IAES International Journal of Robotics and Automation (IJRA)
Vol. 15, No. 2, June 2026, pp. 281~294
ISSN: 2722-2586, DOI: 10.11591/ijra.v15i2.pp281-294

A path generation and control framework for 6-DOF robot in precision writing and drawing

Khoi Hoang Dinh, Khanh Tran Duy, Thien Bui Thanh, Quy Vo Quoc
Faculty of Electrical Engineering Technology, Industrial University of Ho Chi Minh City (IUH), Ho Chi Minh City, Vietnam

Article Info

Article history:

Received Jun 3, 2025
Revised Feb 10, 2026
Accepted Mar 6, 2026

Keywords:

ABB GoFa
Collaborative robot
Educational robotics
Path planning
Robot drawing
Robot writing

ABSTRACT

Robots are becoming increasingly integrated into everyday life, not only in industrial applications but also in creative, educational, and entertainment contexts. With recent advancements, collaborative robots are now lighter, safer, and easier to deploy alongside humans, making them well-suited for tasks that require precision and adaptability. This paper presents an integrated control framework for the ABB GoFa 6-DOF collaborative robot, enabling it to autonomously perform precise writing and drawing tasks. The system leverages CAD-based path design in SolidWorks and ABB RobotStudio's AutoPath tool to generate motion trajectories from a library of modeled characters, symbols, and figures. A socket-based communication interface connects the robot controller with a user-friendly human-machine interface (HMI), allowing users to input custom text or select predefined figures in real time. The framework has been implemented and validated on the physical ABB GoFa robot, demonstrating high accuracy, repeatability, and usability for applications such as public exhibitions and educational settings.

Corresponding Author:

Khoi Hoang Dinh
Faculty of Electrical Engineering Technology, Industrial University of Ho Chi Minh City (IUH)
Ho Chi Minh City, Vietnam
Email: hoangdinhkhoi@iuh.edu.vn

1. INTRODUCTION

Robots have traditionally been employed to enhance efficiency and reliability in manufacturing and automation processes. Their adoption has gradually expanded to domains such as agriculture, logistics, and space exploration, where robotic systems reduce human workload while maintaining high levels of precision and repeatability [1], [2]. More recently, advances in robotics and artificial intelligence (AI) have extended robotic applications beyond industrial environments into human-centered settings, including homes, workplaces, educational institutions, retail spaces, and service environments [3]–[5]. In these contexts, robots increasingly support assistance, collaboration, and service-oriented tasks, reflecting a shift toward closer human–robot interaction.

In parallel with this evolution, there has been growing interest in robots that replicate or augment human capabilities in creative and cognitive-motor activities. Representative examples include robotic systems capable of playing musical instruments, cooking, preparing coffee, or competing in games such as chess and air hockey [6]–[8]. Among these applications, robotic writing and drawing have attracted particular attention due to their technical challenges and symbolic significance. These tasks require precise path generation, smooth motion execution, and consistent surface interaction, while also showcasing robotic dexterity and creativity. Articulated and anthropomorphic robots are especially well suited to such applications because of their kinematic similarity to the human arm. When equipped with appropriate end-effectors, such as pens or markers, these robots can reproduce handwriting and drawings with high fidelity, enabling applications ranging from interactive art installations and STEAM education to public demonstrations and human–robot engagement platforms.

Jean-Pierre and Saïd [9] developed an "Artist Robot" to draw human portraits in a manner that mimics human artists. They used a 6-DOF KUKA industrial robot equipped with a camera, pen, and eraser to capture a visitor's image, process it, and generate a drawing trajectory. The robot operates in Futuroscope Park in France, combining real-time image processing (Canny edge detection) with robotic motion control to create portraits. It can complete a portrait in under five minutes and can process up to 1,000 trajectory points to balance speed and detail. The project has been successfully deployed as a park attraction since 2006, producing over 80 portraits daily and receiving high visitor interest.

Tresset and Fol Leymarie [10] also introduced Paul, a robotic system that creates observational face drawings without explicit knowledge of facial features or human-like learning. Instead, it uses a cycle of processes mimicking artistic techniques, enhanced by two forms of visual feedback, computational and physical, to iteratively refine its work. Pichkalev *et al*. [11] presented a system that extracts edge points from webcam images using a Canny edge detector and converts them into curves via cubic spline interpolation for robotic drawing. The system includes a user-friendly interface with adjustable parameters and uses a KUKA KR3 R540 robot to reproduce the processed image on paper. However, the setup in these works is highly task-specific and usually requires controlled lighting (e.g., spotlights) and a uniform background for reliable image capture.

