

# Color-Based Object Tracking Using Forward and Inverse Kinematics

Haoyang Li<sup>1,\*</sup>, Yangping Li<sup>2</sup> and Beichen Zhang<sup>3</sup>

<sup>1</sup> Beijing International Bilingual Academy Tianjin Campus, Tianjin, 300457, China

<sup>2</sup> Shanghai Guanghua Cambridge International School, Shanghai, 201319, China

<sup>3</sup> Changsha WES Academy, Changsha, 410100, China

\* Corresponding Author Email: 2024michaelli@biba-student.org

**Abstract.** Nowadays, the current surge in diverse applications of robotic arms makes comprehending these concepts of paramount importance. Various studies have been conducted on the topic of robotic arms. However, little attention was paid to the coupling relationship between object tracking and kinematics. Therefore, this research paper delves into the intricacies of robotic arm dynamics, with a specific focus on the topics of Forward and Inverse Kinematics, alongside object tracking facilitated by color-based methodologies to deepen foundational insights into robotic arm functionality. Controlling the motion and movement of robotic arms is the initial step. Leveraging the capabilities of cutting-edge technologies such as OpenCV, Armpi, and Hiwonder, it becomes feasible to precisely rotate the servos to user-input degrees and accurately position the arm based on three-dimensional Cartesian coordinates. Additionally, cameras equipped in the system enable photographing and recognizing target objects. By analyzing various positions of the target object, the robotic arm can dynamically adjust multiple servos to maintain the object's relative position. This tracking mechanism relies on the Proportional-Integral-Derivative algorithm, incorporating a minimum valid area parameter to mitigate interferences. Harnessing these algorithms empowers the robotic arm to discern objects based on camera-detected colors, and is affected slightly by different angles or distances, meaning the system's error margin consistently resides within a favorable interval. The approach exhibits practical utility in daily life scenarios, such as drug classification and recognition.

**Keywords:** Robotic arm; Forward and Inverse Kinematics deduction; Color Tracking; Route Optimization.

## 1. Introduction

In the contemporary landscape, the imperative of optimizing production efficiency resonates globally. Industries, encompassing mining, chemical manufacturing, and nuclear power generation, grapple with high-risk environments and potential hazards that endanger human well-being and disrupt operational productivity. Furthermore, the manufacturing and production sectors demand heightened operational efficiency. Precision assumes pivotal importance, especially in domains such as medicine, where surgical procedures necessitate meticulous accuracy. Nonetheless, sole reliance on human labor in these arenas proves insufficient. Consequently, the integration of mechanical arms emerges as a highly suitable solution for these fields. Mechanical arms excel in their capacity for automation, highly precise and stable movement, and durability and sturdiness, therefore effectively replacing human labor, reducing operational costs, and enhancing output and overall quality.

The initial phase of this research initiative involved an exhaustive review of pertinent academic literature and research papers. Presently, research endeavors actively explore the practical integration of robotic arms into digital manufacturing processes, including applications in milling machines and styrofoam cutting equipment [1]. The infusion of robotic arms into these operational contexts holds promise in addressing real-world challenges, notably by enhancing safety protocols and elevating precision levels.

Mathematical models for Cartesian robotic arm control, aided by piezoelectric controllers [2], simplify control using velocity-related signals, ensuring accuracy in displacement and force. However,

the experiments didn't validate the availability in real-life scenarios. Another avenue of high-precision robotic arm motion is contributed by Andrea Carron and colleagues through a database and data-driven error model [3]. Nonetheless, this database approach faces challenges in accommodating diverse real-world scenarios and relies on pre-collected data. Gathering this data can be time-consuming, and distinguishing between suitable and unsuitable data poses a significant challenge.

Color-based tracking methodologies have been thoroughly examined using the Monte Carlo tracking technique. This probabilistic approach has demonstrated success in diverse testing scenarios but has yet to find practical applications [4]. Another influential reference was the work of Hai-Wu Lee [5], whose paper presented an algorithmic framework aligned with the principles of kinematics proportionality. However, it employed the RGB value processing technique, which affects the precision, as RGB color ranges aren't continuous. Transformation of image data into the LAB color space was a strategic optimization approach.

