Hand Gesture Interface to Teach an Industrial Robots

Mojtaba Ahmadieh Khanesar[a] and David Branson
Faculty of Engineering, NG7 2RD, Nottingham, U.K.

Keywords: Image Recognition, MediaPipe, Gesture Recognition, Industrial Robot Control, Inverse Kinematics.

Abstract: The present paper proposes user gesture recognition to control industrial robots. To recognize hand gestures, MediaPipe software package and an RGB camera is used. The proposed communication approach is an easy and reliable approach to provide commands for industrial robots. The landmarks which are extracted by MediaPipe software package are used as the input to a gesture recognition software to detect hand gestures. Five different hand gestures are recognized by the proposed machine learning approach in this paper. Hand gestures are then translated to movement directions for the industrial robot. The corresponding joint angle updates are generated using damped least squares inverse kinematic approach to move the industrial robot in a plane. The motion behaviour of the industrial robot is simulated within V-REP simulation environment. It is observed that the hand gestures are communicated with high accuracy to the industrial robot and the industrial robot follows the movements accurately.

1 INTRODUCTION

According to ISO 8373:2021 human–robot interaction (HRI) is *information and action exchanges between human and robot to perform a task by means of a user interface* (Standardization 2021). With ever-increasing degree of flexibility within an industry 4.0 settings to produce highly customizable products, it is required to have a flexible shop floor (Burnap, Branson et al. 2019, Lakoju, Ajienka et al. 2021). Such a flexible shop floor may be obtained using robust machine learning approaches to train the factory elements. One of the dominant factory elements are industrial robots. It is highly desirable to train industrial robot for new recipes and procedure required for product changes. Various types of industrial robots programming approaches can be identified in industrial environment (Adamides and Edan 2023). Most industrial robots benefit from teaching pendant which benefits from arrow keys as well as programming interface to program industrial robots. Some tools to control the industrial robot within Cartesian space and joint angle space exist within a teaching pendant. Lead through training (Choi, Eakins et al. 2013, Sosa-Ceron, Gonzalez-Hernandez et al. 2022) of industrial robots may exist within teaching pendant options to program it. PC interfaces to train industrial robots through Python (Mysore and Tsb 2022), C++, and Matlab (Zhou, Xie et al. 2020) may be provided using their corresponding APIs. However, more convenient approaches to provide an intuitive human robot interface are highly appreciable.

Different human robot interface approaches are proposed to provide an intuitive interface between human and robot. In this paper gesture recognition because of its ease of learning is chosen to train industrial robots. To recognize hand landmarks, MediaPipe package is used. The landmarks gathered in real-time using a low-cost camera are further processed to identify hand gestures. Totally five hand gestures representing movements in four directions plus stop command are identified using the proposed approach. The commands are movement command which make the robot move in any of four directions on a plane at a constant speed. The gesture commands are then translated in terms of joint angle movements using damped least-squares inverse kinematics approach. It is observed that using this approach, it is possible to move industrial robots in four directions. The tracking error obtained using the proposed approach demonstrates that the reference command given by hand gesture is followed with high performance within the simulation environment.

[a] https://orcid.org/0000-0001-5583-7295

Khanesar, M. and Branson, D.
Hand Gesture Interface to Teach an Industrial Robots.
DOI: 10.5220/0012205200003543
In *Proceedings of the 20th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2023) - Volume 1*, pages 243-249
ISBN: 978-989-758-670-5; ISSN: 2184-2809

This paper is organized as follows: the forward kinematics of industrial robot is presented in Section II. Section III presents hand gesture recognition procedure. Experimental setups and results are provided in Section IV, and V, respectively. concluding remarks are presented in Section VI. The acknowledgement part and the references are provided in Sections VII and VIII.