Beyond portrait drawing, other works have explored broader aspects of robot-assisted sketching and visualization. For example, Song *et al*. [12] analyzed impedance control to create detailed pen art on arbitrary surfaces without explicit surface reconstruction or visual feedback. Similarly, Jain *et al*. [13] focused on a force-controlled portrait-drawing robot that can operate on arbitrarily shaped, non-calibrated surfaces by using force sensing to dynamically orient the pen, eliminating the need for surface calibration. Extending to 2.5D applications, Lam *et al*. [14] introduced techniques for drawing on contoured surfaces using a 3-DOF sketching platform, integrating image processing and surface projection algorithms to generate pen trajectories for both flat and uneven surfaces.

In parallel, artists and researchers have explored the use of robotic systems as autonomous or collaborative tools for artistic painting. Representative examples include the e-David system, which controls various painting mechanisms to produce robotic artworks [15], as well as approaches that combine non-photorealistic rendering techniques with robotic painting to generate distinctive artistic styles [16]. Additional interactive and collaborative painting systems have been reported in [17] and [18], demonstrating how robots can assist human users in region-based painting and shared creative processes.

In the domain of robotic text writing, Yussof *et al*. [19] proposed a character segmentation algorithm that decomposes letters into straight lines and curves. These primitives are stored and later used to guide a Mitsubishi RV-2AJ robotic arm in reproducing the characters. However, the study focuses solely on the Latin alphabet and lacks comprehensive experimental validation. Abdelaal [20] focused more on movement control programming using robot-specific programming languages. The work analyzed circular and straight-line motions, which are critical in industrial applications such as welding, painting, and machining. It also explored the relationship between motion speed and accuracy when commands are generated internally by the robot controller.

Sun and Xu [21] was among the first to introduce robotic calligraphy writing using a manipulator, a task that poses additional challenges due to variations in brush tip deformation and the need to assess ink wetness and dryness during writing. Similarly, Lin *et al*. [22] demonstrated a robotic arm capable of writing Chinese characters using a brush. In this work, brush stroke behavior was modeled by measuring brush footprints, such as shape, lag, and orientation, and representing strokes using Bézier curves combined with Direct Linear Transformation. While this approach avoids explicit physical modeling of the brush, it relies heavily on precise calibration and is sensitive to brush type. Nevertheless, the system achieved high-fidelity results, with an average evaluation score of 94/100 based on character shape, positioning accuracy, and stroke balance.

Another robotic calligraphy system for Chinese characters was presented by Chao *et al.* [23]. In a different direction, Zhang *et al*. [24] aimed to replicate human handwriting in real time by tracking hand movements using RGB-D data from a Kinect sensor. The authors used geometric moments and contour extraction, followed by a real-time smoothing technique, N-RPRI Interpolation, to convert discrete hand motion into continuous robot trajectories. Numerous studies on robotic writing [25], [26] and drawing [27]–[29] further demonstrate the growing interest in robotic systems for creative and expressive tasks.

Another important aspect in robotic writing and drawing is trajectory generation, as the resulting motions must simultaneously ensure geometric fidelity and smooth velocity and acceleration profiles. These requirements become particularly challenging when trajectories contain sharp corners, dense waypoints, or complex curved strokes. Prior studies have explored curve-based representations, such as Bézier-curve stroke modeling and brush footprint estimation, to obtain compact and smooth stroke parameterizations [22]; however, these methods typically require additional modeling effort and are sensitive to tool and surface characteristics. An alternative approach relies on spline interpolation (e.g., cubic B-splines) to smooth piecewise-linear waypoint paths while maintaining curvature continuity, which improves motion smoothness and reduces jerk but may introduce approximation errors and requires careful tuning of smoothing parameters [30]. Beyond geometric considerations, execution performance also depends on time parameterization under actuator constraints, where trapezoidal or limited-acceleration velocity profiles are commonly used to balance speed and accuracy [31], [32]. These considerations highlight a fundamental trade-off in robotic writing and drawing: high accuracy often favors dense waypoints and minimal blending, whereas high speed and smoothness favor spline-based smoothing and optimized time scaling, which may deviate from the intended geometry.

