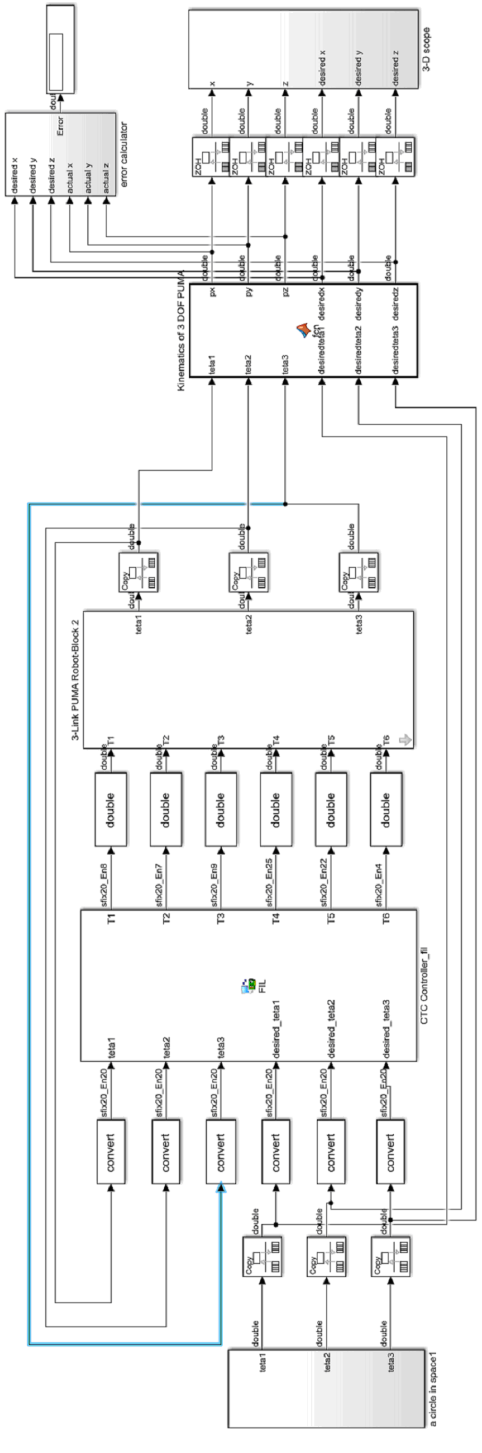


Figure 9 Angular position θ_1 control using a backstepping controller (see online version for colours)

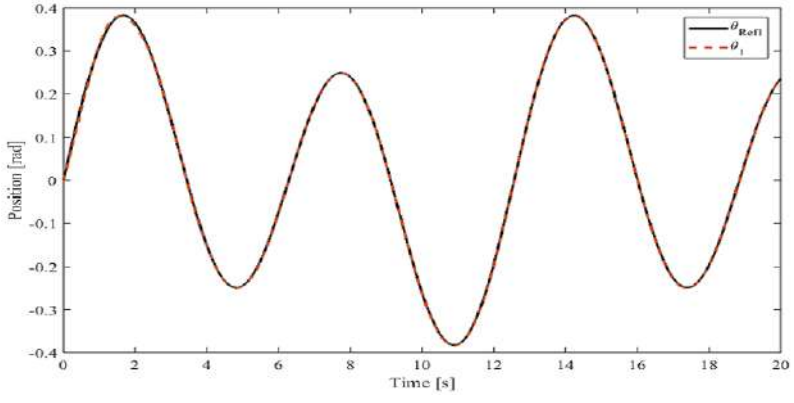


Figure 10 Angular position θ_1 control using a CTC (see online version for colours)

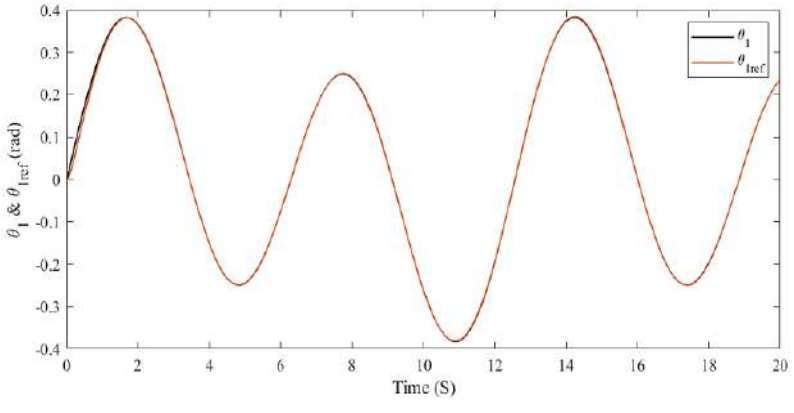


Figure 11 Angular position θ_2 control using a backstepping controller (see online version for colours)

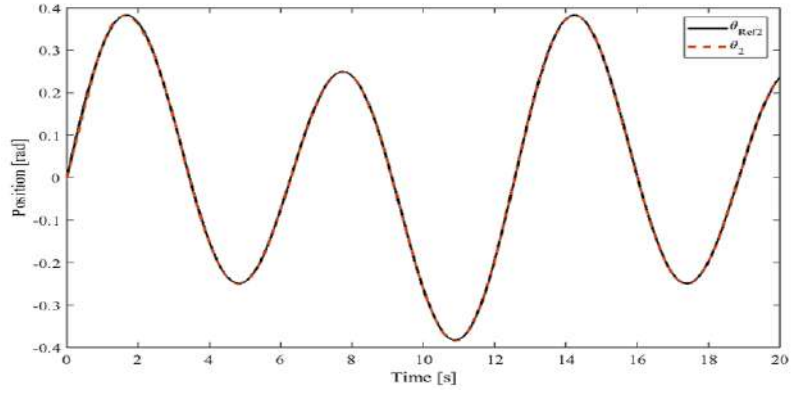


Figure 12 Angular position θ_2 control using a CTC (see online version for colours)

