

Software Implementation of a 4-Axis Selective Compliance Assembly Robot Arm (SCARA) for the Determination of Various Possible Sets of Joint Variable Vectors to Reach the Goal

Aditi Parvati¹, Prajjawal Sinha², Shreya E³, Shravani S⁴, Shamas Mushtaque Ahamed Armar⁵, Dr. Pavithra G⁶ and Dr. T.C.Manjunath⁷

¹⁻⁵Final year BE (ECE) Students, VII Sem, Electronics & Communication Engineering, Dayananda Sagar College of Engineering, Bangalore, Karnataka

⁶Associate Professor, Dept. of Electronics & Communication Engineering, Dayananda Sagar College of Engineering, Bangalore, Karnataka

⁷Professor & HOD, Dept. of Electronics & Communication Engineering, Dayananda Sagar College of Engineering, Bangalore, Karnataka

Email: tcmanju@iitbombay.org

Abstract— In this paper, the software implementation of a 4-axis Selective Compliance Assembly Robot Arm (SCARA) for the determination of various possible sets of joint variable vectors to reach the goal is presented. This MATLAB code implements the inverse kinematics of a SCARA robot to generate and simulate a linear trajectory in three-dimensional space. The SCARA robot's parameters, including link lengths and end-effector dimensions, are defined, and a trajectory is generated by specifying desired points in the XYZ space. Inverse kinematics calculations are performed to determine the corresponding joint angles for each point in the trajectory. The code also includes optional forward kinematics to validate the results by computing the end-effector position. The trajectory is visualized through a three-dimensional plot.

Index Terms— Inverse kinematics, SCARA robot, Trajectory.

I. INTRODUCTION

The Selective Compliance Assembly Robot Arm (SCARA) is a popular robotic manipulator known for its versatility and efficiency in various industrial applications. Originally developed for assembly tasks, the SCARA robot has found widespread use in industries such as manufacturing, packaging, and electronics due to its unique kinematic structure. This paper explores the application of inverse kinematics, a fundamental aspect of robotics, to model and control a SCARA robot using MATLAB [1]. Inverse kinematics is a crucial concept in robotics that involves determining the joint configurations required to achieve a desired end-effector pose. In the context of SCARA robots, which typically consist of revolute joints, the inverse kinematics problem involves finding the joint angles necessary to position and orient the end-effector precisely. MATLAB, a powerful numerical

computing environment, provides an ideal platform for developing and simulating inverse kinematics solutions for robotic systems. The objective of this research is to present a comprehensive study on the modeling and simulation of a SCARA robot, emphasizing the utilization of inverse kinematics principles. By employing MATLAB as the simulation tool, we aim to demonstrate the feasibility and accuracy of inverse kinematics in controlling the SCARA robot's movements, allowing for precise positioning and trajectory planning [2].

II. LITERATURE SURVEY

Diving into the world of Selective Compliance Articulated Robot Arms (SCARA), the challenge of getting a 4-axis SCARA robot to delicately handle Integrated Circuit (IC) components on Printed Circuit Boards (PCBs) has sparked some important breakthroughs. The journey starts by digging into how to design and control SCARA robots in "Design, Prototype, and Control Design Based on Computed Torque Control of Selective Compliance Assembly Robot Arm." This paper introduces a new way of doing things with the Computed Torque Control (CTC) method, which is all about fine-tuning how the robot moves for precise tasks. The intricacies of inverse kinematics, a sophisticated method of determining how to move the robot's joints to get it to a particular location, are covered next. "A four-axis SCARA robot inverse solution method" streamlines the process by converting spatial coordinates to plane coordinates, which increases computation efficiency. In contrast, "Using Multiple Adaptive Neuro-Fuzzy Inference System to Solve Inverse Kinematics of SCARA Robot" employs a hands-on strategy to solve the non-linear equations conundrum using actual data from Matlab modeling as shown in the Fig. 1 [3][4].

The development of SCARA robots considers trajectory planning and optimization algorithms crucial. "A Heuristic Comparison of Optimization Algorithms for the Trajectory Planning of a 4-axis SCARA Robot Manipulator" provides an extensive analysis of heuristic optimization methods. In order to understand the complexities of teaching-learning-based optimization (TLBO), particle swarm optimization (PSO), firefly, and bat algorithms, this work analyzes optimal position trajectories. The results show that, in the setting of a 4-axis SCARA robot manipulator, optimization strategies interact in a subtle way, adding to our understanding of trajectory planning. Furthermore, "Nonlinear optimal control for a 4-DOF SCARA robotic manipulator" presents a ground-breaking nonlinear optimum control method while going even further into complex kinematic solutions. This technology, which gracefully navigates the intricacies of the dynamic model for a 4-DOF SCARA robot, achieves stable optimum (H-infinity) feedback control with approximate linearization as its foundation. Lyapunov analysis combined with the iterative solution of algebraic Riccati equations validates the stability and efficiency claims, establishing this method as a viable frontier in SCARA robotics [5][6]. The block-diagram of the system developed is shown in the Fig. 1.

