IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 14 September 2024, accepted 23 November 2024, date of publication 2 December 2024, date of current version 10 December 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3510226

RESEARCH ARTICLE

Modeling the DLR-HIT II Robotic Hand: A Dual-Platform Simulation Approach in MATLAB and CoppeliaSim

ALI AL-SHAHRABI[1,2], **MASOUD JAHANBAKHSH JAVID**[1], **ASHRAF A. FAHMY**[1], **CHRISTIAN A. GRIFFITHS**[1], **(Senior Member, IEEE), AND CHUNXU LI**[1]

[1]Department of Mechanical Engineering, Swansea University, SA1 8QQ Swansea, U.K.

[2]College of Engineering, Al-Nahrain University, Baghdad 64074, Iraq

Corresponding author: Ali Al-Shahrabi (a.m.a.al-shahrabi@swansea.ac.uk)

This work was supported by Al-Nahrain University.

ABSTRACT The DLR-HIT II hand is a highly flexible and dexterous robot capable of performing complex grasping and manipulation tasks. This study presents a comprehensive simulation of the artificial hand using both MATLAB's Simscape Multibody toolbox and CoppeliaSim (formerly known as V-REP). The simulation model accurately captures the robotic hand's mechanical structure and control aspects, allowing for an in-depth analysis of its behaviour. The model incorporates the technical characteristics of the hand's joints, links, and constraints into a multibody dynamic's framework. The robotic hand is simulated on two platforms: MATLAB, to verify the control system's performance, and CoppeliaSim, which offers a realistic 3D simulation environment with real-time interaction with other robots or objects. The simulation results enable further optimization and enhancement of the robotic hand's design and control strategies for practical robotics and automation applications. This research provides valuable insights into the hand's capabilities and limitations, serving as a foundation for future studies.

INDEX TERMS DLR-HIT II hand, simulation, MATLAB, simscape multibody toolbox, CoppeliaSim V-REP.

I. INTRODUCTION

Robots have been widely used in the manufacturing [1], autonomous transport [2], domestic tasks [3], construction [4], agriculture [5], and mining [6]. The medical field has also made considerable progress in performing delicate surgeries [7]. Furthermore, robots have been deployed to perform hazardous tasks in military and space applications, surpassing human capabilities and preventing risks to human life [8]. Consequently, it can be considered that robots are an essential element of equipment in automatic control and has evolved to to play a significant role in diverse fields and applications.

Deploying the robot directly into the physical environment can be time-consuming, risky, and costly. Moreover, this approach often complicates the testing and debugging processes [9]. Simulating robots will aid in testing procedures by reducing initial costs and avoiding unnecessary hazards [9]. It is also an effective tool for academic research and education [10]. Moreover, collecting data on controller efficiency from the physical workspace can be challenging, since the data is vast and subject to sudden changes such us interacting with other objects [11]. Nonetheless, the process is made much simpler if the environment is simulated by software.

Even though, the simulation of robot eases the cited challenges, simulating DLR-HIT II robotic hand is difficult owing to the complexity of the model, especially, the last two joints of the each finger are mechanically connected by steel cables using a linking mechanism that preserves the angles identical for both joints [12]. In essence, one motor is used to control the last two joints of every finger.

In this paper, DLR-HIT II hand is simulated using MATLAB. Simscape Multibody toolbox is employed for

The associate editor coordinating the review of this manuscript and approving it for publication was Chaitanya U. Kshirsagar.

TABLE 1. Joints constrains for one finger.

Joint angle	Joint constrain(deg)	Joint constrain(rad)
θ_1	$[-20^\circ \quad 20^\circ]$	[-0.35 0.35]
θ_2	$[0 \quad 90^\circ]$	[0 1.57]
θ_3	$[0 \quad 90^\circ]$	[0 1.57]
θ_4=θ_3	$[0 \quad 90^\circ]$	[0 1,57]

this purpose. The multibody simulation environment offered by this toolbox is suitable for 3D mechanical systems like aeroplane landing gear, automobile suspensions, robots, and construction equipment [13]. Additionally, the robotic hand is modeled by CoppeliaSim, which is a popular and effective computer simulation software for robotics research and development. Precise robotic simulations with real-time interaction capabilities are supported by CoppeliaSim's cohesive development environment [14]. The reason for simulating the hand in two programs is MATLAB offers machine learning control algorithms and sophisticated control system, so that the model can be tested first and then can be validated in CoppeliaSim platforms as it has better visualization and real-time engine when compared to MATLAB.

The robot can be modelled utilizing blocks representing bodies, joints and constraints, as well as multi types of sensors. Computer aided design (CAD) assemblies can be imported to MATLAB and CoppeliaSim incorporates all masses, joints, constraints, inertias, and three-dimensional geometry into the model that is created. The dynamics of the system can be observed in an automatically generated 3D animation [13].

II. THE KINEMATIC OF THE DLR-HIT II ROBOTIC HAND

The forward and inverse Kinematic will be derived in this section. The DLR-HIT Hand II Has five fingers and fifteen degrees of freedom (DOFs) overall, All fingers are constructed to be symmetrical to acquire a high level of mobility. Every finger is intended to be symmetrical and has total of 4 joints and three DOFs. The last two joints for each finger are mechanically connected by metallic cables using a coupling mechanism that preserves the angles consistency [12]. The joints constrain for each finger are shown in Table 1 [12].

The Denavit-Hartenberg (DH) model [15] is used to represent the relation between the joints and links of the dexterous robotic hand by utilising the geometrical structure of the system. This approach fixes the coordinate on each link of the robot structure. The homogeneous transformation matrix yields the spatial relation among two adjacent links. In order to determine the end-effector's pose with respect to the base frame, the robot's kinematic solution can be computed by applying the transformation sequence [16].