However, there remains a lack of a comprehensive and flexible framework for generating motion libraries that allow robots, especially articulated or high-degree-of-freedom robots (6 DOFs or more), to write arbitrary text or draw various figures. Most prior approaches are either highly task-specific or require significant manual programming effort. Designing precise trajectories, calibrating tool orientations, and managing communication between user interfaces and robot controllers often require expert knowledge in robot kinematics, and programming. There is a pressing need for systems that lower the entry barrier, enabling artists, educators, students, and other non-specialists to direct robotic drawing and writing without deep technical expertise.

To address these limitations, this work proposes a user-friendly and adaptable control pipeline for enabling writing and drawing capabilities on an ABB GoFa 6-DOF robot. Our approach integrates SolidWorks for creating text and symbol surfaces, ABB RobotStudio for converting those surfaces into motion paths, and socket-based communication with a custom human–machine interface (HMI). Through the HMI, users can input text or select predefined images, which are then converted into executable robot trajectories without manual trajectory programming. SolidWorks enables the design of all English alphabet characters, digits, and symbols as surface profiles, which are then processed in RobotStudio to extract path points. These are transmitted to the robot via TCP/IP socket communication, allowing real-time, dynamic task execution. The proposed framework has been implemented and tested on the ABB GoFa collaborative robot. Experimental results confirm the robot's ability to accurately execute diverse writing and drawing tasks based on user input, demonstrating both the feasibility and practical potential of the system for interactive exhibitions, education, and creative applications.

The remainder of this paper is organized as follows. Section 2 introduces the proposed framework, beginning with a system overview, followed by the design of the end-effector and experimental setup. It then describes the path generation procedure, the integration of the HMI, and the control framework that connects high-level user inputs with low-level robot motion execution. Section 3 presents experimental results obtained on the physical ABB GoFa robot, demonstrating writing, drawing, and cleaning tasks. Finally, section 4 concludes the paper by discussing the strengths and limitations of the proposed system and outlining directions for future work.

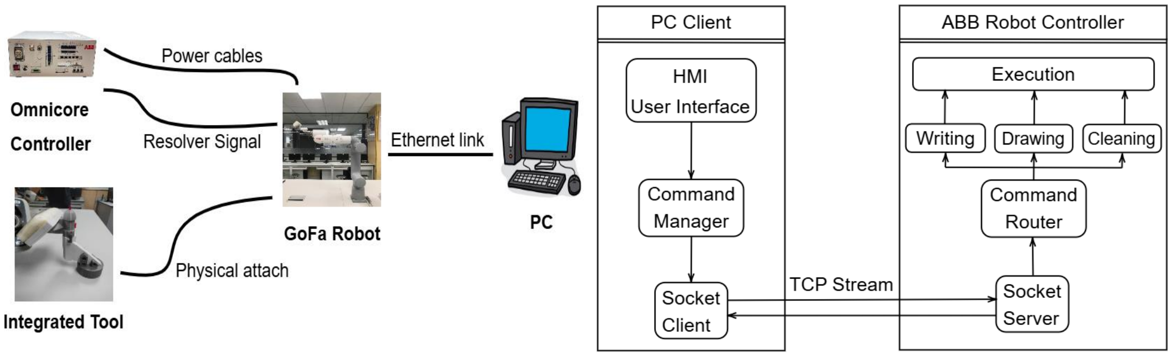
2. METHOD

This section outlines the complete integration process, from the tool and hardware setup to the control architecture and high-level user interface, which enables the GoFa robot to perform a wide range of writing and drawing tasks.

2.1. System overview

The overall system architecture of the proposed framework is illustrated in Figure 1. The hardware components are shown in Figure 1(a), where an ABB GoFa collaborative robot is equipped with a custom-designed end-effector tailored to the three target tasks addressed in this study, namely writing, drawing, and cleaning. The robot is powered and controlled by an ABB OmniCore C30 controller, which is responsible for motion execution, command handling, and communication with peripheral devices. The system interfaces with a user-friendly HMI, typically deployed on a user-side computer, via a TCP/IP network. This HMI enables users to define task parameters and establishes a socket-based TCP/IP connection with the robot controller.