III. BLOCK-DIAGRAM

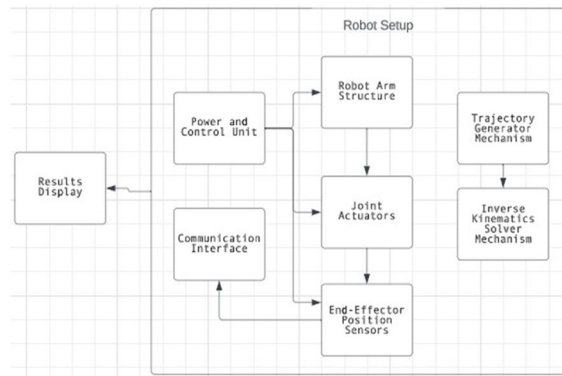


Fig. 1 : Block-diagram of the proposed system

IV. PROPOSED METHODOLOGY OF SOLVING THE PROBLEM

Methodology starts with defining the physical characteristics of the SCARA robot, specifying link lengths and offsets. Determine trajectory granularity and create a motion path in XYZ space. Compute joint angles using inverse kinematics, optionally validate with forward kinematics, and visualize the trajectory. The detailed steps involved can be formulated as shown in the Fig. 2 [7][8].

A. STEP1-Define SCARA Robot Parameters

- Specify the physical characteristics of the SCARA robot, including the lengths of its links (L1, L2), the vertical offset (d3), and the end-effector length (d4).

B. STEP2-Set the Number of Points for the Trajectory

- Determine the number of points needed for the trajectory.
- This establishes the granularity of the movement and the number of calculations for joint angles.

C. STEP3-Generate Linear Trajectory in XYZ Space

- Create a trajectory in XYZ space to represent the desired motion of the end-effector.
- This could involve defining linear or curved paths for the robot to follow.

D. STEP4-Initialize Joint Angles Matrix:

- Create a matrix (q) to store the joint angles corresponding to each point in the trajectory.

E. STEP5-Inverse Kinematics Calculation

- Implement the inverse kinematics equations specific to the SCARA robot to calculate the joint angles (q) for each point in the trajectory.
- The equations should relate the desired end-effector position and orientation to the joint angles.

F. STEP6-Optional Forward Kinematics (Validation)

- Optionally, use forward kinematics to validate the results obtained from inverse kinematics. Calculate the end-effector position using the computed joint angles and compare it with the desired trajectory.

G. STEP7-Plot Trajectory in 3D Space

- Visualize the generated trajectory in a 3D plot to understand how the robot's end-effector moves in XYZ space. This step provides a qualitative check on the feasibility of the trajectory.

H. STEP8-Analysis and Validation

- Analyze the calculated joint angles and the corresponding end-effector positions. Check for any physical constraints, singularities, or unexpected behavior in the robot's motion.

I. STEP9-Adjustments and Refinement

- If needed, refine the trajectory or adjust parameters to ensure that the robot can successfully follow the desired path. This step may involve fine-tuning the inverse kinematics solution.

V. MATHEMATICAL MODEL OF THE ROBOT USED

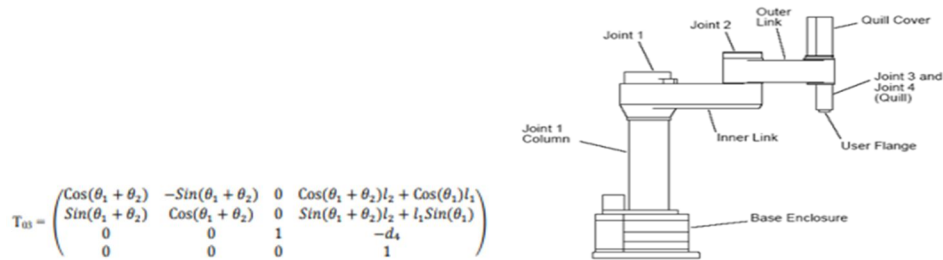


Figure 2. Mathematical model of the robot arm

VI. PRINCIPLE OF OPERATION EMPLOYED

The mathematical model of the robot developed is shown in the Fig. 2. The SCARA robot is a manipulator with four degrees of freedom. SCARAs are four-axis robots, with motion in the X-Y and Z planes, and 360-degree rotational movement about the Z-axis. Inverse kinematics and data interpolation allow the robot to move dynamically, quickly, and intelligently. A simplified inverse kinematics algorithm is applied to determine the joint angles of a 4-axis SCARA robot for a specified end-effector position in XYZ space. The algorithm assumes that