Fig. 2 shows the coordinate system for one finger of the human hand. As there are no motors in the palm, only the fingers need to be considered in the DLR-HIT II structure. The length of a link between two neighbouring finger joints is denoted by L_i, and the coordination system's base is represented by the letter "o." Table 2 below shows the DH parameters for a single finger.The DH parameter table is constructed depending on the world frame of CoppeliaSim. The framework for the DLR-HIT II hand can be easily established through determining the DH parameters for each finger. This allows for a thorough understanding of the hand's kinematic structure and facilitates effective implementation.

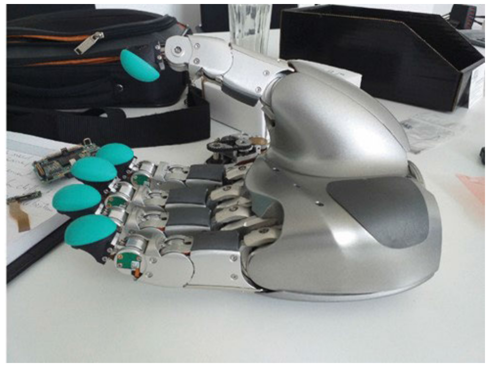

FIGURE 1. The DLR-HIT II hand [16].

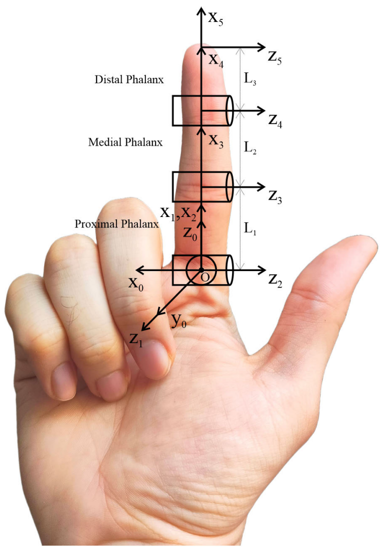

FIGURE 2. The frames distribution on one finger.

TABLE 2. DH parameters for one finger of human hand.

Frames	α_i	$\mathbf{a}_i$	θ_i	$\mathbf{d}_i$
1	$-\pi/2$	0	$-\pi/2+\theta_1$	0
2	$\pi/2$	0	θ_2	0
3	0	L_1	θ_3	0
4	0	L_2	$\theta_4=\theta_3$	0
5	0	L_3	0	0

By using the general transformation matrix (1) of DH, the transformation matrix between each two consecutive frames is obtained.

$$T_i^{i-1} = \left[\begin{array}{ccc|c} & R_i^{i-1} & & P_i^{i-1} \\ \hline \cos\theta_i & -\sin\theta_i & 0 & a_{i-1} \\ \sin\theta_i\cos\alpha_{i-1} & \cos\theta_i\cos\alpha_{i-1} & -\sin\alpha_{i-1} & -d_i\sin\alpha_{i-1} \\ \sin\theta_i\sin\alpha_{i-1} & \cos\theta_i\sin\alpha_{i-1} & \cos\alpha_{i-1} & d_i\cos\alpha_{i-1} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \quad (1)$$

where R_i^{i-1} is the rotational matrix between two frames and P_i^{i-1} is the translation between two frames. to find T_1^0 is the transformation matrix between frame 0 and frame 1 and can be found as follows:

$$T_1^0 = \begin{bmatrix} \cos(\theta_1 - pi/2) & -\sin(\theta_1 - pi/2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\sin(\theta_1 - pi/2) & -\cos(\theta_1 - pi/2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

That is leading to the next transformation matrix T_2^1:

$$T_2^1 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

The transformation matrix between frame 2 and 3 is calculated in the following matrix:

$$T_3^2 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & L_1 \\ \sin\theta_3 & \cos\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

T_4^3 is computed as follows:

$$T_4^3 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & L_2 \\ \sin\theta_3 & \cos\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

The last transformation matrix between the frame 5, that is attached to the tip of the finger, and the previous frame 4 is shown in the following matrix:

$$T_4^3 = \begin{bmatrix} 1 & 0 & 0 & L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

Thus, through multiplying the computed transformation matrices for every frame, the finger pose's tip with regard to its base can be determined by the following equation:

$$T_5^0 = T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 \tag{7}$$

That leads to:

$$T_5^0 = \begin{bmatrix} c_{332}s_1 & -s_{332}c_1 & -c_1 & L2c_{32}s_1 + L1c_2s_1 + L3c_{32}s_1 \\ s_{332} & c_{332} & 0 & L2s_{32} + L1s_2 + L3s_{332} \\ c_{332}c_1 & -s_{332}c_1 & s_1 & L2c_{32}c_1 + L1c_2c_1 + L3c_{32}c_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

The rotational matrix R_5^0 is:

$$R_5^0 = \begin{bmatrix} c_{332}s_1 & -s_{332}c_1 & -c_1 \\ s_{332} & c_{332} & 0 \\ c_{332}c_1 & -s_{332}c_1 & s_1 \end{bmatrix} \tag{9}$$

While the position of the tip of the finger is as follows:

$$P_5^0 = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} L2c_{32}s_1 + L1c_2s_1 + L3c_{32}s_1 \\ L2s_{32} + L1s_2 + L3s_{332} \\ L2c_{32}c_1 + L1c_2c_1 + L3c_{32}c_1 \end{bmatrix} \tag{10}$$

where:

$$s_1 = \sin\theta_1, c_1 = \cos\theta_1, s_2 = \sin\theta_2,$$
$$c_2 = \cos\theta_2, c_{32} = \cos(\theta_3 + \theta_2),$$
$$s_{332} = \sin(2\theta_3 + \theta_2), c_{332} = \cos(2\theta_3 + \theta_2).$$