Additional inspiration for the research methodology stemmed from the work of Tang, Shufeng, et al. [6]. This scholarly contribution provided valuable insights that contributed to the development of the system's operational framework. However, their work lacked a comprehensive tracking system, necessitating manual human control and oversight. Efforts were made to bridge this gap by conceiving and implementing a sophisticated system. This innovation effectively facilitates the autonomous movement of the mechanical arm, thereby enhancing its operational autonomy and versatility. Consequently, there exists a compelling imperative to redouble efforts in the pursuit of viable and pragmatic applications for robotic arm technology.

Conducting research and experiments on mechanical arms holds great academic significance. Through such technical innovation, the development of flexible and efficient automation solutions is facilitated, leading to improved productivity and resource utilization. Substituting human intervention in hazardous missions not only serves the purpose of ensuring human safety but also broadens the scope of human exploration and operations. Additionally, human-robot collaboration enables mechanical arms to seamlessly cooperate with humans in the execution of intricate tasks.

This research extensively examines the feasibility and algorithmic intricacies inherent in a mechanical arm grasping system. The primary focus lies in a comprehensive investigation of the kinematics aspects governing the actions of the robotic arm, concurrently cultivating foundational concepts and algorithms within the domain of computer vision.

The system demonstrates its feasibility and practicality, proficiently commanding the mechanical arm's movements with precision. However, real-world contingencies and environmental variables can influence the system's accuracy. Notably, color tracking algorithms, enhanced by the Proportional-Integral-Derivative (PID) algorithm, may suffer in suboptimal lighting conditions. Addressing this limitation necessitates further algorithmic refinements.

This paper centers on the movement and operation of a mechanical arm controlled by a Raspberry Pi, incorporating a color-tracking system through Python programming. This system allows precise servo control for accurate positioning. An algorithm converts coordinates into servo rotational angles, validated through Matlab simulations. The integration of the OpenCV library enables color tracking. Employing the PID algorithm reduces dataset errors and enhances efficiency. Image analysis transforms real images into the LAB color space for effective computer recognition. Contouring targets using RGB color values yields actual 3D coordinates. The mechanical arm adeptly manipulates and positions objects at the image center, facilitated by the developed control system, enabling precise object manipulation via color-tracking data.

## **2. Method and Process**

### **2.1. General Introduction**

This paper expounds on the intricacies of Forward Kinematics and Inverse Kinematics principles, coupled with object tracking through color-based methodologies. The methodology involves employing cameras integrated into the system for object tracking and guiding the robotic arm to reach

specified three-dimensional Cartesian coordinates. The entire process adheres to a well-defined flow chart Fig. 1, which serves as the guiding framework for the investigation.

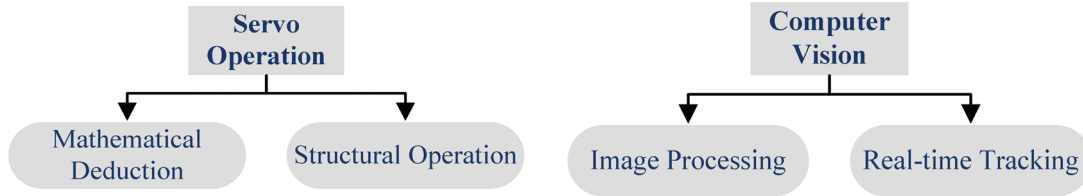


Fig. 1 2 Component of Object-tracking System

**2.1.1 Servo Operation**

Fig. 2 and Fig. 3 illustrate the basic principles and concepts of the structural operation of the 6 different servos.