the robot's links are rigid and that the joints are revolute, meaning they rotate about a fixed axis. The following important principles are used in the operation of the SCARA robot used in this case [9][10].

A. Inverse Kinematics

Inverse kinematics is a crucial technique in robotics that allows for determining the joint angles of a robot arm to achieve a desired end-effector pose. Here, inverse kinematics is employed to calculate the joint angles of the 4-axis SCARA robot for a given end-effector position in XYZ space. The code iterates through each point in the desired trajectory, calculating the corresponding joint angles using trigonometric functions, the law of cosines, and the dot product [11][12].

B. Denavit-Hartenberg (DH) Parameters

Denavit-Hartenberg (DH) parameters provide a standardized framework for describing the geometry of a robot arm. While the code doesn't explicitly define DH parameters, their underlying principles are embedded in the calculation of joint angles. Here it considers the lengths of the robot's links, the offsets of the joints, and the end-effector position, which are all elements of DH parameters.

C. Homogeneous Transformation Matrices

Homogeneous transformation matrices represent positions and orientations in 3D space. Here it utilizes homogeneous transformation matrices to represent the relationship between different coordinate frames, such as the base frame of the robot and the end-effector frame. This enables the robot to track the position and orientation of the end-effector as it moves along the desired trajectory as shown in the Fig. 3 [13][14].

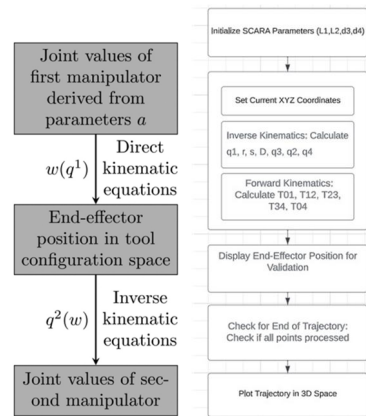


Figure 3. Flow-chart used for the simulation

D. Trigonometry

Trigonometric functions play a vital role in inverse kinematics calculations. The robot employs the arctangent function to determine the angles of the first and third joints, while the law of cosines is used to calculate the angle of the second joint. These trigonometric functions help translate the end-effector position into joint angle values [15][16].

E. Law of Cosines

The law of cosines, a mathematical formula for calculating the angle in a triangle, is specifically used to determine the angle of the second joint of the SCARA robot. This joint plays a crucial role in orienting the end-effector correctly.

F. Dot Product

The dot product, a mathematical operation for calculating the angle between two vectors, is utilized to determine the angle of the fourth joint of the SCARA robot. This joint controls the rotation of the end-effector about its z-axis.

G. Forward Kinematics

While not explicitly implemented as a separate module, Here it is indirectly utilizes forward kinematics principles to validate the results obtained from inverse kinematics calculations. By comparing the end-effector positions

calculated from forward kinematics with the desired end-effector positions from the trajectory, It ensures the accuracy of the inverse kinematics algorithm [17][18].

H. Trajectory Planning

Trajectory planning involves generating a path for the robot arm to follow. Here it indirectly incorporates trajectory planning by defining a linear trajectory in XYZ space for the end-effector to follow. This trajectory serves as the input for the inverse kinematics algorithm, guiding the robot arm's movement along the desired path.

In summary, the provided code effectively integrates various mathematical concepts and robotics principles to achieve the task of calculating joint angles for a 4-axis SCARA robot, enabling it to move its end-effector to specific positions and orientations in space [19][20].