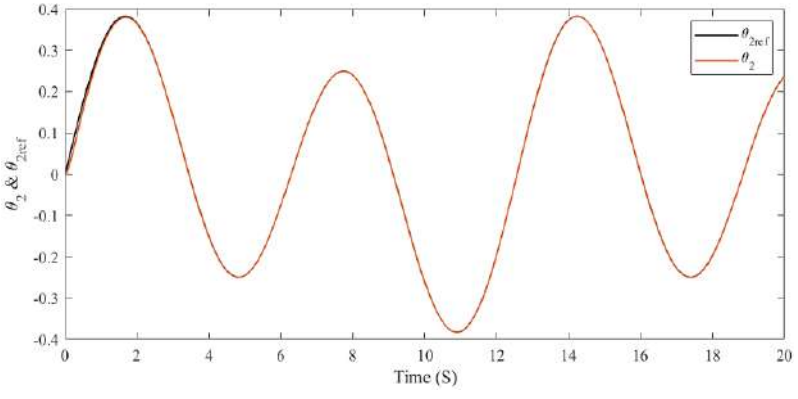


Figure 13 Angular position θ_3 control using a backstepping controller (see online version for colours)

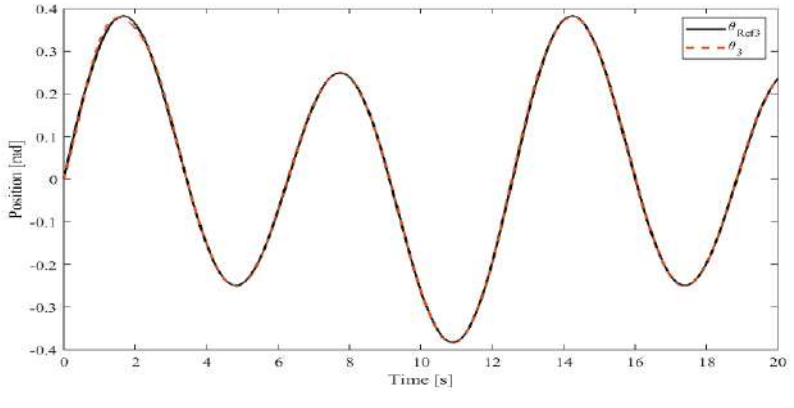


Figure 14 Angular position θ_3 control using a CTC (see online version for colours)

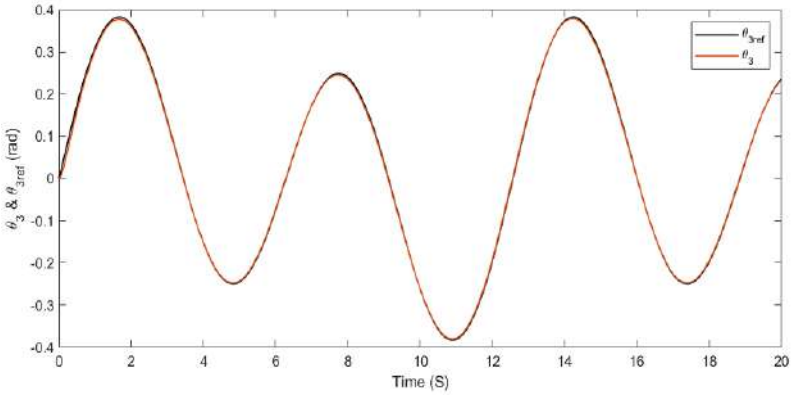


Figure 15 Linear position control of X-axis utilising a backstepping controller (see online version for colours)

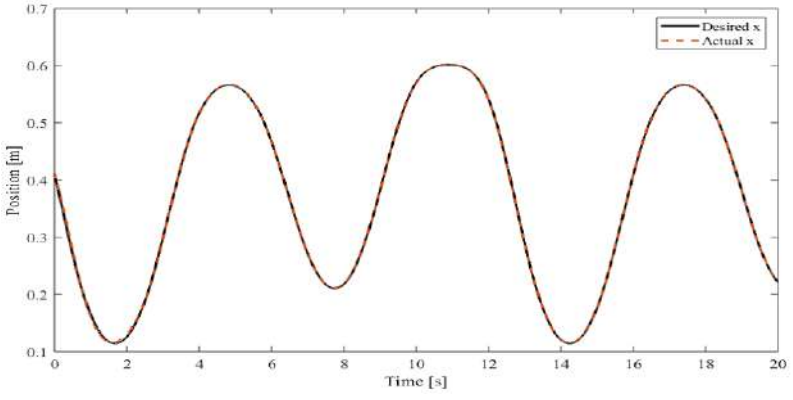


Figure 16 Linear position control of X-axis utilising a CTC (see online version for colours)