Figure 1(b) depicts the operational workflow of the framework. The process begins with the user selecting one of the three available task modes. In the writing mode, the user inputs textual content and specifies the desired font size. In the drawing mode, predefined geometric figures are selected along with their corresponding dimensions. In the cleaning mode, the robot executes erasing motions to clear the workspace. Once the task parameters are defined, the HMI initiates a TCP/IP connection to the OmniCore controller, which operates a RAPID-based socket server. The overall communication follows a client–server architecture, where the HMI functions as the client and the robot controller acts as the server. Task commands are then transmitted to the robot for execution. Upon task completion, the robot sends feedback to the HMI to confirm successful execution and waits for subsequent commands, after which the procedure repeats.



(a) (b)

Figure 1. System architecture of the proposed framework: (a) hardware configuration and communication setup and (b) operational workflow and task execution process

The following sections present the key technical components of the proposed system in detail. First, the end-effector design is introduced, covering the mechanical structure, material selection, compliance mechanism, and mounting strategy used to ensure stable and versatile task execution. Next, the path generation workflow is described, including the CAD-based modeling process, the use of AutoPath for trajectory generation, scaling operations, and trajectory formatting for robot execution. The HMI is then presented, focusing on user input handling, socket-based communication, and task selection logic. Finally, the control algorithm is detailed, including the overall flowchart, data parsing procedures, motion planning strategy, and execution logic within the robot controller.

2.2. Integrated end-effector tool design and setup

One of the key challenges in robotic writing and drawing tasks lies in the design of an end-effector that enables accurate, stable, and repeatable interaction with a planar writing surface. The tool must be mechanically compatible with robot kinematics while providing sufficient compliance to compensate for surface unevenness and positioning uncertainties. Based on the concept introduced in [33], we developed a customized end-effector, referred to as the "Pentool", specifically tailored for use with the ABB GoFa collaborative robot.

The Pentool is designed to be fully compatible with the ABB GoFa flange interface, lightweight enough to remain within the robot's payload limitations, and easily replaceable to support rapid tool changes. To meet these requirements, the entire structure is fabricated using PET-G material through fused deposition modeling (FDM) 3D printing. PET-G was selected due to its favorable balance between mechanical strength, impact resistance, and flexibility. These properties allow the tool to withstand repeated contact forces during writing and cleaning tasks while maintaining sufficient robustness for long-term operation.

The mechanical structure of the Pentool consists of four main components: a socket flange, a base body, an upper body, and an integrated cleaning element. The socket flange is designed to mount securely onto the ABB GoFa robot's end-effector and incorporates a spring-loaded locking mechanism that enables fast, screw-free attachment and removal. Two lateral buttons actuate the locking mechanism, allowing the tool to be released or secured within seconds. This design significantly reduces setup time and supports rapid switching between different end-effectors during experiments or demonstrations in Figure 2.

The main body of the Pentool employs a modular two-part structure consisting of a base and an upper body connected through a twist-lock mechanism. This configuration allows quick replacement of the writing instrument without disassembling the entire tool. A compression spring is integrated inside the base, providing controlled axial compliance along the tool's vertical axis. This compliance compensates for minor height variations of the writing surface and ensures consistent contact force between the pen tip and the surface, thereby improving line quality and reducing the risk of excessive contact force that could damage either the tool or the writing medium. The effective compliance range is approximately 5–8mm, which was experimentally determined to be sufficient for stable writing performance.

Figure 2. The socket flange with a spring-loaded locking mechanism for quick, screw-free tool attachment to the robot's end-effector

In addition, a cleaning pad mounted at a 45-degree angle on the upper body enables the robot to perform self-cleaning actions without requiring tool replacement. This feature supports automated cleaning routines and improves system autonomy. The complete Pentool assembly, including its mounting position on the robot flange, is illustrated in Figure 3. The total mass of the assembled tool is approximately 180g, ensuring safe operation well within the payload capacity of the ABB GoFa robot.