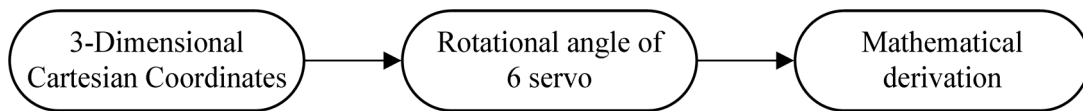


Fig. 2 Mathematical Deduction for the Kinematics

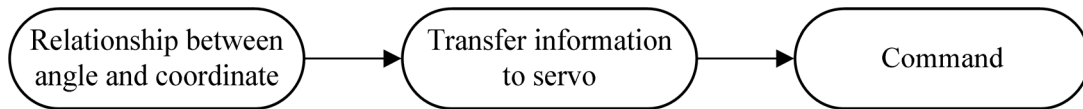


Fig. 3 Structural Operation for the Kinematics

**2.1.2 Computer Vision**

The complicated algorithms in the whole system are concentrated in this proportion in Fig. 4 and Fig. 5. The Flowcharts briefly introduce the essential logic of designing the system. In the rest of the paper, the process is introduced specifically.

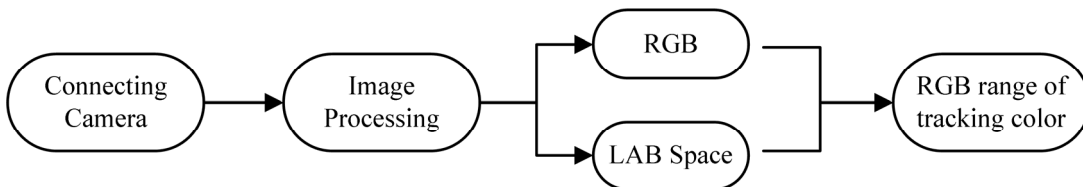


Fig. 4 Image Processing in Color-tracking

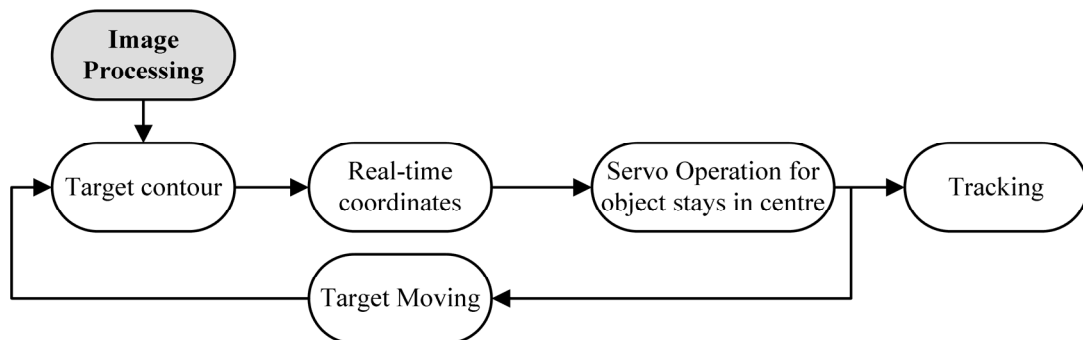


Fig. 5 Real-time Tracking of the Aim Object

**2.2. OpenCV Part: Color Tracking System**

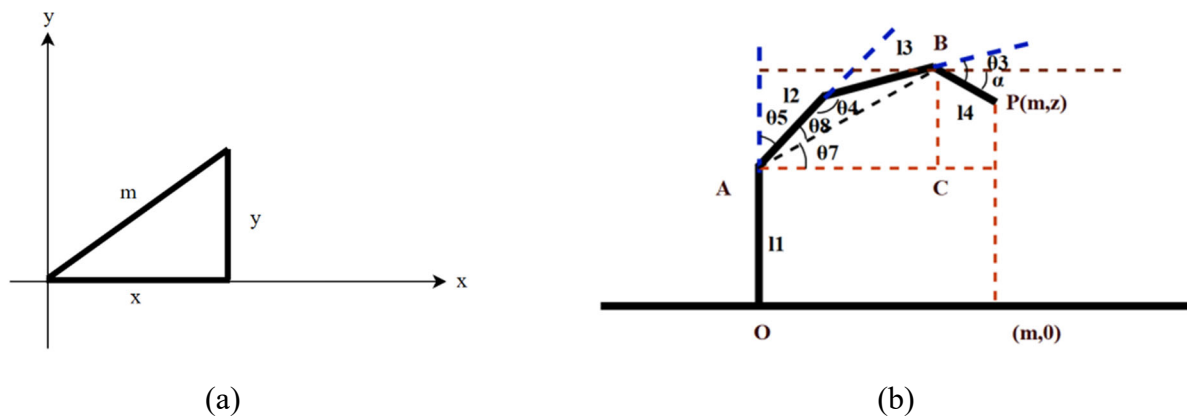
The Color Tracking System relies on the PID algorithm for precise color recognition [7]. A camera is linked to the second servo to capture images of both the background and the target object. Subsequently, the captured images undergo pixel analysis in the RGB mode and are then transformed into the LAB color space [8-9]. This transformation facilitates the identification of the pixel's corresponding color range [10-11], which is then compared against the user-defined target color range. The system further discerns and isolates the contours of the relevant pixels. Each enclosed contour's

area is diligently computed, with the maximum area being determined. To minimize potential interference, only areas exceeding a predetermined threshold are considered valid. The subsequent stage involves communicating the precise position of the object's center to the servo, thereby enabling the robotic arm to comprehend its intended destination. The algorithm diligently maintains the target object at the graph's center throughout the tracking process, ensuring seamless color tracking as the robotic arm continuously adjusts its movements to accommodate any displacements of the object.

**2.3. Kinematics Part: Servo Operation**

The operation of the six servos entails the practical application of Inverse Kinematics principles [12-13]. Once the object's position is obtained through the Color Tracking system, the Inverse Kinematics algorithm takes charge of controlling the rotation angles of the different servos, thereby guiding the robotic arm to its specific destination