Since there is no tool attached at the tip of each finger of DLR-HIT II and the robotic hand is not attached to robotic arm, then there is no need to find the orientation of the fingers tip. A vector of the the position (μ) will be constructed as follows:

$$\mu = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{11}$$

The linear velocity Jacobian matrix J_v will be the derivative of position (X, Y, Z) with respect to joints angles $(\theta_1, \theta_2, \theta_3)$ as follows:

$$J_v = \begin{bmatrix} \frac{\partial X}{\partial \theta_1} & \frac{\partial X}{\partial \theta_2} & \frac{\partial X}{\partial \theta_3} & \frac{\partial X}{\partial \theta_3} \\ \frac{\partial Y}{\partial \theta_1} & \frac{\partial Y}{\partial \theta_2} & \frac{\partial Y}{\partial \theta_3} & \frac{\partial Y}{\partial \theta_3} \\ \frac{\partial Z}{\partial \theta_1} & \frac{\partial Z}{\partial \theta_2} & \frac{\partial Z}{\partial \theta_3} & \frac{\partial Z}{\partial \theta_3} \end{bmatrix} \tag{12}$$

The DLR-HIT II robotic hand's inverse kinematics can be solved numerically with the use of the Newton-Raphson method. Finding the parameters of the joint that enable the robotic hand's end effector to be positioned in the desired manner is the goal of inverse kinematics. In order to reduce the error within the current end effector position and the objective's position, the Newton-Raphson method constantly modifies the joint angles. The method begins with an estimation of the joint angles and calculates the error using the Jacobian matrix, which connects the joint velocities of the end effector velocities. The Newton-Raphson method converges to the solution that results in the desired end-effector position by constantly updating the joint angles in away to make this error smaller. The DLR-HIT II hand is especially appropriate for this method given its intricate joint structure and high dexterity, which make mathematical solutions difficult. The Newton-Raphson method gives a reliable means for handling the kinematics of the hand's nonlinearities and constraints, leading to accurate approximation for the joint angles. The pseudo code for Newton-Raphson method is as follows:

III. THE METHODOLOGY OF SIMULATING THE DLR-HIT II HAND IN MATLAB AND CoppeliaSim

The simulation of robotic systems requires powerful, precise, and user-friendly tools. This work aims to evaluate the performance of the DLR-HIT II robotic hand and develop a model suitable for future integration with deep learning

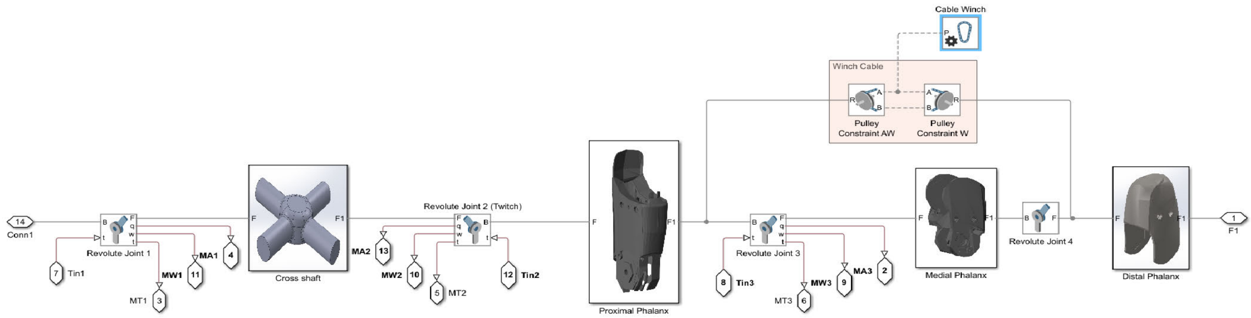

FIGURE 3. Finger's Simulink block diagram.

Algorithm 1 Newton-Raphson Method

Initialize positions of finger's tip $\mu_{initial}$
Initialize joint values $q_{initial} = (\theta_1, \theta_2, \theta_3)$
for $i = 1$ to 100 **do**
  $\theta_1 \leftarrow q_{initial}(1)$
  $\theta_2 \leftarrow q_{initial}(2)$
  $\theta_3 \leftarrow q_{initial}(3)$
  Compute pseudo inverse of Jacobian matrix J_v
  Compute estimated position μ_e
  Compute task-space error $\delta = \mu_a - \mu_e$
  if $|\delta| < 1e-5$ **then**
    break {Check convergence}
  end if
  Update joint values $q_{new} \leftarrow q_{intial} + J_v^{-1}\delta$
end for

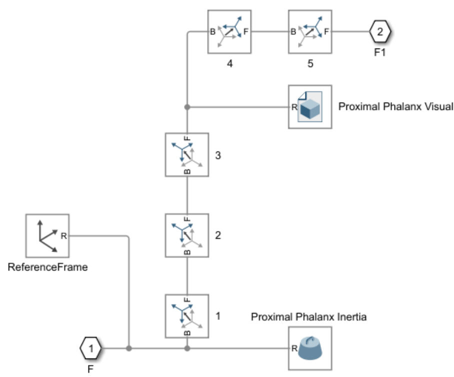

FIGURE 4. The block diagram of the Proximal phalanx.

TABLE 3. DLR-HIT II robotic hand dimensions and weights.