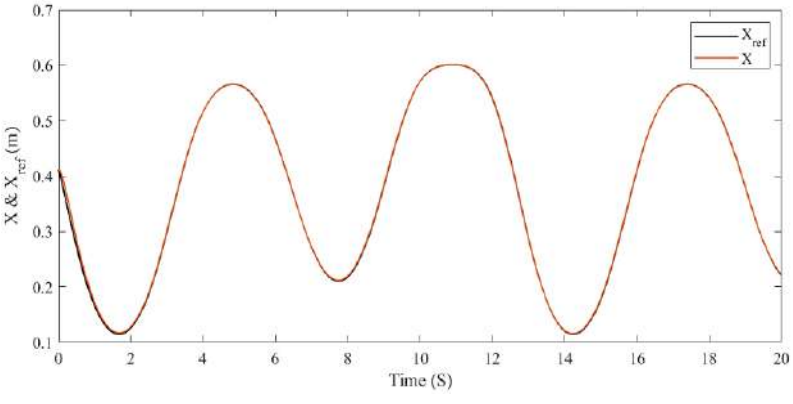


Figure 17 Linear position control of Y-axis utilising a backstepping controller (see online version for colours)

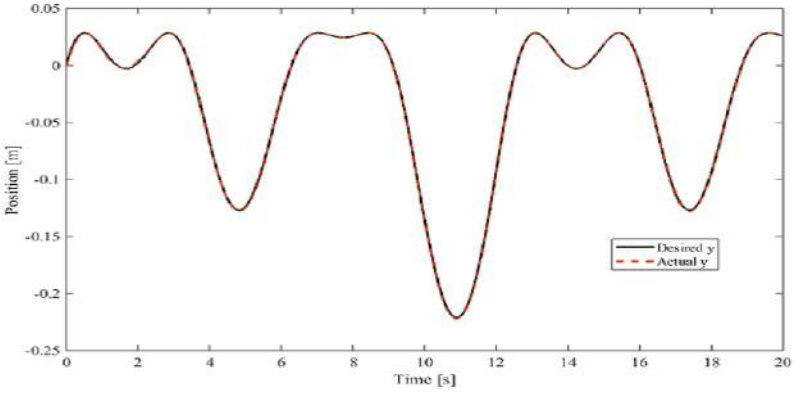


Figure 18 Linear position control of Y-axis utilising a CTC (see online version for colours)

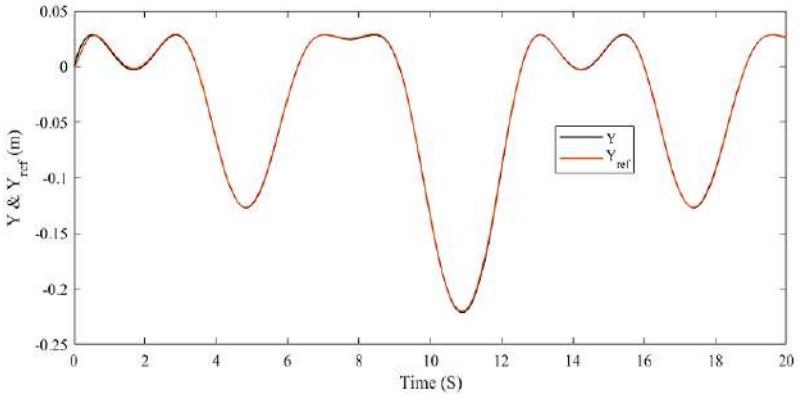


Figure 19 Linear position control of Z-axis utilising a backstepping controller (see online version for colours)

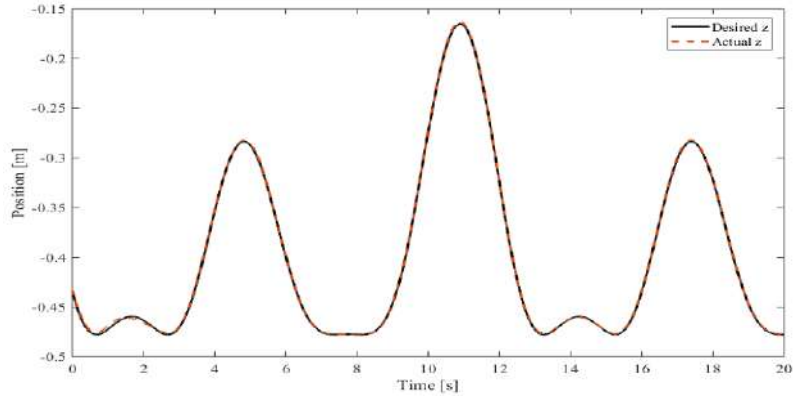


Figure 20 Linear position control of Z-axis utilising a CTC (see online version for colours)

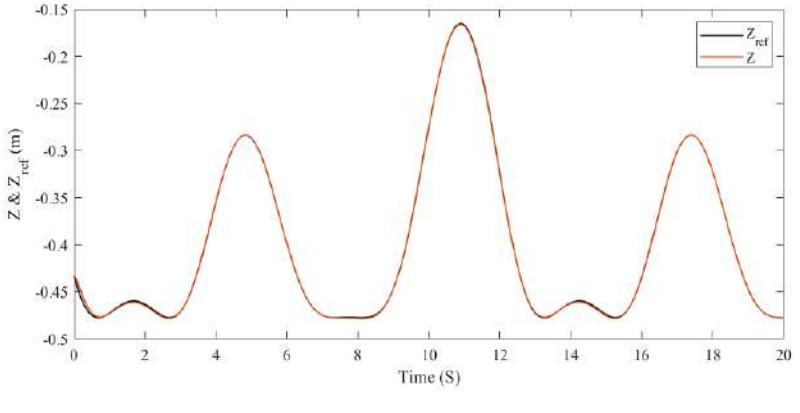


Figure 21 Speed control of the X-axis using a backstepping controller (see online version for colours)