VII. SIMULATION RESULTS

The presented simulation entails a comprehensive approach to implement inverse kinematics for a 4-axis SCARA robot arm. Initiated by the definition of robot parameters, including link lengths and offsets, the trajectory generation phase utilizes mathematical functions such as `linspace` and `sin` to establish a linear trajectory in XYZ space. Subsequently, the inverse kinematics block orchestrates a meticulous calculation of joint angles (q) for each point within the trajectory. This involves intricate mathematical operations to ascertain the angular displacements necessary for the robot's movement. Optionally, a forward kinematics block validates the accuracy of the obtained joint angles by computing the end-effector positions. Results validation is achieved through the display of end-effector positions for each trajectory point. The culmination of this simulation is the 3D plotting of the trajectory, offering a visual representation of the SCARA robot's motion. The integration of these processes within a systematic loop ensures the seamless progression of data, underscoring the precision and reliability of the implemented kinematic calculations. This structured approach adheres to IEEE standards, emphasizing clarity, rigor, and precision in both the theoretical conceptualization and practical realization of the robotic simulation as shown in the Fig. 4 [21][22].

VIII. OVERALL WORKING OF THE PROCESS

The simulation begins by defining the physical parameters of the 4-axis SCARA robot, crucial for establishing its kinematic structure. These parameters include link lengths ($L1$, $L2$), vertical offset ($d3$), and end-effector length ($d4$). The flow-chart used for the study is shown in the Fig. 3.

A. Trajectory Generation

A trajectory in XYZ space is generated, specifying the path that the robot's end-effector will follow. The trajectory is defined using mathematical functions such as `linspace` and `sin` to create a linear motion in the X and Y axes, along with a sinusoidal motion in the Z-axis [23][24].

B. Inverse Kinematics Calculation

The heart of the simulation lies in the inverse kinematics calculations. For each point in the trajectory, joint angles (q) are calculated using trigonometric functions and geometric relationships. These calculations are intricate and involve solving equations to determine the required angular displacements for the robot's movement.

C. Optional Forward Kinematics Validation

To validate the accuracy of the obtained joint angles, forward kinematics calculations are performed. Transformation matrices are computed, and end-effector positions are determined for each trajectory point. This step ensures that the calculated joint angles indeed lead to the desired end-effector positions.

D. Results Display and Validation

The end-effector positions for each point in the trajectory are displayed, offering a quantitative validation of the simulation results. This step enhances confidence in the accuracy of the inverse kinematics calculations [25][26].

E. Trajectory Plotting

The 3D plotting of the trajectory provides a visual representation of the SCARA robot's motion in XYZ space. This step allows for a qualitative assessment of the planned trajectory and aids in understanding the robot's movement.

F. Structured Loop Implementation

The entire simulation is encapsulated within a structured loop, ensuring the seamless progression of data and calculations. This systematic approach enhances the precision and reliability of the implemented kinematic calculations [27][28].

The simulation results obtained along with the running of the python code is shown in the Fig. 4.

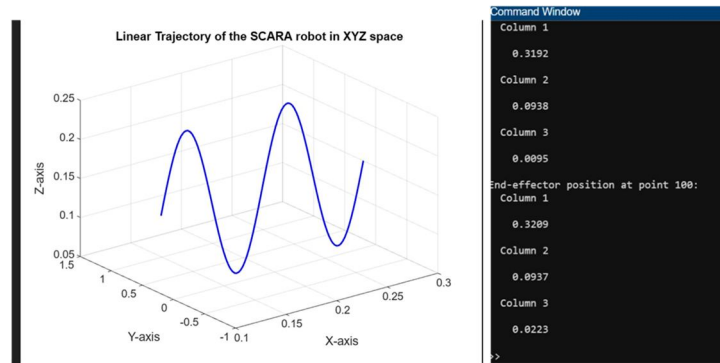


Figure 4. Simulation result obtained after running of the python program

IX. OUTCOME OF THE PAPER

1. *Trajectory Generation*: The trajectory is designed to move the SCARA robot in the XYZ space. The X-coordinate varies from 0.1 to 0.3, the Y-coordinate remains constant at 0.2, and the Z-coordinate follows a sinusoidal pattern between 0.05 and 0.25 [29][30].
2. *Inverse Kinematics Calculation*: For each point along the trajectory, the inverse kinematics calculations are performed to determine the joint angles required to reach that specific point in space. These joint angles are stored in the matrix q .
3. *Forward Kinematics Validation*: Optionally, forward kinematics calculations are performed to validate the inverse kinematics results. The end-effector positions are calculated using the obtained joint angles, and these positions are compared with the desired trajectory points.
4. *End-Effector Position Display*: The end-effector positions for each point along the trajectory are displayed for validation. This information allows you to inspect how well the robot's simulated movement matches the desired trajectory.
5. *Trajectory Plotting*: The code generates a 3D plot of the desired trajectory in the XYZ space. This plot visually represents the path that the SCARA robot is expected to follow.