Parameters	Length (mm)	Width (mm)	Weight (g)
Distal Phalanx	25	19.2	24
Medial Phalanx	25	20	25
Proximal Phalanx	55	20	61
Hand Palm	140.4	126.2	-

algorithms and advanced control systems. Since not all the necessary tools for simulation and control of the robotic hand could be found in one software, the robotic hand was modelled in MATLAB and CoppeliaSim. MATLAB is a mathematical modelling and control environment. The features found in MATLAB can provide accurate verification of the system which can be used for control system learning, developing AI control techniques, and evaluating dynamic performance through simulations. The CoppeliaSim can provide a 3D dynamic modelling environment that can simulate the total workspace including the workpieces and robot arms. CoppeliaSim is designed specifically for real-time robotics simulation, offering better performance for simulating real-time interactions and better representation compared to MATLAB.

So, the model was first made in MATLAB to control the simple movement of the fingers and verification of its performance. Subsequently, the model in CoppeliaSim facilitated the simulation and analysis of object grasping tasks. To ensure a comprehensive visualization of the simulation, efforts were made to ensure that the CAD files used for the robotic hand's appearance accurately reflected the actual model. To do this, STL files for the DLR-HIT II have been utilized from the GitHub repository [17], where the fingers' phalanx are downloaded, then modified in SolidWorks, and assembled in MATLAB/Simulink aiming to achieve the same visual quality as the DLR-HIT II robotic hand. Due to the low resolution of the files, they were modified in SolidWorks (the coarse bodies were converted to fine mesh bodies) for more accurate visualization.

DLR-HIT II consists of five identical fingers with three joints each [18]. The actuation system of this device uses brushless motors and belts to ensure its accuracy. Using a differential bevel gear, the base joint of each finger has two DOF [18]. The dimensions and the mass of each finger's parts of the DLR-HIT II used in the simulation are presented in Table 3 [12].

Each finger's weight is 220 g including motors and other components, and the whole robotic hand weight is 1.5 Kg [12].

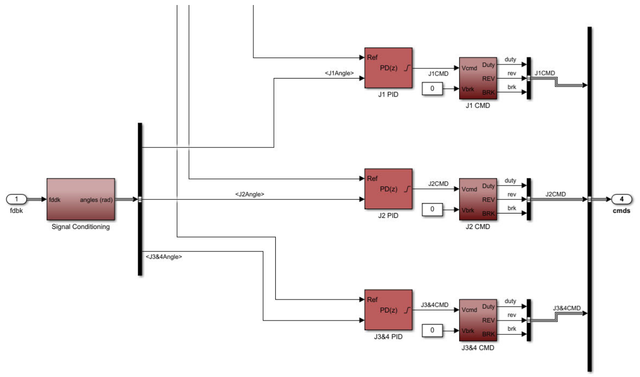

FIGURE 5. The controller system block diagram.

A. SIMULATION IN MATLAB/SIMULINK

Simulation of a robotic hand using MATLAB/Simulink toolboxes involves creating a detailed and dynamic model similar to the behaviour of a physical robotic hand. This process starts with designing the kinematic and dynamic models of the robotic hand, which includes defining the joints, linkages, and actuators using the Simulink Simscape toolboxes. Then PID blocks are employed to develop control algorithms, simulate the hand's movements, and verify accuracy. The simulation environment in MATLAB/Simulink supports extensive analysis and visualization tools, enabling fine-tuning control strategies, optimising performance, and validating the robotic hand's performance. This integrated approach provides a robust iterative design and testing platform, significantly reducing development time and costs.

1) MECHANICAL STRUCTURE

A highly complex mechanical structure, the DLR-HIT II is designed to emulate the human hand in terms of proficiency and capabilities [18]. A universal joint block from Simscape MATLAB/Simulink is required to model base joint. This block is designed to rotate around the x-axis and y-axis [19]. As shown in Figure 2, the rotations are considered to be around the z-axis which creates a conflict in the calculations. The base joint differential bevel gear mass and inertia affect the forces and torque generated [18]. To model DLR-HIT II 2 DOF base joint, a cross shaft with a similar weight and dimension is used.

Modeling the structure of each finger also presented challenges concerning joints 3 and 4. These joints are actuated by the same motor. The motor is placed at joint 3 location, actuating both joint 3 and 4 simultaneously using a belt. To simulate joints 3 and 4 with the right actuation model, two pulley constraints and a cable were used in the model. The pulley constraint size is 3 mm and its mass is neglected. So, the motor torque compensates for the movement of both links 3 and 4. The following block diagram in Figure 3 represents each finger simulation model in Simulink. The angular velocity and the position of the joints are sensed from each joint block which is used to feedback the DC motors and the control systems. The hand's palm is considered as the robotic hand's base with its coordinate frame placed at the same position as the world frame. The CAD files improve the visuality of the model and an inertia block parameter was used to define the mass, the center of mass, and the moment of inertia values of each phalanx. The block diagram model used for each phalanx is shown in Figure 4.

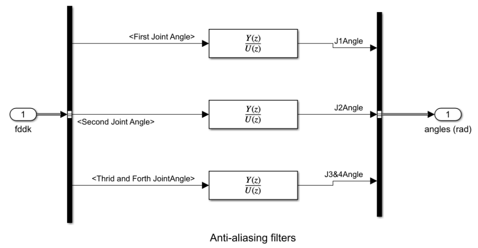

FIGURE 6. Signal conditioning block diagram.

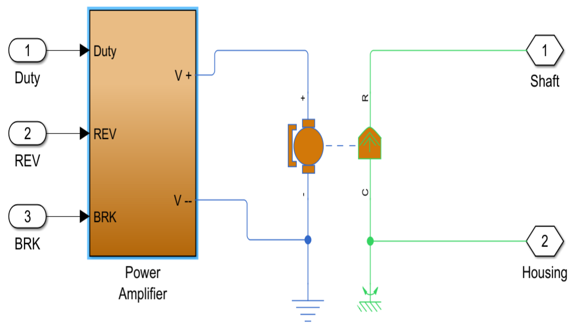

FIGURE 7. DC motor model block diagram.