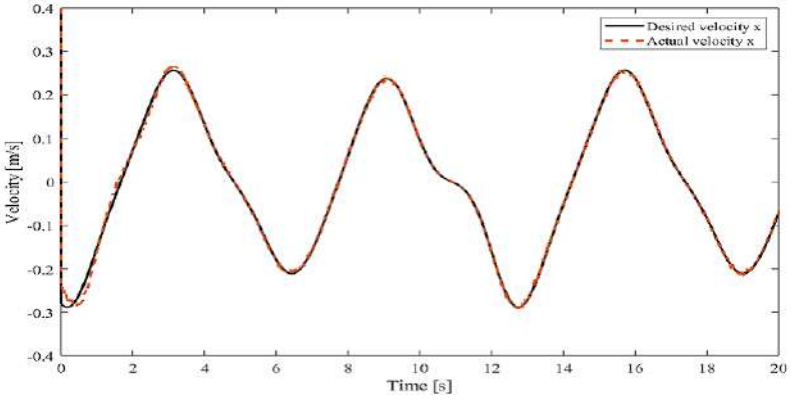


Figure 22 Speed control of the X-axis using a CTC (see online version for colours)

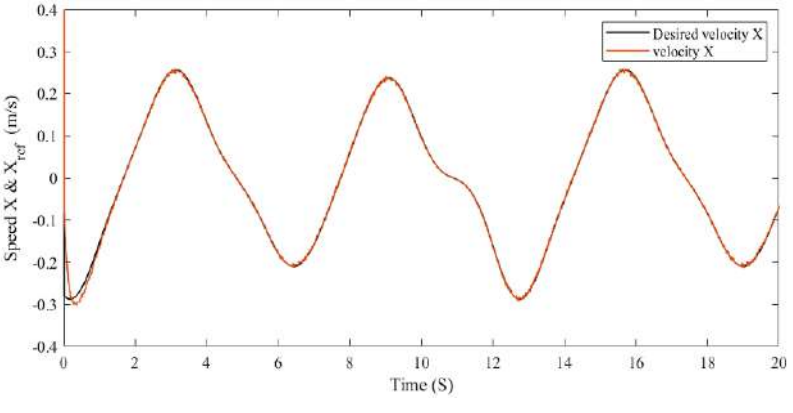


Figure 23 Speed control of the Y-axis using a backstepping controller (see online version for colours)

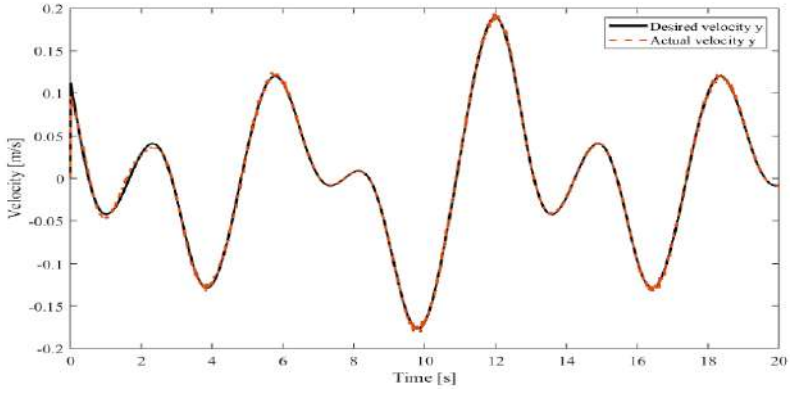


Figure 24 Speed control of the Y-axis using a CTC (see online version for colours)

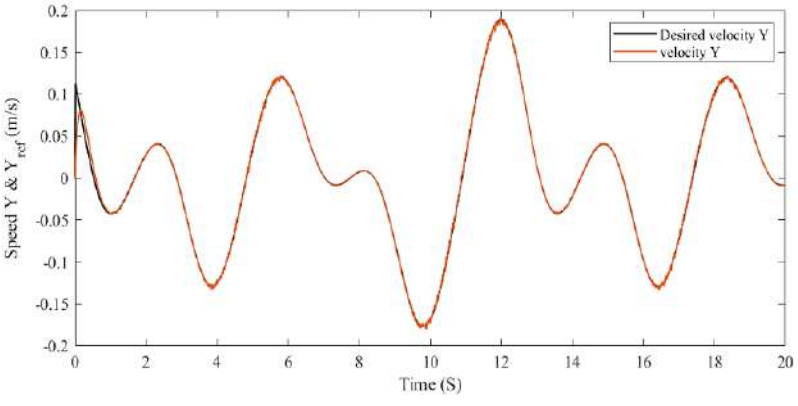


Figure 25 Speed control of the Z-axis using a backstepping controller (see online version for colours)

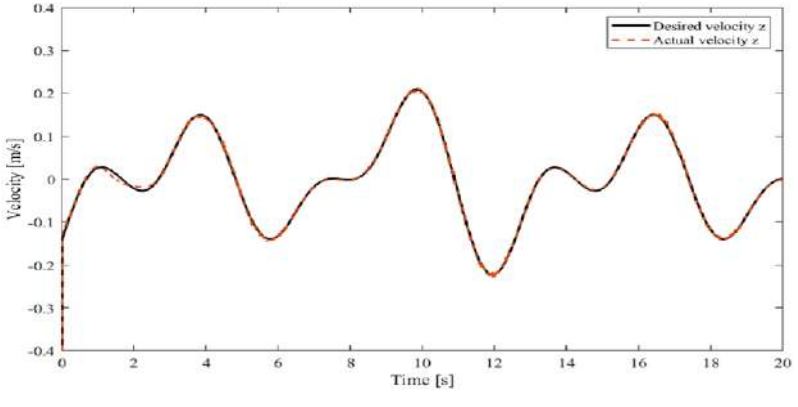


Figure 26 Speed control of the Z-axis using a CTC (see online version for colours)