where the target object resides. By effectively utilizing the Color Tracking System, the robotic arm is endowed with precise awareness of the exact coordinates it must navigate to. Subsequently, the arm's motion is deconstructed into the rotations of its six servos through meticulous mathematical derivations. Users are required to input crucial parameters, including the camera's angle deviation from the horizontal line, the interservo distance, and the distance between the table and the robotic arm's base. The algorithm then automatically computes the four angles required for the rotation of four servos, transmitting this vital information to the servos, consequently orchestrating seamless and efficient movements of the robotic arm. To ensure the safety of the system, the complete rotational movement is set with a minimum duration of 2000 ms.

To establish a precise geometric model for the mechanical arm system, transforming the configuration into a 2-dimensional coordinate system is inevitable, as depicted in Fig. 6. This conversion simplifies the subsequent calculations and analysis. Taking the bottom of the robotic arm as the center O of the Cartesian coordinates, point P is the position of the top of the mechanical arm that contacts the potential target. The X-axis represents the horizontal distance and y-axis represents the front-and-back distance; z-axis refers to the height of point P. In Fig. 6, the left diagram indicates the front view of the arm, and the right diagram shows the plan view of the real-time situation.



**Fig. 6** 2-dimensional System of the Robotic Arm. (a) Coordinates from Top View; (b) Coordinates from Side View

In the plan view, m represents the horizontal distance between point O and the top of the arm. Thus, m can be calculated by:

$$m = \sqrt{x^2 + y^2} \tag{1}$$

Proceeding to the front view, through applying trigonometric principles, specifically the cosine rule, to determine the angles. The equation for calculating this angle was as follows:

$$\cos \theta_4 = \frac{l_2^2 + l_3^2 - (l_4 \cdot \sin \alpha + z - l_1)^2 + (m - l_4 \cdot \cos \alpha)^2}{2l_2l_3} \quad (2)$$

$$\theta_4 = \pi - \arccos\left(\frac{l_2^2 + l_3^2 - (l_4 \cdot \sin \alpha + z - l_1)^2 + (m - l_4 \cdot \cos \alpha)^2}{2l_2l_3}\right) \quad (3)$$