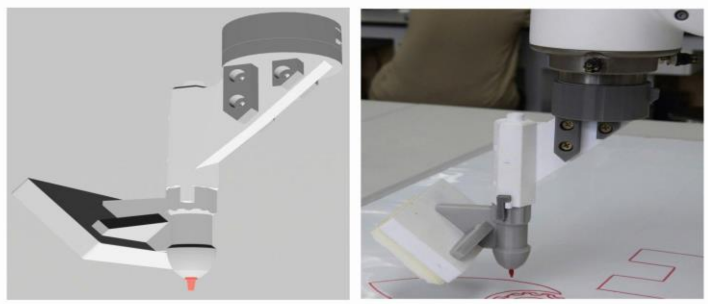


Figure 3. Design of the Pentool and its mounting on the ABB GoFa robot

2.3. Path generation for writing and drawing tasks

To perform each writing or drawing task, the robot requires a series of predefined waypoints. Using the built-in position control of the ABB Omnicore C30 controller, the robot can accurately move to the desired positions, provided that the motion data is correctly formatted and within the robot's reachable workspace. However, manually generating motion paths for letters, numbers, or figures is tedious, repetitive, and prone to human error, especially when handling large datasets. To address this, we propose a semi-automated method for generating motion paths. Our approach uses SolidWorks to create 3D models of letters, numbers, and symbols, which are then imported into RobotStudio. Using RobotStudio's AutoPath feature, way-points and trajectories are automatically generated from models. AutoPath analyses the geometry and robot workspace, then computes a collision-free, optimized sequence of points that the robot can follow. This method allows us to build a custom library containing all letters of the English alphabet, numerical digits, and various user-defined symbols or images. Importantly, new content can be added to the library with minimal additional effort. The overall procedure for generating path points is illustrated in Figure 4, and the following section provides a detailed explanation of the path generation method.

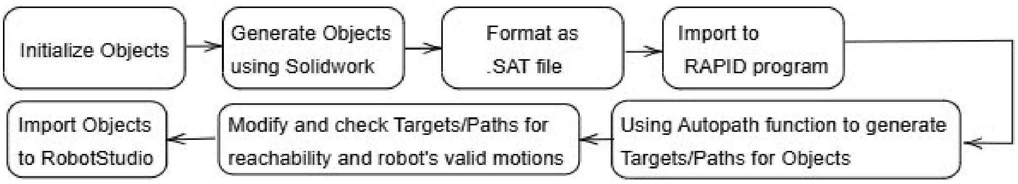
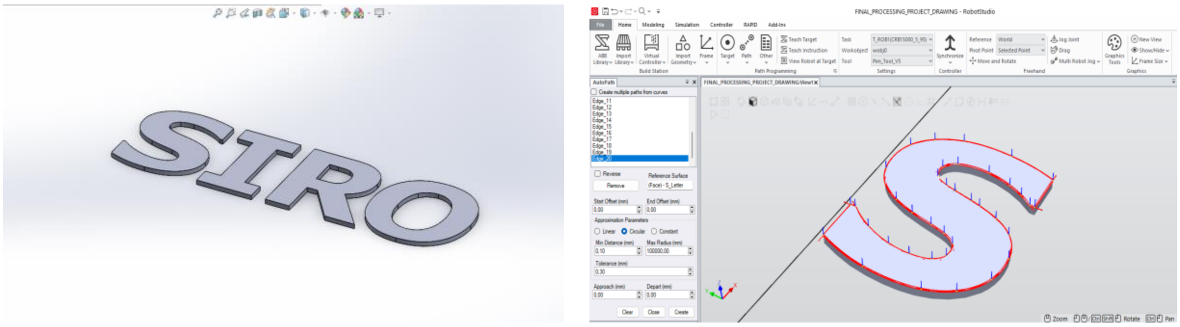


Figure 4. Step-by-step procedure for generating path points

2.3.1. Initialization and preprocessing

Path point generation begins by modeling letters, symbols, and images as 3D objects in a CAD environment like SolidWorks. For letters and numbers, they can be created using the Text Tool and extruded to form 3D features. For symbols and figures, they are either manually sketched or imported from vector files. After completing the sketches, these models are exported in .SAT format and imported into ABB RobotStudio using the Import Geometry feature. They are then aligned and positioned on a virtual surface that simulates the actual working area of the GoFa robot. The overall process of creating the 3D model and generating the corresponding path points is illustrated in Figure 5. Figure 5(a) illustrates an example of a 3D letter model created in SolidWorks that is imported into RobotStudio for path point generation.