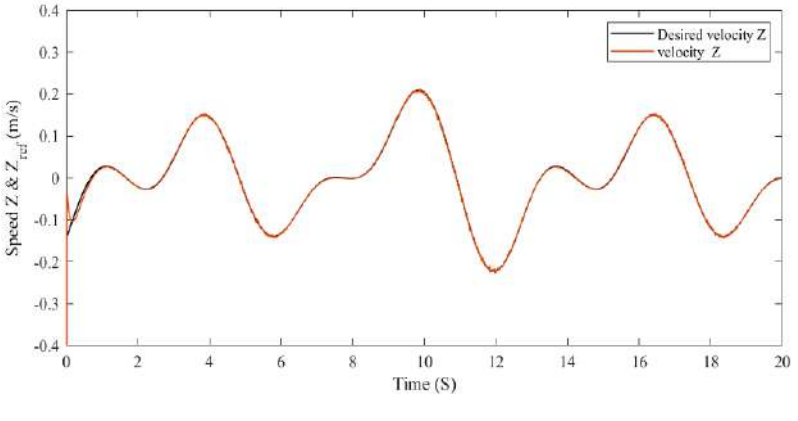


Figure 27 Trajectory tracking error using backstepping controller (see online version for colours)

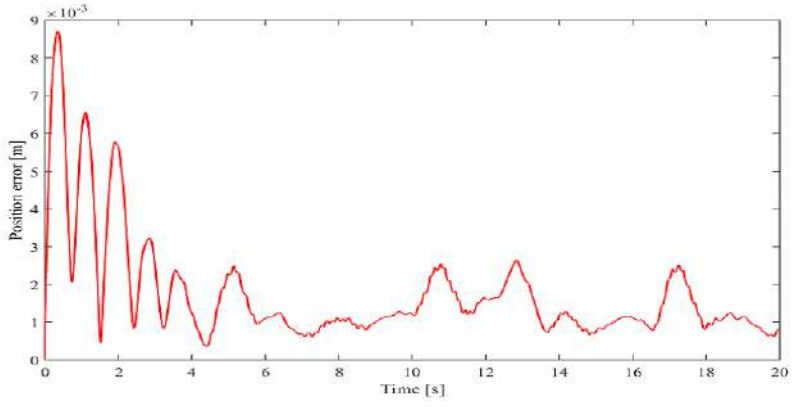
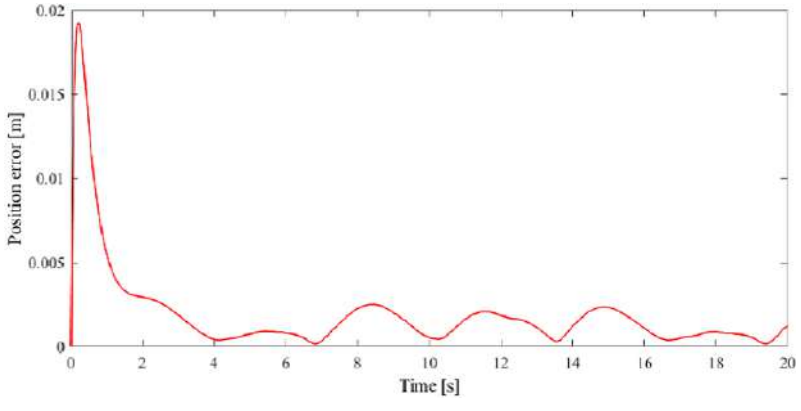


Figure 28 Trajectory tracking error using CTC (see online version for colours)



The 'FPGA in-the-loop' feature of the HDL Coder tool enables the user to simulate the Simulink model within the FPGA platform. This simulation process, depicted in Figures 7 and 8, allows for the validation of the system's behaviour and the identification of any potential design flaws. To accomplish this, a FIL Simulink block is generated using the VHDL language. This block enables the controllers to operate on the Zedboard while the remainder of the system executes in the Simulink environment.

7 Results

The results of the comparison between the FPGA in the loop technique using backstepping and CTC in terms of resource consumption on the Zed board Zynq FPGA board is illustrated in Tables 1, and 2. When using backstepping, the consumed resources are as follows: look-up table (LUT): 61%; flip-flops (FF): 28%; digital signal processor (DSP): 76%. On the other hand, when using CTC, the consumed resources are slightly lower: LUT: 53%; FF: 25%; DSP: 67%. These results indicate that the use of CTC allows for more efficient utilisation of the resources on the Zed board Zynq FPGA. It

consumes fewer LUTs, FFs, and DSPs compared to the use of backstepping. This can be advantageous in terms of cost, energy efficiency, and overall system performance.