After careful simplification of the deductions and equations, a concise expression for the length of segment AB as well as two angles between the servos are obtained:

$$AB = \sqrt{(l_4 \cdot \sin \alpha + z - l_1)^2 + (m - l_4 \cdot \cos \alpha)^2} \quad (4)$$

$$\theta_7 = \arctan\left(\frac{l_4 \cdot \sin \alpha + z - l_1}{m - l_4 \cdot \cos \alpha}\right) \quad (5)$$

$$\theta_8 = \arcsin\left(\frac{\sin \theta_4}{AB} \cdot l_3\right) \quad (6)$$

The rest of the angle can be obtained by the equation:

$$\theta_5 = \frac{\pi}{2} - \theta_7 - \theta_8 \quad (7)$$

$$\theta_3 = \frac{\pi}{2} - \theta_4 - \theta_5 + \alpha \quad (8)$$

By implementing these calculations, accurately determining the coordinates and angles required for the precise positioning and movement of the mechanical arm during its operation is possible.

## 2.4. The Integration of the Algorithms

Through the aforementioned approach, the integrated camera system adeptly tracks and follows the object's movements, while the IK algorithm exerts precise control over the robotic arm's actions. This amalgamation empowers the system to effectively and dynamically relocate the target object when subjected to external forces. By continuously capturing images, the camera facilitates real-time repositioning of the aimed object, enabling the robotic arm to flawlessly track and pursue the object's movements.