2 INDUSTRIAL ROBOT FORWARD KINEMATICS AND INVERSE KINEMATICS

Forward kinematics is a function which operates on the robot joint angle and results in its position and orientation. Inverse kinematics is the reverse process to identify the joint angles of a robot from its position and orientation. Damped least squares approach (Boucher, Laliberté et al. 2019, Tang and Notash 2021) is a computational inverse kinematics approach used in this paper. This inverse kinematic approach with geometrical forward kinematics methods is summarized in this Section.

2.1 Industrial Robot Dh Parameters

Serial industrial robots are investigated in this paper. A serial industrial robot which benefits from *n* number of joints has *n+1* number of links connecting joints together. Each joint is identified with its local coordinate. Four geometrical parameters are used to describe the spatial relationship between successive link coordinate frames in a DH system (Li, Wu et al. 2016). The four parameters in a DH system which are presented in Fig. 1 are joint angle θ_i, link offset d_i, link length a_i, and link twist α_i.

- Joint angle θ_i: the angle between x_{i-1} and x_i axes about the z_{i-1} axis
- Link offset d_i: the distance from the origin of frame $(i-1)$ to the x_i axis along the z_{i-1} axis
- Link length a_i: is the distance between the z_{i-1} and z_{i-1} axis along the x_i axis; for intersecting axis is parallel to $z_{i-1} \times z_i$;
- Link twist α_i: is the angle between the z_{i-1} and z_i axes about the x_i axis.

[1] https://www.universal-robots.com/articles/ur/application-installation/dh-parameters-for-calculations-of-kinematics-and-dynamics/ (visited: 1/5/2022)

2.2 Industrial Robot DH Parameters

Joint angle measurements using the rotary encoder sensors on joint shafts are used as the input to industrial robot FK to express the Cartesian coordinates of robot within its 3D workspace. The link transformation matrix from the link *i-1* to the link *i* using its DH parameters depends on the corresponding joint angle of the industrial robot and its D-H parameters (Kufieta 2014, Sun, Cao et al. 2017).

$$ {}^{i-1}_{\ \ i}T = \begin{bmatrix} cq_i & -c\alpha_i sq_i & s\alpha_i sq_i & a_i cq_i \\ sq_i & c\alpha_i cq_i & -s\alpha_i cq_i & a_i sq_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1) $$

where $q_i's, i = 1, \ldots, 6$ represent the joint angle i, $\alpha_i's, i = 1, \ldots, 6$, $a_i's, i = 1, \ldots, 6$, and $d_i, i = 1, \ldots, 6$ present other DH parameters of robot (see Fig. 1). Furthermore, cq_i, $sq_i, c\alpha_i$, and $s\alpha_i, i = 1, \ldots 6$ represent $cos(q_i)$, $sin(q_i)$, $cos\alpha_i$, and $sin(\alpha_i), i = 1, \ldots, 6$, respectively. Overall robot transformation matrix in robot base coordinates is obtained as follows.

$$ T_e = {}^0_6T = {}^0_1T\,{}^1_2T\,{}^2_3T\,{}^3_4T\,{}^4_5T\,{}^5_6T \quad (2) $$

The values of α_i's is given as follows.

$$ \begin{gathered} \alpha_1 = \alpha_4 = -\alpha_5 = \frac{\pi}{2} \\ \alpha_2 = \alpha_3 = \alpha_6 = 0 \end{gathered} \quad (3) $$

The 3D end effector coordinates are obtained as follows, the numerical DH parameter values according to the robot manufacturer are as follows[1].

$$ \begin{gathered} d_1 = 0.08916m, a_2 = -0.425m, \\ a_3 = -0.392m, d_4 = 0.1092m, \\ d_5 = 0.0947m, d_6 = 0.0823m \end{gathered} \quad (4) $$

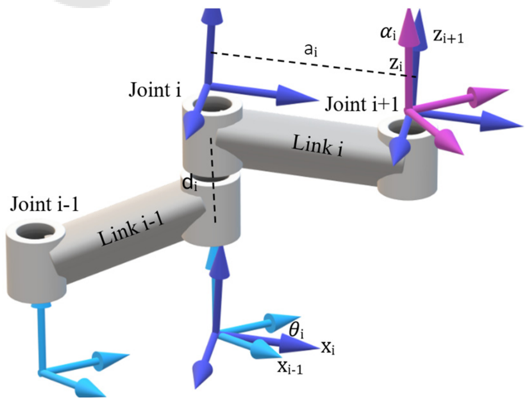

Figure 1: DH parameters.