Table 1 Resource utilisation of the FPGA board in the context of the backstepping controller

Resources	*Utilisation*	*Available*	*Utilisation (%)*
LUT	32,645	53,200	61.36
LUTRAM	1,208	17400	6.94
FF	30,170	106,400	28.36
BRAM	6.5	140	4.64
DSP	167	220	75.91
IO	2	200	1
BUFG	5	32	15.63
MMCM	1	4	25

Table 2 Resource utilisation of the FPGA board in the context of the CTC

Resources	*Utilisation*	*Available*	*Utilisation (%)*
LUT	28,268	53,200	53.14
LUTRAM	704	17,400	4.05
FF	26,715	106,400	25.11
BRAM	6.5	140	4.64
DSP	147	220	66.82
IO	2	200	1
BUFG	5	32	15.63
MMCM	1	4	25

8 Discussion

To improve the robot's linear and angular positions, the CTC approach appears to provide better performance, leading to smaller errors and reduced oscillations in the linear positions, indicating a more precise control of the robot's motion. However, without specific details about the angular positions, a comprehensive analysis of the robot's overall performance in terms of its three degrees of freedom cannot be provided.

The results of the comparison between the FPGA in the loop technique using backstepping and CTC in terms of trajectory tracking error on the Zed board Zynq FPGA board are indicated in Figures 27 and 28.

When using backstepping, the trajectory tracking errors are observed as follows:

- During the transient regime, the error is approximately 0.09 meters. It oscillates around 0 to 0.06 meters until $t = 4$ seconds. In the steady-state regime, the error oscillates around 0.01 to 0.03 meters.

On the other hand, when using CTC, the trajectory tracking errors are observed as follows:

- During the transient regime, the error is approximately 0.02 meters. It does not exhibit significant oscillations during this period. In the steady-state regime, the error oscillates around 0 to 0.02 meters.

These results indicate that both backstepping and CTC techniques are effective in reducing the trajectory tracking error on the Zed board Zynq FPGA board. However, the CTC approach demonstrates better performance, with a lower error during both the transient and steady-state regimes. This suggests that CTC can provide more precise trajectory tracking compared to backstepping in this particular setup.

9 Conclusions

The performance of backstepping state feedback control and CTC methods in controlling the PUMA 560 robot was compared in this study. Both control methods were implemented on a simulation system and a hardware system using a Zed board Zynq FPGA. The evaluation of the performance focused on trajectory tracking accuracy and speed. The related work section provided an overview of industrial robotics, including the PUMA 560 robot and industrial robot control. Backstepping state feedback control and CTC methods were introduced as widely used nonlinear control methods in industrial robot control. The design and implementation section described the details of each control method's design and implementation process. The experimental results and discussion section presented the obtained results, evaluating the performance of both methods in terms of trajectory tracking accuracy and responsiveness. A comparison of the experimental results for the two control methods was also provided. In summary, the study successfully compared the performance of backstepping state feedback control and CTC methods in controlling the PUMA 560 robot. The results and discussions highlighted the strengths and limitations of each method. Further research and improvements are recommended to enhance the control methods for better performance in industrial robot applications.

Acknowledgements

This research is supported by Automated Systems and soft Computing Lab (ASSCL), Prince Sultan University Riadh, Saudi Arabia. The authors would like to thanks Prince Sultan University, Riadh, Saudi Arabia for thier support.

References

Ahmed, S., Azar, A.T. and Tounsi, M. (2022) 'Design of adaptive fractional-order fixed-time sliding mode control for robotic manipulators', *Entropy*, Vol. 24, No. 12, p.1838.

Ajeil, F.H., Ibraheem, I.K., Azar, A.T. and Humaidi, A.J. (2020) 'Autonomous navigation and obstacle avoidance of an omnidirectional mobile robot using swarm optimization and sensors deployment', *International Journal of Advanced Robotic Systems*, Vol. 17, No. 3, p.1729881420929498.

Al-Safi, A., Al-Khayyat, A., Manati, A.M. and Alhafadhi, L. (2020) 'Advances in FPGA based PWM generation for power electronics applications: literature review', in *2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, IEEE, November, pp.252–259.

Anjum, Z., Guo, Y. and Yao, W. (2021) 'Fault tolerant control for robotic manipulator using fractional-order backstepping fast terminal sliding mode control', *Transactions of the Institute of Measurement and Control*, Vol. 43, No. 14, pp.3244–3254.

Azar, A.T. and Serrano, F.E. (2015) 'Stabilization and control of mechanical systems with backlash', in *Advanced Intelligent Control Engineering and Automation, Advances in Computational Intelligence and Robotics (ACIR) Book Series*, IGI Global, USA.

Bhattacharya, S., Reang, H.B., Golani, P., Nagdev, S., Lodhi, D.S. and Mude, G. (2022) 'Versatile robots and their predomination in pharmaceutical industries', *International Journal of Pharmaceutical Research*, Vol. 14, No. 3, p.9752366.

Chen, Y., Ma, G., Lin, S. and Gao, J. (2012) 'Adaptive fuzzy computed-torque control for robot manipulator with uncertain dynamics', *International Journal of Advanced Robotic Systems*, Vol. 9, No. 6, p.237.

Chen, C., Zhang, C., Hu, T., Ni, H. and Luo, W. (2018) 'Model-assisted extended state observer-based computed torque control for trajectory tracking of uncertain robotic manipulator systems', *International Journal of Advanced Robotic Systems*, Vol. 15, No. 5, p.1729881418801738.

El Ouanjli, N., Mahfoud, S., Bhaskar, M.S., El Daoudi, S., Derouich, A. and El Mahfoud, M. (2022) 'A new intelligent adaptation mechanism of MRAS based on a genetic algorithm applied to speed sensorless direct torque control for induction motor', *International Journal of Dynamics and Control*, Vol. 10, No. 6, pp.2095–2110.