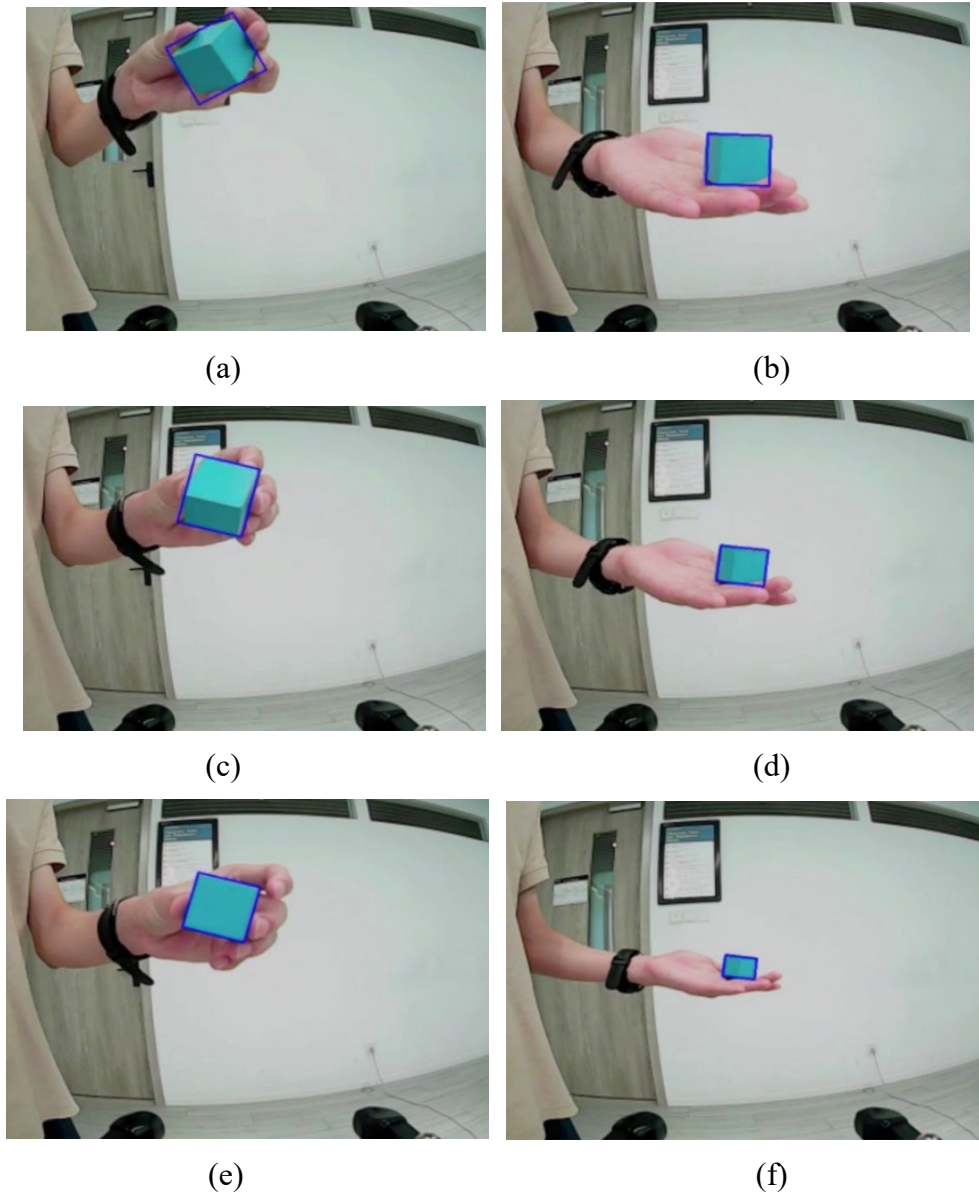
## 3. Experiment and Analysis

In the preceding sections, the integration of the system has been demonstrated, successfully achieving both motion control and object tracking. The results indicate that significant errors are absent; however, it is essential to acknowledge the inherent limitations of the robotic arms and the precision limitations arising from data rounding.

### 3.1. Accuracy of the Color Tracking System

The Color Tracking System effectively traces the target object by leveraging the LAB color space interpretation instead of relying on the less uniform and continuous RGB mode [14]. By adopting the LAB mode, the system demonstrates an enhanced success rate in identifying target objects. The graphical representations of the experiments show that the camera adeptly identifies the aim object, irrespective of the object's positioning angles. Additionally, the distance between the object and the camera exhibits only a negligible impact on the tracking process.

Six different images are presented in Fig. 7. The three on the left-hand side depict the object at the same distance but at different angles, while the others showcase the object at the same angles but varying distances.



**Fig. 7** Performances under Different Distances or Angles

(a), (c), (e): from Different Distances; (b),(d),(f): from Different Angles

### 3.2. Accuracy of Forward Kinematics

The Forward Kinematics process entails the precise rotation of the servos based on specified angles. The servos theoretically rotate up to 240 degrees, with the user being able to input integers ranging from 0 to 1000 to control the rotation. Notably, the input integer 500 corresponds to 0 degrees of the robotic arm. Consequently, the conversion formula from degrees ( $\theta$ ) to input integers ( $\beta$ ) is expressed as follows:

$$\beta = 500 + \frac{1000}{240} \theta \tag{9}$$

Simplified as

$$\beta = 500 + 4.17\theta \tag{10}$$

However, subsequent rigorous testing and data collection [Figure 4] have necessitated adjustments to the formula concerning distinct servos, as evidenced by the following equations:

$$\beta_3 = 502 + 4.21\theta_3 \tag{11}$$

$$\beta_4 = 499 + 4.20\theta_4 \tag{12}$$

$$\beta_5 = 501 + 4.12\theta_5 \tag{13}$$

$$\beta_6 = 502 + 4.11\theta_6 \tag{14}$$

According to Table 1, the subscript denotes the servo number. It becomes evident that the average values in the first two rows do not precisely correspond to those in the third row, signifying a level of inaccuracy in the calculation of Forward Kinematics, as illustrated in the following figure. Fig. 8 displays the data collected during the experiment, while Fig. 8 visually represents the discrepancies in the calculation of Forward Kinematics.