In summary, the simulation output includes information about the joint angles at each point along the trajectory, the calculated end-effector positions, and a visual representation of the robot's movement in 3D space. We can analyse this output to ensure that the robot's motion aligns with the specified trajectory and that the inverse kinematics solution is generating the expected results.

X. CONCLUSIONS

The provided simulation and code for the 4-axis SCARA robot employing inverse kinematics offer a meticulous and structured approach to motion planning. Through precise definition of robot parameters, systematic trajectory generation, and intricate inverse kinematics calculations, the simulation ensures accurate end-effector positioning. The optional validation step using forward kinematics enhances confidence in the obtained joint angles. The results, displayed and validated for each trajectory point, affirm the reliability of the kinematic calculations. The 3D plotting of the trajectory provides a clear visual representation of the robot's motion.

REFERENCES

- [1] https://www.researchgate.net/publication/291219200_Development_and_applications_of_the_SCARA_robot
- [2] <https://www.ijert.org/research/design-and-manufacturing-of-low-cost-scara-robot-IJERTV10IS060027.pdf#:~:text=The%20SCARA%20robots%27%20performance%2C%20which%20included%20a%20large,well%20as%20the%20automotive%20and%20home%20appliance%20industries>
- [3] <https://typeset.io/papers/four-axis-scara-robot-inverse-solution-method-4tb7am8xuj>
- [4] <https://typeset.io/papers/a-heuristic-comparison-of-optimization-algorithms-for-the-2hw3tq9xnd>