Fekik, A., Azar, A.T., Denoun, H., Kamal, N.A., Hamida, M.L., Kais, D. and Amara, K. (2021) '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)*, Springer International Publishing, Vol. 2, No. 8, pp.445–456.

Fekik, A., Azar, A.T., Hamida, M.L., Denoun, H., Kais, D., Saidi, S.M. and Njima, C.B. (2023) 'Sliding mode control of the PUMA 560 robot', in *2023 International Conference on Control, Automation and Diagnosis (ICCAD)*, May, pp.1–6.

Fekik, A., Denoun, H., Azar, A.T., Kamal, N.A., Zaouia, M. and Benyahia, N. (2021) 'Direct power control of three-phase PWM-rectifier with backstepping control', in *Backstepping Control of Nonlinear Dynamical Systems*, pp.215–234, Elsevier, Academic Presses.

Gadaleta, M., Pellicciari, M. and Berselli, G. (2019) 'Optimization of the energy consumption of industrial robots for automatic code generation', *Robotics and Computer-Integrated Manufacturing*, Vol. 57, No. 1, pp.452–464.

Hady, S.A., Ali, A.A., Breesam, W.I., Saleh, A.L., Al-Yasir, Y.I.A. and Abd-Alhameed, R.A. (2023) 'Control of three-links robot arm based on fuzzy neural Petri Nets', *International Journal of Automation and Control*, Vol. 17, No. 1, pp.116–131.

Hanna, A., Larsson, S., Götvall, P.L. and Bengtsson, K. (2022) 'Deliberative safety for industrial intelligent human-robot collaboration: regulatory challenges and solutions for taking the next step towards Industry 4.0', *Robotics and Computer-Integrated Manufacturing*, Vol. 78, No. 1, p.102386.

Ibraheem, G.A.R., Azar, A.T., Ibraheem, I.K. and Humaidi, A.J. (2020) 'A novel design of a neural network-based fractional PID controller for mobile robots using hybridized fruit fly and particle swarm optimization', *Complexity*, Vol. 2020, No. 1, pp.1–18.

Ibrahim, M.Y. (1996) 'Computer study on the effect of an articulated robot's parameters on its dynamic characteristics under different balancing conditions', *Mathematics and Computers in Simulation*, Vol. 41, No. 3, pp.297–306.

Ibro, M. and Marinova, G. (2022) 'Literature review on FPGA-based e-learning: power consumption design methodologies perspective', in *2022 29th International Conference on Systems, Signals and Image Processing (IWSSIP)*, IEEE, June, pp.1–4.

Jain, G., Jain, A. and Mishra, D. (2023) 'Applications of the internet of robotic things in Industry 4.0 based on several aspects', in *Artificial Intelligence Techniques in Human Resource Management*, 1st ed., pp.127–152, CRC Press, Taylor & Francis Group, Apple, Academic Press.

Jokić, D., Lubura, S., Rajs, V., Bodić, M. and Šiljak, H. (2020) 'Two open solutions for industrial robot control: the case of PUMA 560', *Electronics*, Vol. 9, No. 6, p.972.

Khamis, A., Meng, J., Wang, J., Azar, A.T., Prestes, E., Takács, Á. and Haidegger, T. (2021) 'Robotics and intelligent systems against a pandemic', *Acta Polytechnica Hungarica*, Vol. 18, No. 5, pp.13–35.

Khan, M.A. (2020) *Design and Control of a Robotic System based on Mobile Robots and Manipulator Arms for Picking in Logistics Warehouses*, Doctoral dissertation, Normandie.

Larbi, M., Guechi, E-H., Maidi, A., Ounnas, D. and Belharet, K. (2023) 'Observer-based 2D tracking control for a vascular microrobot based on the T-S fuzzy model', *International Journal of Automation and Control*, Vol. 17, No. 6, pp.573–594.

Lewis, F.L., Dawson, D.M. and Abdallah, C.T. (2003) *Robot Manipulator Control: Theory and Practice*, CRC Press, USA.

Luan, F., Yang, X., Chen, Y. and Regis, P.J. (2022) 'Industrial robots and air environment: a moderated mediation model of population density and energy consumption', *Sustainable Production and Consumption*, Vol. 30, No. 1, pp.870–888.

Mahfoud, S., Derouich, A., El Ouanjli, N., Mossa, M.A., Bhaskar, M.S., Lan, N.K. and Quynh, N.V. (2022) 'A new robust direct torque control based on a genetic algorithm for a doubly-fed induction motor: experimental validation', *Energies*, Vol. 15, No. 15, p.5384.

Maraveas, C., Asteris, P.G., Arvanitis, K.G., Bartzanas, T. and Loukatos, D. (2023) 'Application of bio and nature-inspired algorithms in agricultural engineering', *Archives of Computational Methods in Engineering*, Vol. 30, No. 3, pp.1979–2012.

Marcus, H., Nandi, D., Darzi, A. and Yang, G.Z. (2013) 'Surgical robotics through a keyhole: from today's translational barriers to tomorrow's 'disappearing' robots', *IEEE Transactions on Biomedical Engineering*, Vol. 60, No. 3, pp.674–681.

Mehendale, N., Aurobindo, A. and Kakatkar, S.(2022) *A Review on Applications of FPGAS*, SSRN: 4094931.

Meng, F., Zhao, L. and Yu, J. (2020) 'Backstepping based adaptive finite-time tracking control of manipulator systems with uncertain parameters and unknown backlash', *Journal of the Franklin Institute*, Vol. 357, No. 16, pp.11281–11297.