(a) (b)

Figure 5. Example of 3D letter objects: (a) letters modeled in SolidWorks and (b) automatically generated path points of a letter using the AutoPath function in RobotStudio

2.3.2. Automatic path point generation

Using the AutoPath function in RobotStudio's Path Programming tool, the contours of imported 3D objects are automatically detected to generate motion paths. RobotStudio creates a sequence of targets along the selected contour, each containing position data from the geometry, orientation aligned with the surface, and configuration details such as the default workobject and joint setup. This process converts the object's geometry into a trajectory the robot can follow for writing or drawing. Figure 5(b) shows an example of path points generated along a letter's contour using AutoPath. After generating the path, some parameters may need to be tuned to ensure accurate execution. For example, individual targets may require updates to the workobject or tool orientation, while the overall path settings define motion type, speed, transition zones, and tool selection. These refinements enhance precision and reliability during operation.

Although AutoPath generates trajectories at a fixed scale defined by the original CAD model, these trajectories can be efficiently resized to match user-defined writing dimensions. This is achieved through a geometric scaling operation applied directly to the target positions generated.

Let a path point in the local coordinate frame of the imported geometry be defined as (1).

$$\boldsymbol{p} = [x, y, z]^T \tag{1}$$

To resize the writing pattern, a uniform scaling factor s is applied to the planar components of the path. Since writing occurs on a planar surface, scaling is performed only in the XY plane, while the vertical coordinate z is kept constant to preserve consistent surface contact (2).

$$\boldsymbol{p}' = [s \cdot x, s \cdot y, z]^T \tag{2}$$

After scaling, the path is transformed from the local object coordinate frame into the robot workspace using a rigid body transformation. This transformation aligns the path with the desired writing location and orientation on the working surface (3),

$$\boldsymbol{p}_{world} = R \cdot \boldsymbol{p}' + \boldsymbol{p_t} \tag{3}$$

where $R \in \mathbb{R}^{3\times3}$ is the rotation matrix defining the orientation of the workobject frame, and $\boldsymbol{p_t} \in \mathbb{R}^{\mathbf{3}}$ is the translation vector specifying its position relative to the robot base frame.

This transformation strategy allows the same geometric path to be reused at different sizes and positions without regenerating the underlying CAD model or recalculating the trajectory in AutoPath. As a result, the system supports flexible scaling and placement of writing and drawing tasks while maintaining consistent motion behavior and surface contact quality. The implementation of path generation and scaling procedure is summarized in Algorithm 1.

Algorithm 1. AutoPath-Based Trajectory Generation

Input: CAD model G, workobject $WObj$, tool $Tool_0$, scale factor s, transform (R, p_t)
Output: Executable target list T_{exec}
$G \leftarrow$ ImportAndAlign$(G, WObj)$
$T \leftarrow$ AutoPathExtractTargets(G)
for each $\tau \in T$ **do**
    AssignToolWObj $(\tau, Tool_0, WObj)$
    OrientToPlaneNormal $(\tau, WObj)$
    $\tau.p \leftarrow R \cdot [s \cdot \tau.x, s \cdot \tau.y, \tau.z]^T + p\_t$
end for
ExportPath (T)

2.3.3. Limitations and discussion

While the AutoPath-based approach significantly simplifies trajectory generation, several limitations should be noted. First, complex geometries containing overlapping contours or nested shapes may result in ambiguous traversal order. In such cases, manual reordering or segmentation of the path is required to ensure correct stroke sequencing. Second, AutoPath does not inherently optimize stroke direction or pen lifting behavior, which may lead to unnecessary motion or discontinuities. These issues are addressed through post-processing and rule-based path refinement. Finally, since AutoPath operates on geometric features rather than semantic stroke information, the generated paths may not always reflect human-like writing behavior. Future work will integrate stroke-level planning and vision-based feedback to enable adaptive correction and more natural writing trajectories.

2.4. Human–machine interface (HMI)

The human–machine interface (HMI), shown in Figure 6, provides an intuitive platform for configuring and executing robot tasks. The interface is organized into three functional modes: writing, drawing, and cleaning. In writing mode, users input text content and specify the desired font size. In drawing mode, users select predefined shapes or import custom images and adjust their geometric scale before execution. In cleaning mode, the user can command either a full area erasing operation or a localized cleaning action targeting a specific region. In addition, the HMI allows users to configure communication parameters such as the robot controller's IP address and port number, enabling flexible deployment across different network environments without modifying the underlying control software.