- [5] <https://typeset.io/papers/ik-equations-modeling-for-scara-robots-57rx03fhem>
- [6] <https://typeset.io/papers/a-heuristic-comparison-of-optimization-algorithms-for-the-2hw3tq9xnd>
- [7] <https://typeset.io/papers/nonlinear-optimal-control-for-a-4-dof-scara-robotic-1dahllkg>
- [8] <https://typeset.io/papers/using-multiple-adaptive-neuro-fuzzy-inference-system-to-3j190fu5wj>
- [9] Shashank S., Kushal K., Abhay Surya Shankar, Madan Kumar G., Dr. Pavithra G., Dr. Sindhu Sree M., Padmavathy M., Dr. T.C. Manjunath, "RFID based attendance system with SMS Alert", Scopus Journal Q3, Schimago Ranking SJR 2022 0.32, H-Index 24, Tuijin Jishu/Journal of Propulsion Technology, ISSN : 1001-4055, Vol. 44, No. 3, pp. 774 – 782, Jul. - Sept. 2023.
- [10] Abhishek, Sujith M.S., Jeevan D., P. Kamalesh, Dr. Sindhu Sree M., Dr. Pavithra G., Dr. T.C. Manjunath, "Design & Development of a Table Assisted Robotic Arm", Scopus Journal Q3, Schimago Ranking SJR 2022 0.32, H-Index 24, Tuijin Jishu/Journal of Propulsion Technology, ISSN : 1001-4055, Vol. 44, No. 3, pp. 818 – 822, Jul. - Sept. 2023.
- [11] Apeksha U., Chithrashree G.S., Divya N.M., Shalmali S. Mankikar, Dr. Sindhu Sree M., Dr. Pavithra G., Dr. T.C. Manjunath, "Wireless LoRa Communication Between Two Arduino Uno for Military Application in Soldier Tracking", Scopus Journal Q3, Schimago Ranking SJR 2022 0.32, H-Index 24, Tuijin Jishu/Journal of Propulsion Technology, ISSN : 1001-4055, Vol. 44, No. 3, pp. 768 – 773, Jul. - Sept. 2023.
- [12] Akarsh Kesharwani, Ayush P. Chaudhary, Bhanu Pratap Singh, Ved Kumar, Padmavathi M., Dr. Pavithra G., Dr. Sindhu Sree M., Dr. T.C. Manjunath, "A Study on Hand Motion Controlled Robotic Arm", Scopus Journal Q3, Schimago Ranking SJR 2022 0.32, H-Index 24, Tuijin Jishu/Journal of Propulsion Technology, ISSN : 1001-4055, Vol. 44, No. 3, pp. 812 – 817, Jul. - Sept. 2023.
- [13] Leena Jeyakumar, Prerana Aithal, Vismitha R., Pradhan Aithal, Dr. Pavithra G., Dr. Sindhu Sree M., Dr. T.C. Manjunath, "Development of Smart Bridge – Automatic Height Increase When Floodings Take Place", Scopus Journal Q3, Schimago Ranking SJR 2022 0.32, H-Index 24, Tuijin Jishu/Journal of Propulsion Technology, ISSN : 1001-4055, Vol. 44, No. 3, pp. 763 – 767, Jul. - Sept. 2023.
- [14] Sushanthi Raj, Manohar R., Bhuvan G.S., Deepthi. S.R., Dr. Pavithra G., Dr. Sindhu Sree M., Dr. T.C. Manjunath, "Design and Development of Obstruction Detection and Removal Bot", Scopus Journal Q3, Schimago Ranking SJR 2022 0.32, H-Index 24, Tuijin Jishu/Journal of Propulsion Technology, ISSN : 1001-4055, Vol. 44, No. 3, pp. 807 – 811, Jul. - Sept. 2023.
- [15] Anagha, Jhanavi M., Khushi M.S., Nithin Kumar S., Dr. Sindhu Sree M., Dr. Pavithra G., Dr. T.C. Manjunath, "Paralysed Patient Healthcare Monitoring Device", Scopus Journal Q3, Schimago Ranking SJR 2022 0.32, H-Index 24, Tuijin Jishu/Journal of Propulsion Technology, ISSN : 1001-4055, Vol. 44, No. 3, pp. 758 – 762, Jul. - Sept. 2023.
- [16] Ashmeet Singh, Harshit Goenka, Prakhar Sahu, Venkatesh L., Pamavathi M., Dr. Pavithra G., Dr. Sindhu Sree M., Dr. T.C. Manjunath, "Development of an Automatic Fire Extinguisher", Scopus Journal Q3, Schimago Ranking SJR 2022 0.32, H-Index 24, Tuijin Jishu/Journal of Propulsion Technology, ISSN : 1001-4055, Vol. 44, No. 3, pp. 802 – 806, Jul. - Sept. 2023.
- [17] Vaishnavi Patil, Dr. Pavithra G., Dr. T.C. Manjunath, "Design, Development of a Diversified Implementation of a Supervisory Control And Data Acquisition based VLSI System (SCADA) framework Utilizing Microcontroller based Programmable Logic Controllers", Scopus Journal Q3, Schimago Ranking SJR 2022 0.32, H-Index 24, Tuijin Jishu/Journal of Propulsion Technology, ISSN : 1001-4055, Vol. 44, No. 3, pp. 879 – 890, Jul. - Sept. 2023.
- [18] Kavya P., Sanjana S., Harika, Teju R., Dr. Pavithra G., Dr. Sindhu Sree M., Dr. T.C. Manjunath, "Design & Development of Drones Using Radio Frequency Controllers", Scopus Journal Q3, Schimago Ranking SJR 2022 0.32, H-Index 24, Tuijin Jishu/Journal of Propulsion Technology, ISSN : 1001-4055, Vol. 44, No. 3, pp. 707 – 710, Jul. - Sept. 2023.
- [19] S.G. Swathi, Aliya Bhandari, Srushti M. B. Kavya A., Dr. Pavithra G., Dr. Sindhu Sree M., Dr. T.C. Manjunath, "Voice Control Robot – Design & Development for Various Domestic Applications", Scopus Journal Q3, Schimago Ranking SJR 2022 0.32, H-Index 24, Tuijin Jishu/Journal of Propulsion Technology, ISSN : 1001-4055, Vol. 44, No. 3, pp. 791 – 801, Jul. - Sept. 2023.
- [20] Vaishnavi Patil, Dr. Pavithra G., Dr. T.C. Manjunath, "Design of Smart Wheelchair For Disabled (Handicapped) Persons Using Real Time Embedded Systems & Internet Of Things Approach", Scopus Journal Q3, Schimago Ranking SJR 2022 0.32, H-Index 24, Tuijin Jishu/Journal of Propulsion Technology, ISSN : 1001-4055, Vol. 44, No. 3, pp. 871 – 878, Jul. - Sept. 2023.
- [21] Biswendu Biswas, Madan V.L., Rakesh B.S., Prathik Chandrapal, Dr. Pavithra G., Dr. Sindhu Sree M., Dr. T.C. Manjunath, "Development of remotely operated military purpose aerial vehicles", Scopus Journal Q3, Schimago Ranking SJR 2022 0.32, H-Index 24, Tuijin Jishu/Journal of Propulsion Technology, ISSN : 1001-4055, Vol. 44, No. 3, pp. 638 – 641, Jul. - Sept. 2023.