Merheb, A.R. (2008) *Nonlinear Control Algorithms Applied to 3 DOF PUMA Manipulator*, pp.1–114, Doctoral dissertation, Middle East Technical University.

Mohamed, M.J., Oleiwi, B.K., Abood, L.H., Azar, A.T. and Hameed, I.A. (2023) 'Neural fractional order PID controllers design for 2-link rigid robot manipulator', *Fractal Fract.*, Vol. 7, No. 9, p.693.

Moreno-Valenzuela, J., Moyrón, J. and Montoya-Cháirez, J. (2023) 'Limited integrator anti-windup-based control of input-constrained manipulators', *IEEE Transactions on Industrial Electronics*, Vol. 71, No. 2, pp.1738–1748.

Najm, A.A., Ibraheem, I.K., Azar, A.T. and Humaidi, A.J. (2020) 'Genetic optimization-based consensus control of multi-agent 6-DoF UAV system', *Sensors*, Vol. 20, No. 12, p.3576.

Pandit, S. and Shet, V.N. (2017) 'Review of FPGA based control for switch mode converters', in *2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, IEEE, February, pp.1–5.

Qasim, S.M., Abbasi, S.A. and Almashary, B. (2009) 'A review of FPGA-based design methodology and optimization techniques for efficient hardware realization of computation intensive algorithms', *2009 International Multimedia, Signal Processing and Communication Technologies*, pp.313–316.

Rahali, H., Zeghlache, S. and Benyettou, L. (2021) 'Fault tolerant control of robot manipulators based on adaptive fuzzy type-2 backstepping in attendance of payload variation', *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 4, pp.312–325.

Ramu, S.K., Irudayaraj, G.C.R., Subramani, S. and Subramaniam, U. (2020) 'Broken rotor bar fault detection using Hilbert transform and neural networks applied to direct torque control of induction motor drive', *IET Power Electronics*, Vol. 13, No. 15, pp.3328–3338.

Rani, K. and Kumar, N. (2022) 'Intelligent optimal hybrid motion/force control of constrained robot manipulator', *International Journal of Automation and Control*, Vol. 16, No. 6, pp.740–768

Sahu, N., Londhe, N.D. and Kshirsagar, G.B. (2017) 'FPGA applications in inverter and converter circuits: a review on technology, benefits and challenges', in *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, IEEE, March, pp.1–5.

Salinas, A., Moreno-Valenzuela, J. and Kelly, R. (2016) 'A family of nonlinear PID-like regulators for a class of torque-driven robot manipulators equipped with torque-constrained actuators', *Advances in Mechanical Engineering*, Vol. 8, No. 2, p.1687814016628492.

Shang, W.W., Cong, S. and Ge, Y. (2012) 'Adaptive computed torque control for a parallel manipulator with redundant actuation', *Robotica*, Vol. 30, No. 3, pp.457–466.

Siciliano, B., Sciavicco, L., Villani, L. and Oriolo, G. (2009) 'Differential kinematics and statics', *Robotics: Modelling, Planning and Control*, pp.105–160, Springer, UK.

Singh, G. and Banga, V.K. (2022) 'Kinematics and trajectory planning analysis based on hybrid optimization algorithms for an industrial robotic manipulators', *Soft Computing*, Vol. 26, No. 21, pp.11339–11372.

Slotine, J.J.E. and Li, W. (1991) *Applied Nonlinear Control*, Vol. 199, p.705, Prentice Hall, Englewood Cliffs, NJ.

Soliman, M., Azar, A.T., Saleh, M.A. and Ammar, H.H. (2020) 'Path planning control for 3-omni fighting robot using PID and fuzzy logic controller', in *The International Conference on Advanced Machine Learning Technologies and Applications* pp.442–452.

Spong, M.W., Hutchinson, S. and Vidyasagar, M. (2020) *Robot Modeling and Control*, John Wiley & Sons, UK.

Taylor, R.H., Simaan, N., Menciassi, A. and Yang, G.Z. (2022) 'Surgical robotics and computer-integrated interventional medicine', *Proceedings of the IEEE*, Vol. 110, No. 7, pp.823–834.

Vaidyanathan, S. and Azar, A.T. (2020) *Backstepping Control of Nonlinear Dynamical Systems*, Academic Press, ISBN: 978-0128175828.

Vaidyanathan, S., Jafar, S., Pham, V.T., Azar, A.T. and Alsaadi, F.E. (2018) 'A 4-D chaotic hyperjerk system with a hidden attractor, adaptive backstepping control and circuit design', *Archives of Control Sciences*, Vol. 28, No. 2, pp.239–254.

Van, M., Mavrovouniotis, M. and Ge, S.S. (2018) 'An adaptive backstepping nonsingular fast terminal sliding mode control for robust fault tolerant control of robot manipulators', *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 49, No. 7, pp.1448–1458.

Wang, E.Z., Lee, C.C. and Li, Y. (2022) 'Assessing the impact of industrial robots on manufacturing energy intensity in 38 countries', *Energy Economics*, Vol. 105, No. 3, p.105748.

Wang, W., Zhou, X., Xu, B., Chen, S., Lu, M., Li, J. and Gu, Y. (2024) 'Few-shot reasoning-based safe reinforcement learning framework for autonomous robot navigation', *International Journal of Automation and Control*, Vol. 18, No. 1, pp.30–52.

Zhu, X., Wan, J., Xu, W. and Zhou, C. (2022) 'Robust attitude stabilisation reactive control of a quadruped robot under load disturbance', *International Journal of Automation and Control*, Vol. 16, No. 6, pp.649–669.