Furthermore, $d_2 = d_3 = 0$, and $a_i = 0,\ i = 1,4,5,6$.

2.3 IK Model of UR5

To control an industrial robot, it is required to use inverse kinematic (IK) algorithm for the desired position and orientation of the robot to find its joint angle values. There exist different inverse kinematic approaches. Geometrical inverse kinematics damped least square approaches, and deep learning algorithms for IK are among top algorithms used to solve the inverse kinematic problems of industrial robots. Among these inverse kinematic approaches, DLS algorithm is widely used with V-REP software. This algorithm uses the pseudo inverse method to find the relationship between the cartesian space velocity and the increment in robot joint angles.

$$\dot{X} = J\Delta\theta \tag{5}$$

where $\dot{X}$ is the cartesian space velocity vector of the robot, $\Delta\theta$ is the joint angle increment in the robot joints, and J is the Jacobian of the industrial robot. To find the inverse kinematics equation, we have:

$$\Delta\theta = J^{\dagger}\dot{X} \tag{6}$$

where $J^{\dagger}$ is the pseudo-inverse of the Jacobian matrix of the system. Hence using DLS, the IK problem is formulated as a minimization problem as follows.

$$min_{\Delta\theta} \left\| \begin{bmatrix} J \\ \lambda I \end{bmatrix} \Delta\theta - \begin{bmatrix} \dot{X} \\ 0 \end{bmatrix} \right\| \tag{7}$$

The solution to this optimization problem is obtained as follows.

$$\Delta\theta = J^T (J^T J + \lambda^2 I)^{-1} \dot{X} \tag{8}$$

Using this equation, it is possible to calculate the required increment within the industrial robot joint angles to control the industrial robot towards its desired position. This algorithm is available as a built-in algorithm within V-REP and is utilized within this paper.

3 HAND GESTURE RECOGNITION

The preferred HRI method used in this paper is hand gesture recognition technique. This method is summarized in this section. To perform hand gesture recognition, OpenCV is used for image preprocessing such as filtering, and resizing. The filtered and resized hand image is then processed by MediaPipe software to extract hand landmarks. The angle of the hand is then extracted from the landmarks which is then used to detect hand gestures. The five hand gestures recognized by this approach are then used to move the industrial robot on a 2D plane. To translate the robot movements to robot joint angles DLS inverse kinematic approach as summarized in Section 2.3 is used.

3.1 Open-Source Computer Vision Library (OpenCV)

Open-source computer vision library (OpenCV) is a computer vision program originally developed in 1999. Further upgrades on this library occurred in 2009 as OpenCV2 and 2015 as OpenCV3 (Culjak, Abram et al. 2012). This software has been tested successfully tested under different operating systems including Windows, Linux, Mac, and ARM. OpenCV can be used within academic applications as well as industrial applications under BSD license. This package benefits from live communications with RGB and RGB-D cameras, processing images, pre-processing functions, processing functions, functions to add text and shapes to images, display options to display processed camera feed in real time, and image writing functions to save processed images as pictures. This package can also be used under C++, Python, and Matlab programming languages.