2) ACTUATION AND CONTROL SYSTEM

A variety of actuation and control methods were used for the simulation to achieve a more realistic simulation. This led us to increase the similarity of the model to the actual robot. Consequently, a power amplifier system and DC motor model were used to actuate the finger joints, PD controllers were used for computed torque control implementation, and signal conditioning filters were used to filter signal noise. Each finger joint was actuated by a motor, so three PD controller blocks were used to control a finger. Tuning each PD controller block separately can be challenging and time-consuming, so the multi-loop tuning method can be helpful [20]. The PD block generates signals that are sent to a command block to make duty, reverse threshold, and braking threshold voltage references to be used on the H-Bridge block. To avoid saturation of the output of the PD block multi-loop tuning session, it is limited to $+0.5$ to -0.5 Nm which these values obtained experimentally. The controlling system and command block diagrams can be seen in Fig. 6 and Fig. 7. According to Fig. 8, a signal conditioning subsystem for the controlling system is considered to filter feedback signal noise generated by the DC motors and other system components since the PD controllers are more sensitive to noise.

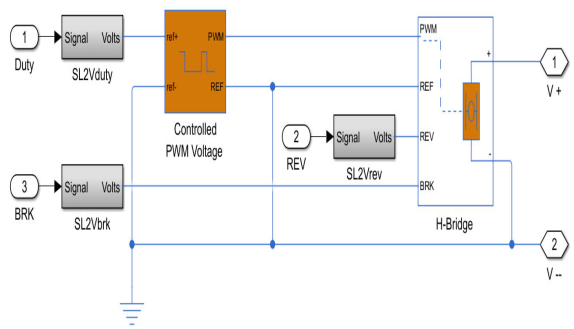

FIGURE 8. Power amplifier and DC motors drive system.

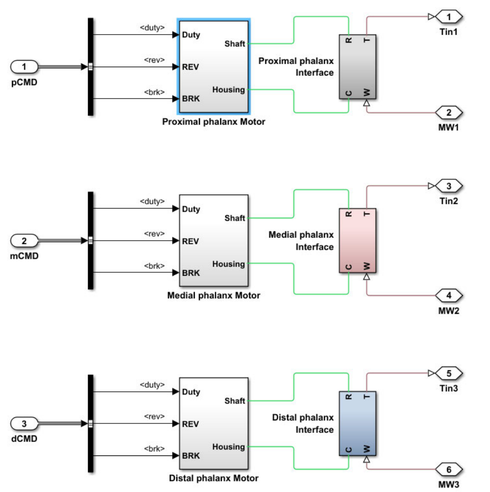

FIGURE 9. DC motor model block diagram and its interfaces.

To drive the DC motors, a power amplifier block is considered including a PWM generator and a H-Bridge block to drive the motor. Figure 7 and 8 represent the DC motors and the drive system block diagram. Figure 9 provides an overview of the DC motor models and their interfaces with the fingers' mechanical model, illustrating how they receive angular velocity feedback signals and sense the generated torque.

B. MODELLING THE DLR-HIT II IN CoppeliaSim

To validate the modeling work, it is required to plan an object grasping process for the DLR-HIT II robotic hand. Thus, CoppeliaSim is more convenient to perform this task. Regarding modeling the robotic hand in CoppeliaSim, the work was more straightforward as it does not have the limitation of MATLAB which has a matrix basis. Modeling the robotic hand had fewer challenges in CoppeliaSim in comparison with MATLAB/Simulink as some features like joint dependency mode are defined in the software in which the feature is employed for the 4th and 3rd joints in each finger. Like modeling the robotic hand in MATLAB/Simulink, the Palm coordinate frame is placed at the same position as the world frame, and the fingers' frames are matched with the DH model. CoppeliaSim requires

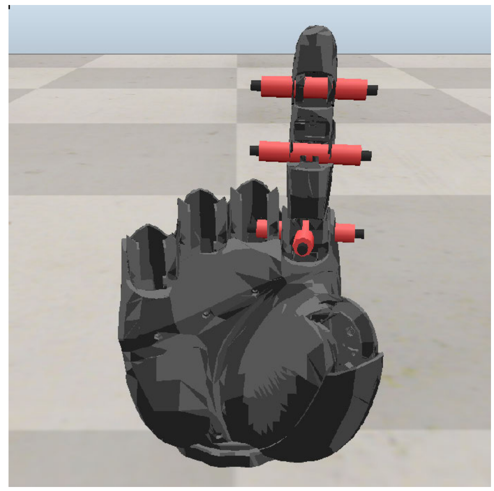

FIGURE 10. The structure of the joint and links of the DLR-HIT II in CoppeliaSim.

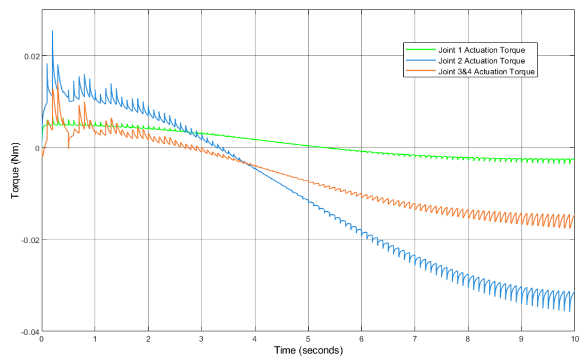

FIGURE 11. Computed torques for the thumb finger.