**Table 1.** Dataset of the Percentage Error in Forward Kinematics

No. Servo	Expected Angle (degrees)	Processing Angle (degrees)	Percentage Error
3	30	29.929	-0.237%
3	45	44.893	-0.238%
3	60	59.587	-0.238%
4	18	18.095	0.528%
4	36	36.19	0.528%
4	72	72.143	0.199%
5	30	29.854	-0.487%
5	45	44.903	-0.216%
5	60	59.951	-0.682%
6	30	29.927	-0.243%
6	45	44.769	-0.513%
6	60	59.854	-0.243%

Notwithstanding, it is noteworthy that, in the algorithm, once integers (beta) are calculated, its data type is coerced into an integer to conform to the servos' numerical input requirements. This transformation results in an error range of less than 1 percent. Overall, the error range is within an acceptable interval, affirming the validity and practicality of the algorithm.

### 3.3. Accuracy of Inverse Kinematics

Inverse Kinematics, a fundamental aspect of this study, involves the input of a three-dimensional coordinate to the robotic arm, thereby orchestrating the rotation of multiple servos to attain the specified position. The Inverse Kinematics process crucially determines the outcome of the arm's movement. While Forward Kinematics exhibits notable precision in servo rotation, the paramount factor affecting accuracy resides in the calculation of coordinates. Consequently, the algorithm governing Inverse Kinematics assumes paramount significance.

This algorithm is built upon Python's math library, which computes angles in radians rather than degrees. The transition from degrees to radians can amplify inaccuracies since the math library retains only 16 significant figures. Despite this limitation, the algorithm remarkably maintains a relatively high level of precision, as demonstrated in Table 2.

**Table 2.** Dataset of the Percentage Error in Inverse Kinematics

Coordinates Input	Coordinates Output	Percentage Error
(10,15,20)	(10.528,14.292,19.467)	4.22%
(20,10,5)	(19.663,9.332,5.152)	6.20%
(8,10,12)	(8.820,9.274,12.176)	6.33%

The table explicitly illustrates that the percentage error hovers around 5%. Notwithstanding, the IK algorithm continues to uphold an impressive level of accuracy in computing the correct three-dimensional Cartesian coordinates.

The experiment and analysis highlight the proficient functioning of the Color Tracking System, exhibiting precise object-tracing capabilities. Additionally, the Kinematics process successfully controls the servo rotations, albeit with minor variations between distinct servos that may contribute to some inaccuracy. However, the observed error range remains within an acceptable threshold, substantiating the overall effectiveness and practical viability of the proposed methodologies in Forward Kinematics, Inverse Kinematics, and color-based object tracking applications.

#### 4. Conclusion

The paper emphasizes the color-based object tracking of the robotic arms through kinematics. In conclusion, this study has underscored the fundamental tenets of robotic arm kinematics and color-driven object tracking, harmoniously amalgamating these facets to achieve both target identification and precise spatial coordination. Mathematical formulations were adroitly applied within the OpenCV module, thus facilitating meticulous servo rotations essential for the attainment of predefined three-dimensional Cartesian coordinates. Python, as the designated programming language for arms control, provided a robust and versatile computational framework. The transformation from RGB to the more discerning LAB color space allowed the robotic arm to proficiently discern target objects by their color attributes. Notably, the error margins remain within a commendable range. Specifically, the Forward Kinematics algorithm exhibits an error rate of less than 1%, while the Inverse Kinematics demonstrates an average error rate of approximately 5%.

This investigation substantiates the considerable potential of employing robotic arms in real-world scenarios on a broad scale, primarily attributable to their inherent precision. The ability to carry out tasks with precision and efficiency presents a viable alternative to human labor, particularly for mundane and repetitive functions. As the trend towards automating straightforward tasks continues to gain momentum, the implementation of robotic arms warrants further exploration in practical applications. Their capacity to undertake repetitive work with unwavering consistency presents a notable advantage under specific circumstances.