3.2 MediaPipe

MediaPipe is an open source perception framework for applied machine learning pipeline to process videos to detect some objects such as hands multiple hands, whole body, face and their landmark (Halder and Tayade 2021). This framework provides human body landmark detection, hand and finger position recognition, and face landmark recognition . To have a robust machine learning framework, MediaPipe is trained on the most diverse Google dataset. Landmarks in an image processed by MediaPipe are presented in terms of nodes on a graph generally specified in pbtxt file format. The normalized three-dimensional coordinates of these landmarks are given by MediaPipe. The hand landmark generated by MediaPipe is presented in Figure 2. Totally 21 landmarks are returned using MediaPipe. The names of the landmarks returned from MediaPipe are as follows. WRIST, THUMB_CMC, THUMB _MCP, THUMB_IP, THUMB_TIP, INDEX_FINGER_MCP, INDEX_FINGER_PIP, INDEX_FINGER_DIP, INDEX_FINGER_TIP, MIDDLE_FINGER_MCP, MIDDLE_FINGER_PIP,

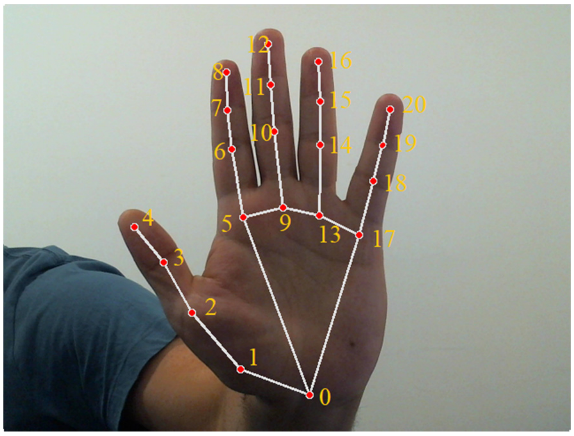

Figure 2: Landmarks identified by MediaPipe.

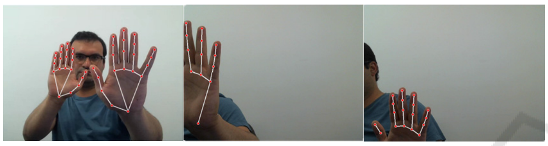

Figure 3: Landmarks identified by MediaPipe.

MIDDLE_FINGER_DIP, MIDDLE_FINGER_TIP, RING_FINGER_MCP, RING_FINGER_PIP, RING_FINGER_DIP, RING_FINGER_TIP, PINKY_MCP, PINKY_PIP, PINKY_DIP, and PINKY_TIP.

3.3 Hand Angle Recognition

The four landmarks of up, down, right, and left which are needed to be identified in this paper can be recognized by identifying the hand angle. The fifth hand gesture corresponds to the case when hand is closed. To recognize closed hand, a polynomial is fitted to the landmarks 0, 2, 5, 9 and 17. If landmarks 8, 12, 16 and 20 which present the fingertips fall within the boundaries of this polynomial, it means that the hand is closed. Otherwise, according to middle finger angle the respective gesture is recognized. To detect the angle of the middle finger, a polynomial is fitted to the landmark 9-12 which represent the middle finger. Then the angle of this polynomial is calculated using available command within Numpy package under Python. The hand gesture is recognized based on the angle of the middle finger.

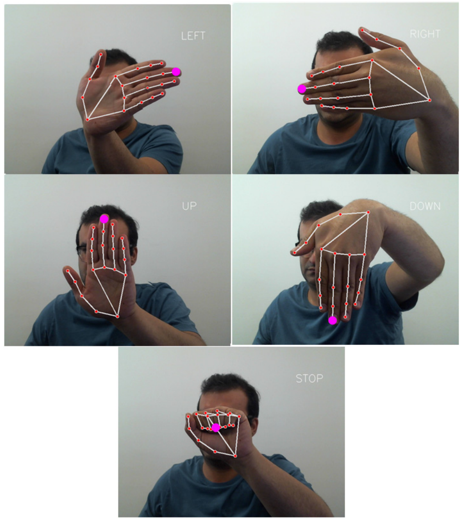

Figure 4: Five different gestures to control robot.