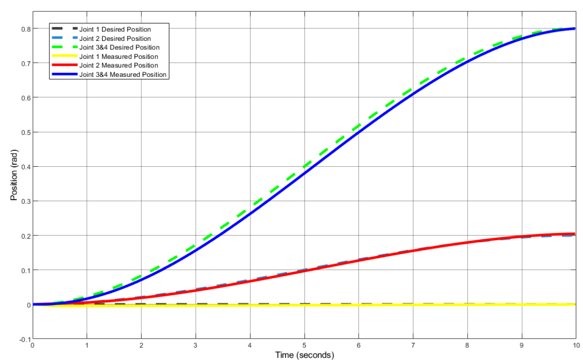

FIGURE 12. Compared the actual joints position with the desired joints positions for the thumb finger.

convex or pure shapes of the links to perform dynamically accurate movements, and response to the other objects. Therefore, to reserve both the appearance and the dynamical accuracy of the robotic hand, non-visible convex shapes are used in the background, and similar shapes of the robotic hand are used at the top level of visibility. One of the robotic hand's fingers and its joints' structure is shown in Figure 10.

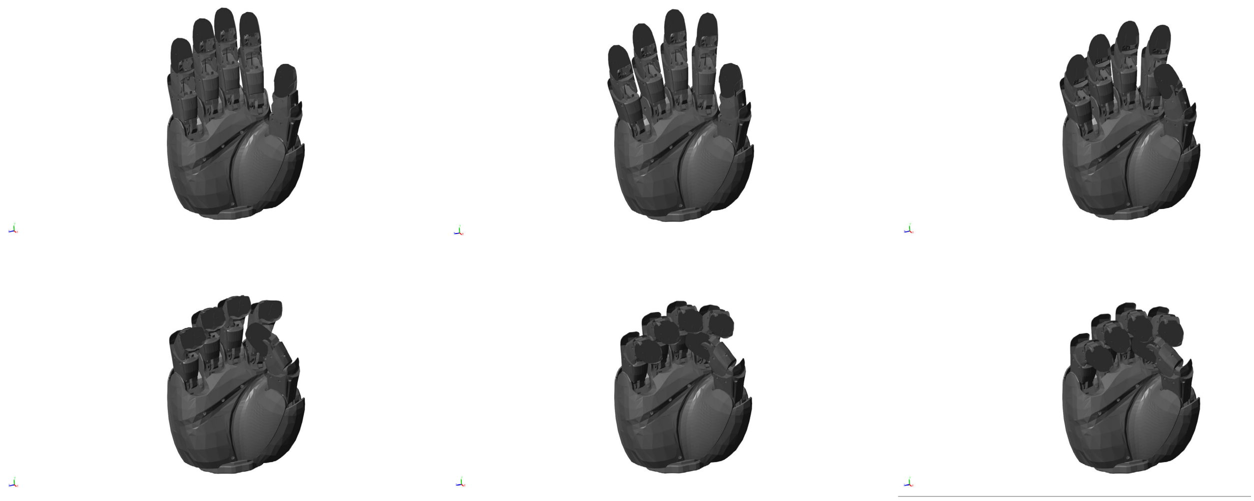

FIGURE 13. Demonstrates the movement of the modeled hand for 10s using PD motion base control in Matlab platform.

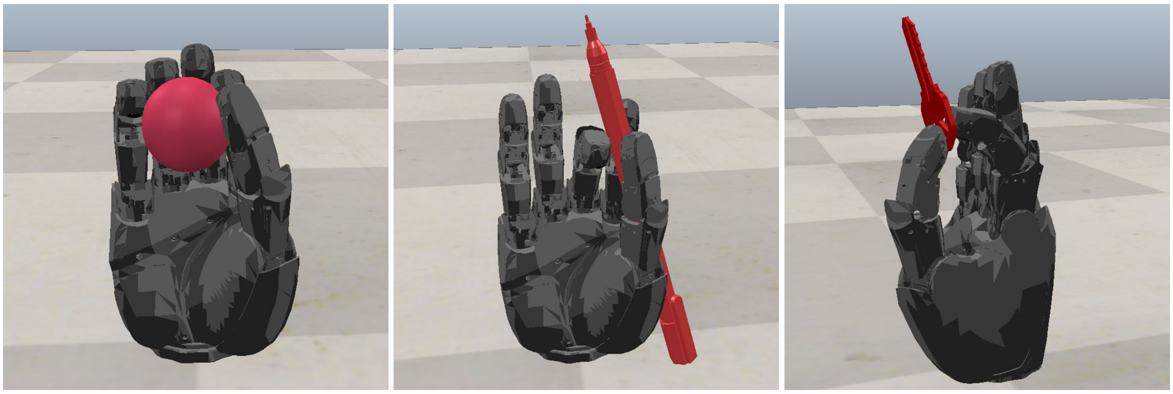

FIGURE 14. Shows the grasping of the hand for different objects in CoppeliaSim.

The Lua script is used within CoppeliaSim to simulate the grasping process of a robotic hand. This process involves several key steps: setting up the environment, configuring the inverse kinematics, moving the robotic arm to the end-effector's desired position, and grasping the target object. Maximum torque and movement parameters (velocity, acceleration, jerk) for both forward kinematics and inverse kinematics are defined to control the joints' behaviour. This simulation approach allows for accurate and efficient testing and validation of robotic grasping behaviours in a real-time controlled virtual environment.