Furthermore, this study has thoroughly explored the core functionalities of the robotic arm system, skillfully leveraging its capabilities to seamlessly integrate kinematics and Color Tracking. This synergy enables the robotic arm to undertake diverse real-world missions, aligning with our objective of effectively merging mobile robotic arm capabilities with object-tracking functionalities. The capacity to manipulate and relocate specific objects is duly fulfilled, and this potential could conceivably be harnessed in scenarios characterized by color differentiation, such as drug classification and recognition within healthcare facilities or industrial settings.

While color-based object tracking represents an effective and accessible method for object tracing, it is prudent to acknowledge the presence of potentially superior alternatives. The efficacy of color-based identification may be compromised by factors such as similar background hues, excessive ambient lighting, or the coexistence of objects within the same color spectrum. Furthermore, the algorithm's presumption that the target object may occupy the largest portion of the image may restrict its ability to detect smaller objects. Additionally, the assumption that objects conform to regular geometric shapes imposes limitations. Future endeavors could explore the development of alternative high-precision object-tracking algorithms capable of adapting to a wider array of real-world scenarios.

#### Authors Contribution

All the authors contributed equally, and their names were listed in alphabetical order.

## References

- [1] Barbosa William S., Gioia Mariana M., Natividade V.G., et al. Industry 4.0: Examples of the use of the robotic arm for digital manufacturing processes. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 2020, 14(4): 1569-1575.
- [2] Mohsen Dadfarnia, Nader Jalili, Liu Zeyu, et al. An observer-based piezoelectric control of flexible Cartesian robot arms: theory and experiment. *Control Engineering Practice*, 2004, 12(8): 1041-1053.
- [3] Carron Andrea, Andrea Elena, Wermelinger Martin, et al. Data-driven model predictive control for trajectory tracking with a robotic arm. *IEEE Robotics and Automation Letters*, 2019, 4(4): 3758-3765.
- [4] Heyden Anders, Sparr Gunnar, Nielsen Mads, et al. *Computer Vision-ECCV in 7th European Conference on Computer Vision*. Peter Johansen, Copenhagen, 2002.
- [5] Lee Hai-Wu. The study of the mechanical arm and intelligent robot. *IEEE Access*, 2020, 8(7): 119624-119634.
- [6] Tang Shufeng, Zhou Pengfei, Xu Wang, et al. Design and experiment of dry-ice cleaning mechanical arm for insulators in a substation. *Applied Sciences*, 2020, 10(7): 2461.
- [7] Farkh Rihem, Khaled Aljaloud. Vision navigation based PID control for line tracking robot. *Intelligent Automation & Soft Computing*, 2023, 35(1): 901-911.
- [8] Kumah Charles, Zhang Ning, Raji Rafiu King, et al. Color measurement of segmented printed fabric patterns in lab color space from RGB digital images. *Journal of Textile Science and Technology*, 2019, 5(1): 1-18.
- [9] Süsstrunk Sabine, Robert Buckley, Steve Swen. Standard RGB color spaces, in *Proc. IS&T/SID 7th Color Imaging Conference*, Scottsdale, Arizona, 1999.
- [10] Kucuk Serdar, Zafer Bingul. *Robot kinematics: Forward and inverse kinematics*. Open Access Publisher, London, 2006.
- [11] Ragland Kirubaraj, Tharcis P. A survey on object detection, classification, and tracking methods. *Int. J. Eng. Res. Technol*, 2014, 3(11): 622-628.
- [12] Li Yun, Kiam Heong Ang, Gregory CY Chong. PID control system analysis and design. *IEEE Control Systems Magazine*, 2006, 26(1): 32-41.
- [13] Danelljan Martin, Khan Fahad Shahbaz, Felsburg Micheal, et al. Adaptive color attributes for real-time visual tracking." *Proceedings of the IEEE conference on computer vision and pattern recognition*. Columbus, OH, 2014, pp. 1090-1097
- [14] Recky Michal, Franz Leberl. Windows detection using k-means in cie-lab color space. *20th International Conference on Pattern Recognition*. Massachusetts Ave., NW Washington, DC, 2010.