4 EXPERIMENTAL SETUP

4.1 Webcam

The webcam used in this experiment is a Logitech B525 HD Webcam with its resolution being equal to 640x480 pixels. It is capable of capturing images and streaming in real time. The frame rate of this camera is 30fps and can manually rotate 360 degrees. This camera is a low-cost camera which is used within experiments.

Table 1: Gesture commands and their interpretation.

Gesture	Interpretation
Up	Increment 1cm in z-direction
Down	Decrement 1cm in z-direction
Right	Increment 1cm in y-direction
Left	Decrement 1cm in y-direction
Stop	Stop

4.2 V-REP Simulation Environment

Among different robotic simulation software packages, V-REP is a general-purpose one developed by Coppelia Robotics. This software package is capable of working under different operating systems including Windows, Linux, and Mac (Nogueira 2014). It can be operated either using connectivity with robotic operating system (ROS) or by using APIs for Matlab and Python. Wide range of mobile robots as well as non-mobile robots including industrial robots, and components to provide a robotic environment are available under this software. It supports four physics engines including Bullet, ODE, Newton, and Vortex. Position control, velocity control, and torque control for industrial robots are supported within this software package. Joint angle control within joint space and control under Cartesian space is performed using built-in DLS as the preferred inverse kinematics method. In this paper using Python programming language under Spyder and Anaconda IDE, and the remote API, simulations are provided to test the proposed algorithm.

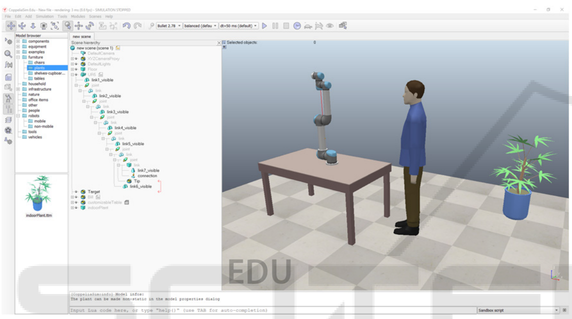

Figure 5: V-REP simulation environment.

Among different industrial robots available under VREP, UR5 is used in this paper. UR5 is a collaborative industrial robot which means that it can collaboratively work with a human being within a close proximity without extra safety measures such as cages. UR5 is manufactured by Universal Robots and is capable of handling up to 5Kg of load with maximum no-load angular velocity of $180°/sec$. The inverse kinematics model of this robot under V-REP environment is DLS. A brief description of this inverse kinematic model has already been presented in Section 2.3.

5 EXPERIMENTAL RESULTS

The webcam as explained in Section 4.1 is used to recognize hand gesture and control the robotic system. The hand gestures are then communicated with a simulated UR5 within V-REP environment. To test the efficacy of the hand gesture recognition algorithm, a dataset of five different hand gestures with different orientations are generated. The generated dataset is made available online under the github address: www.github.com/moji82/gestures. The number of pictures used to perform gesture recognition analysis is totally 8424 pictures. The result of applying the proposed angle estimation algorithm to recognize the four gestures are presented in Table 2. As can be seen from the table, although the gesture recognition has been performed with high performance, some misclassifications can still be observed.

After testing the performance of the proposed gesture recognition algorithm, the gestures are communicated to UR5 within VREP simulation environment. The finger landmark recognition is performed in Python using MediaPipe version 0.8.10.1. Then the algorithm which is presented in Section 3.3 is used to recognize the five different gestures. The total length of experiment is equal to 80 seconds. Figure 6 presents the gesture commands recognized by the proposed algorithm. The video associated with this gesture recognition is uploaded online in https://www.youtube.com/watch?v=CD1eNkpl1Hk. Figures 7, 8, and 9 present the reference commands generated according to the gesture commands given to the simulated robot. As can be seen from the figure the tracking performance of the industrial robot simulated within V-REP is satisfactory. The mean integral of absolute tracking error for the industrial robot is calculated numerically within the simulation environment and is equal to $1.78 \times 10^{-4} m$.