IV. RESULTS AND DISCUSSION

The DLR-HIT robotic hand's geometric shape and structure were effectively modelled by the simulation. The simulation model accurately represented the intricate structure of the hand, which comprises its articulated fingers, joints, motors, and links. To ensure realistic dynamics and interactions, every part of the hand is constructed as a rigid body with the proper mass, measurements, and inertial features. A main challenge confronted through the modelling process was simulating the mechanical coupling, which is designed using a metal cable driven with only one motor, connecting the final two joints in every finger. Because of this coupling, variations in one joint's position would have an impact on the other joint's position, making it challenging to precisely capture the dynamics and movements among the linked joints. The robotic hand's movement and manipulation abilities were verified efficiently by the PD computed torque control scheme that is used in the MATLAB's simulation. The control algorithm made it possible for the fingers to move smoothly and precisely by calculating the desired position relying on the desired joint positions and velocities. Figure 11 shows the torques for the motors in thump finger, and Figure 12 illustrate the tracking of actual joints position to the desired joints position. The gains of the PD controller were thoroughly adjusted to provide responsive and stable performance under a variety of operating scenarios. The results of the simulation confirm that the geometry and motion of the DLR-HIT robotic hand can be accurately modelled and simulated using the Simscape Multibody toolbox in MATLAB. The hand's motion can be precisely and quickly controlled due to the integration of PD motion base control as shown in Figure 13. Moreover, The DLR-HIT II hand is simulated utilizing CoppiliaSim software. The challenge of joints coupling is solved by changing the mode of last joint of each finger of the artificial hand to depended mode. The robotic hand is tested using foreword and inverse kinematic control to grasp various objects as illustrated in Figure 14. As mentioned earlier in this paper, The hand is simulated in both programs because MATLAB provides advanced machine learning control algorithms and a robust control system, allowing for initial testing. The model is then validated in CoppeliaSim, which offers better visualization and a real-time interaction compared to MATLAB. The simulation represents insightful information about the physical structure and operation of the DLR-HIT robotic hand. The simulation model yields a valuable knowledge into the DLR-HIT robotic hand's structure and control, enabling additional improvement for real-world robotics and automation applications. To further enhance the performance and versatility of the hand, future work might concentrate on extending the simulated design with more features like adaptive control methods and interaction between humans and robots' capabilities.

V. CONCLUSION

This study illustrates an accurate modelling of the mechanical structure and control aspects of the robotic hand with the use of CoppeliaSim and MATLAB's Simscape Multibody toolbox.The simulation model accurately captures the mechanical structure of the robotic hand and resolves the issue of mechanical coupling between the last two joints of each finger. The simulated DLR-HIT II robotic hand can perform grasping and manipulation tasks with flexibility and dexterity. First, the robotic hand is modelled in MATLAB employing computed torque control to evaluate its movement in joint space. After that, the model is validated using the forward and inverse kinematics control for grasping various objects in CoppeliaSim. The rationale behind utilizing two applications for modelling the hand is twofold. CoppeliaSim provides real-time interaction and superior visualization compared to MATLAB, while MATLAB offers advanced machine learning control algorithms and a robust control system for preliminary testing. Additionally, simulating the artificial hand on different platforms allows for a more precise analysis of its motion and a deeper understanding of its interaction with the environment. The simulation experiments used to validate the DLR-HIT II hand's performance offer invaluable data for enhancing the robotic hand's functional and structural parameters. Consequently, the hand's flexibility and effectiveness can be improved in real-world conditions by constantly enhancing its grasping techniques, motion planning algorithms, and interaction dynamics through the simulation. The repetitious procedure of simulation-based design improvement not only facilitate the development of robotic system but also reduces the hazards and expenses corresponding with actual model. The outcome of this work identifies the limitations and potential of the DLR-HIT II robotic hand and establishes a standard for more challenging robotics investigations such the application of deep learning control methods, automated decision-making processes, manipulating the robot in complex environments, and collaboration between humans and robots.

ACKNOWLEDGMENT

The authors express their sincere gratitude to the Iraqi Ministry of Higher Education for their support. They also extend their appreciation to Swansea University, particularly to Prof. Johann Sienz, the Deputy Executive Dean of Science and Engineering, for providing essential facilities, workspace, and financial support.

REFERENCES

[1] P. M. Bhatt, R. K. Malhan, A. V. Shembekar, Y. J. Yoon, and S. K. Gupta, "Expanding capabilities of additive manufacturing through use of robotics technologies: A survey," *Additive Manuf.*, vol. 31, Jan. 2020, Art. no. 100933.

[2] Y. Minami, H. Okajima, K. Sawada, and K. Sekiguchi, "Special issue on navigation and control technologies for autonomous mobility," *J. Robot. Mechatronics*, vol. 35, no. 2, pp. 229–230, Apr. 2023.

[3] E. Schneiders, A. M. Kanstrup, J. Kjeldskov, and M. B. Skov, "Domestic robots and the dream of automation: Understanding human interaction and intervention," in *Proc. CHI Conf. Human Factors Comput. Syst.*, May 2021, pp. 1–13.

[4] M. Gharbia, A. Chang-Richards, Y. Lu, R. Y. Zhong, and H. Li, "Robotic technologies for on-site building construction: A systematic review," *J. Building Eng.*, vol. 32, Nov. 2020, Art. no. 101584.

[5] V. Marinoudi, C. G. Sørensen, S. Pearson, and D. Bochtis, "Robotics and labour in agriculture. A context consideration," *Biosystems Eng.*, vol. 184, pp. 111–121, Aug. 2019.

[6] I. V. Tishchenko and Y. V. Vanag, "Automation and robotization of solid mineral mining," *Interexpo GEO-Siberia*, vol. 2, no. 3, pp. 325–333, May 2022.

[7] M. Yip, S. Salcudean, K. Goldberg, K. Althoefer, A. Menciassi, J. D. Opfermann, A. Krieger, K. Swaminathan, C. J. Walsh, H. Huang, and I.-C. Lee, "Artificial intelligence meets medical robotics," *Science*, vol. 381, no. 6654, pp. 141–146, Jul. 2023.

[8] C. Gnanaprakasam, M. Swarna, R. Geetha, G. Saranya, and S. Murugan K H, "A novel design of smart and intelligent soldier supportive wireless robot for military operations," in *Proc. Int. Conf. Adv. Comput., Commun. Appl. Informat. (ACCAI)*, May 2023, pp. 1–7.