Table 2: Performance of the proposed gesture recognition algorithm.

Gesture	UP	DOWN	LEFT	RIGHT	STOP
Detection performance	99.02	99.84	99.95	99.60	98.48

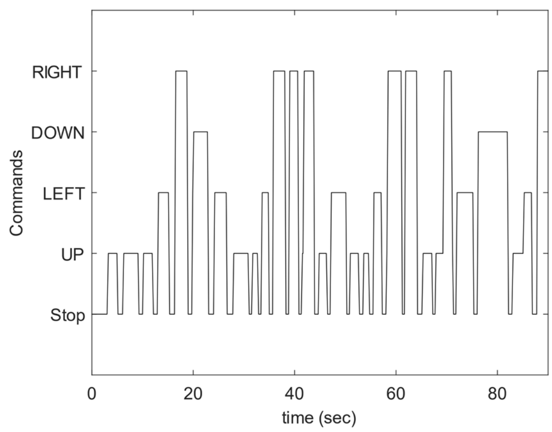

Figure 6: Gesture commands recognized by the proposed algorithm.

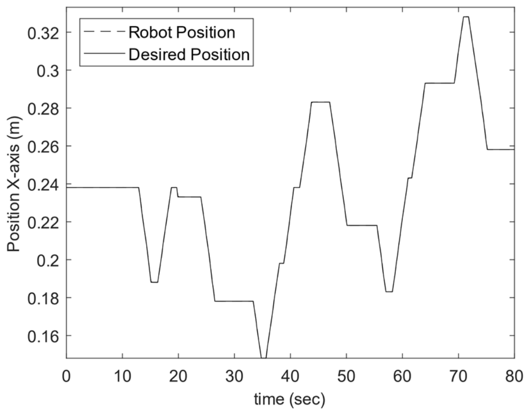

Figure 7: Robot movements in x-axis.

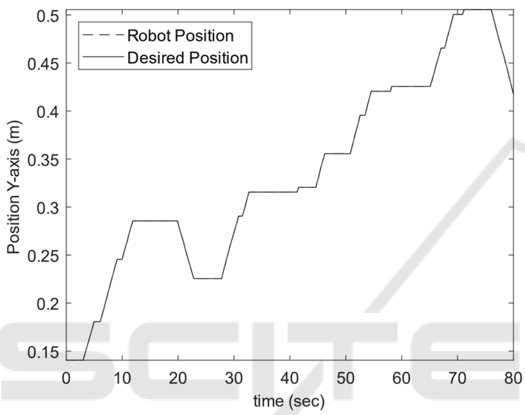

Figure 8: Robot movements in y-axis.

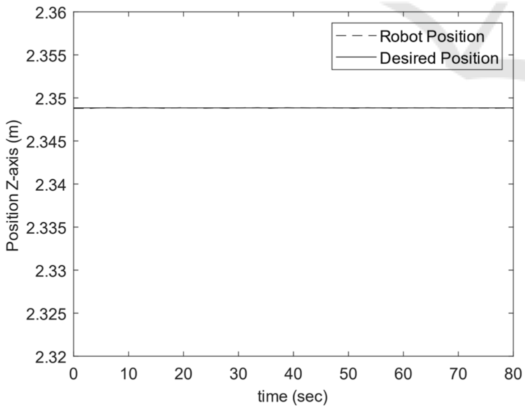

Figure 9: Robot movements in z-axis.

6 CONCLUSIONS

In this paper, hand gestures are used to control an industrial robot. To perform this task, MediaPipe is used to detect the features associated with human hand. These features include the place of different parts of finger. The angle of the middle finger is then used to detect four different hand gestures of up, down, right, and left. The closed hand status is recognize using the hand landmarks as well. The hand gesture recognition algorithm is highly accurate algorithm. It is observed that the gesture recognition is performed with high performance. Inspired by the results obtained from the gesture recognition, it is used to generate gesture commands which are then used to control UR5 industrial robot within V-REP environment. The python-VREP API is used to transfer the recognized gestures to the simulated robot within V-REP environment. Inverse kinematics algorithm of DLS is used to find the joint angle control of the industrial robot. Tracking performance of the proposed algorithm is investigated which demonstrate that the proposed gesture recognition and control approach are capable of controlling the industrial robot with high performance.

7 FUTURE WORKS

As a future work, more gestures are added to the gesture recognition library to make the robot move in the 3D dimensions and increase functionality. Deep neural networks will be used for detecting more gestures. Furthermore, the implementation of the proposed HRI on real industrial robot in real time will be considered.

ACKNOWLEDGEMENT

This research was funded by the Engineering and Physical Sciences Research Council (EPSRC)—projects EP/R021031/1—Chatty Factories.

REFERENCES

"Fast and flexible ML pipelines." Retrieved 27/04/2023, from https://developers.google.com/mediapipe.

"Google blog." Retrieved 27/04/2023, from https://developers.googleblog.com/.

"Hand landmarks detection guide." from https://developers.google.com/mediapipe/solutions/vision/hand_landmarker.

Adamides, G. and Y. Edan (2023). "Human–robot collaboration systems in agricultural tasks: A review and roadmap." Computers and Electronics in Agriculture 204: 107541.

Boucher, G., et al. (2019). A parallel low-impedance sensing approach for highly responsive physical human-robot

interaction. 2019 International Conference on Robotics and Automation (ICRA), IEEE.
Burnap, P., et al. (2019). "Chatty factories: A vision for the future of product design and manufacture with IoT."
Choi, S., et al. (2013). Lead-through robot teaching. 2013 IEEE Conference on Technologies for Practical Robot Applications (TePRA), IEEE.
Culjak, I., et al. (2012). A brief introduction to OpenCV. 2012 proceedings of the 35th international convention MIPRO, IEEE.
Halder, A. and A. Tayade (2021). "Real-time vernacular sign language recognition using mediapipe and machine learning." Journal homepage: www. ijrpr. com ISSN 2582: 7421.
Kufieta, K. (2014). "Force estimation in robotic manipulators: Modeling, simulation and experiments." Department of Engineering Cybernetics NTNU Norwegian University of Science and Technology.
Lakoju, M., et al. (2021). "Unsupervised learning for product use activity recognition: An exploratory study of a “chatty device”." Sensors 21(15): 4991.
Li, C., et al. (2016). "POE-based robot kinematic calibration using axis configuration space and the adjoint error model." IEEE Transactions on Robotics 32(5): 1264-1279.
Mysore, A. and S. Tsb (2022). "Robotic exploration algorithms in simulated environments with Python." Journal of Intelligent & Fuzzy Systems 43(2): 1897-1909.
Nogueira, L. (2014). "Comparative analysis between gazebo and v-rep robotic simulators." Seminario Interno de Cognicao Artificial-SICA 2014(5): 2.
Sosa-Ceron, A. D., et al. (2022). "Learning from Demonstrations in Human–Robot Collaborative Scenarios: A Survey." Robotics 11(6): 126.
Standardization, I. O. f. (2021). "ISO 8373: 2021 (en) Robotics—Vocabulary."
Sun, J.-D., et al. (2017). Analytical inverse kinematic solution using the DH method for a 6-DOF robot. 2017 14th international conference on ubiquitous robots and ambient intelligence (URAI), IEEE.
Tang, H. and L. Notash (2021). "Neural network-based transfer learning of manipulator inverse displacement analysis." *Journal of Mechanisms and Robotics* 13(3): 035004.
Zhou, D., et al. (2020). "A teaching method for the theory and application of robot kinematics based on MATLAB and V-REP." Computer Applications in Engineering Education 28(2): 239-253.