[9] D. T. Long, T. V. Binh, R. V. Hoa, L. V. Anh, and N. V. Toan, "Robotic arm simulation by using Matlab and robotics toolbox for industry application," *Int. J. Electron. Commun. Eng.*, vol. 7, no. 10, pp. 1–4, Oct. 2020.

[10] G. Xiao, "Research on artificial intelligence robots and simulation software in virtual training pendant," *J. Phys., Conf. Ser.*, vol. 1952, no. 4, Jun. 2021, Art. no. 042042.

[11] Z. Wang, "Real-time debugging and testing a control system using Matlab," *Open J. Appl. Sci.*, vol. 3, no. 2, pp. 61–64, 2013.

[12] H. Liu, K. Wu, P. Meusel, N. Seitz, G. Hirzinger, M. H. Jin, Y. W. Liu, S. W. Fan, T. Lan, and Z. P. Chen, "Multisensory five-finger dexterous hand: The DLR/HIT hand II," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2008, pp. 3692–3697.

[13] MathWorks, Inc, Natick, MA, USA. (2024). *Simscape Multibody Toolbox Documantaion*. Accessed: May 07, 2024. [Online]. Available: https://uk.mathworks.com/help/sm/getting-started-with-simmechanics.html

[14] E. Rohmer, S. P. N. Singh, and M. Freese, "V-Rep: A versatile and scalable robot simulation framework," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 1321–1326.

[15] A. Lamikiz, L. N. López De Lacalle, O. Ocerin, D. Díez, and E. Maidagan, "The denavit and hartenberg approach applied to evaluate the consequences in the tool tip position of geometrical errors in five-axis milling centres," *Int. J. Adv. Manuf. Technol.*, vol. 37, nos. 1–2, pp. 122–139, Apr. 2008.
[16] C. Li, A. Fahmy, and J. Sienz, "Development of a neural network-based control system for the DLR-HIT II robot hand using leap motion," *IEEE Access*, vol. 7, pp. 136914–136923, 2019.
[17] E. Arruda. (2019). *Eaa3/dlr_hit_hand_ii*.GitHub. Accessed: Apr. 25, 2024. [Online]. Available: https://github.com/eaa3/dlr_hit_hand_ii
[18] J. Butterfass, M. Grebenstein, H. Liu, and G. Hirzinger, "DLR-hand II: Next generation of a dextrous robot hand," in *Proc. ICRA. IEEE Int. Conf. Robot. Autom.*, vol. 1, May 2001, pp. 109–114.
[19] MathWorks, Inc. *Universal Joint*. Accessed: May 07, 2024. [Online]. Available: https://www.mathworks.com/help/stats/index.html
[20] MathWorks, Inc. *Multi-Loop PI Control of a Robotic Arm*. Accessed: May 07, 2024. [Online]. Available: https://www.mathworks.com/help/stats/index.html

ALI AL-SHAHRABI received the B.Eng. degree (Hons.) in control and system engineering from the University of Technology, Baghdad, Iraq, in 2007, and the M.Sc. degree in mechatronics and robotic technology engineering from National Research University (MPEI), Moscow, Russia, in 2017. He is currently pursuing the Ph.D. degree in mechanical engineering with Swansea University. He is currently an Assistance Lecturer with the College of Engineering, Al-Nahrain University. He is also a Demonstrator with Swansea University. His research interests include robotic systems, deep learning, control systems, and data mining.

MASOUD JAHANBAKHSH JAVID received the B.Sc. degree in mechanical engineering from Islamic Azad University, Tehran, Iran, in 2019, and the M.Sc. degree (Hons.) in mechanical engineering from Swansea University, Swansea, U.K., in 2024. He is currently researching robotic systems simulation and control with Swansea University. His research interests include robotic systems, deep learning, control systems, and data mining.

ASHRAF A. FAHMY received the B.Eng. degree (Hons.) in electrical engineering and the M.Sc. degree in flux vector control of electric machines from Helwan University, Cairo, Egypt, in 1992 and 1999, respectively, and the Ph.D. degree in neuro-fuzzy control of robotic manipulators from Cardiff University, U.K., in 2005. He has over 28 years of expertise in soft computing decision-making, data mining and machine learning, manufacturing systems, robotic manipulation, soft robotics, cybernetics, control systems engineering, electric drives, and electrical power engineering. He is currently a Senior Lecturer with the Faculty of Science and Engineering, Swansea University. He is also an Electrical Power and Machines Control Engineer by education, robotics and control engineer by research, and a manufacturing systems consultant by U.K. and MENA region experience.

CHRISTIAN A. GRIFFITHS (Senior Member, IEEE) received the Ph.D. degree in mechanical engineering from Cardiff University, Cardiff, U.K., in 2009. He is currently an Associate Professor with the College of Engineering, Swansea University. His research interests include advanced manufacturing, micro and nano technology, robotics and automation, 3D printing, and novel condition monitoring techniques. He has 16 years experience in the automotive industry and has authored or co-authored more than 84 papers in refereed journals and conference presentations.

CHUNXU LI received the Ph.D. degree from Swansea University, in 2019. He was appointed as a Lecturer with the School of Engineering, Computing and Mathematics, University of Plymouth, in January 2020. He has published over 40 academic articles, 33 of which are SCI/EI indexed. He won the best student paper award and was shortlisted for the best paper award in IEEE international conferences. He has expertise in ROS, JAVA, C++, Python, and MATLAB with over eight years working experience on multiple robot platforms, e.g., KUKA iiwa, Baxter, NAO, and Universal. He is also an Associate Fellow of the Higher Education Academy. As the PI and the main co-I, he successfully granted a number of projects and fundings.