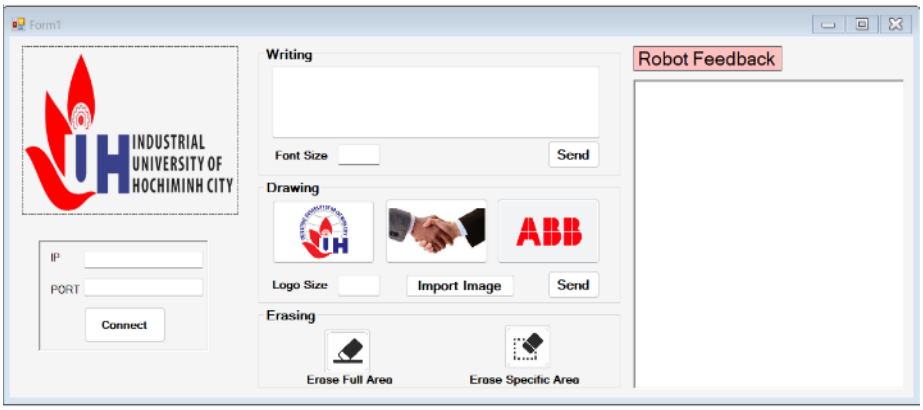


Figure 6. User-oriented HMI

2.4.1. Communication architecture and data exchange

Communication between the HMI and the robot controller follows a client–server architecture implemented over a TCP/IP socket connection. The HMI acts as the client, while the ABB OmniCore controller runs a RAPID-based socket server listening on port 1025. Once connected, the HMI transmits structured command packets containing the task type (writing, drawing, or cleaning), relevant parameters (e.g., text content, scale, or cleaning mode), and execution flags. The lightweight message format ensures low communication latency, typically within a few milliseconds, which is sufficient for supervisory control. Basic error handling is implemented to reject malformed data and notify the user in case of connection failures, ensuring safe and reliable operation.

2.4.2. Execution feedback and error handling

Once a command is received, the robot controller validates the data format and initiates the corresponding RAPID routine. Execution status messages, including task start, completion, or error notifications, are sent back to the HMI in real time. This feedback is also displayed in the HMI panel shown in Figure 6, allowing users to monitor system state and execution progress. Basic error handling mechanisms are implemented to ensure safe and reliable operation. If a connection attempt fails, the HMI notifies the user and prevents command transmission until communication is restored. Similarly, malformed or incomplete data packets are rejected by the controller, and an error message is returned to the HMI. These measures prevent unintended robot motion and enhance system robustness during operation.

2.5. Control algorithm and motion execution

This section outlines the control scheme developed to process user input into executable commands for the ABB GoFa robot. Communication begins with a TCP/IP socket connection between the HMI client and the robot controller. The user configures the correct IP address and port to match the robot's settings. Once connected, the system is ready to receive commands. The HMI supports two main functions: writing/drawing and cleaning, with the appropriate tool selected for each. Users can input text and font size or choose predefined figures from a graphical library. When the "Send" button is pressed, the data is encoded and transmitted to the controller for processing.

Upon receiving a command, the robot controller processes the input according to its type: text, drawing, or cleaning. For text input, the content is parsed character by character, with each character matched to a corresponding entry in a local stroke library and scaled to the specified font size. Drawing inputs are handled using predefined templates, which are similarly scaled to the user-defined dimensions. For cleaning operations, two modes are supported: selective cleaning and full-surface cleaning. In selective mode, the user specifies a rectangular region on the board to be cleaned. In full-surface mode, the robot executes a coverage path over the entire previously marked area to ensure complete cleaning. Tool changes, positioning, workspace boundaries, and safety checks are systematically handled for each mode prior to execution to ensure reliable and safe operation.

The final output is a normalized list of path points, meaning that all coordinates are transformed into a consistent reference frame, ensuring that paths generated from different input types—text, drawings, or cleaning commands—share a common coordinate basis. This unified representation is then passed to the motion planning module for trajectory generation and execution. A detailed flowchart of the overall control framework is shown in Figure 7.

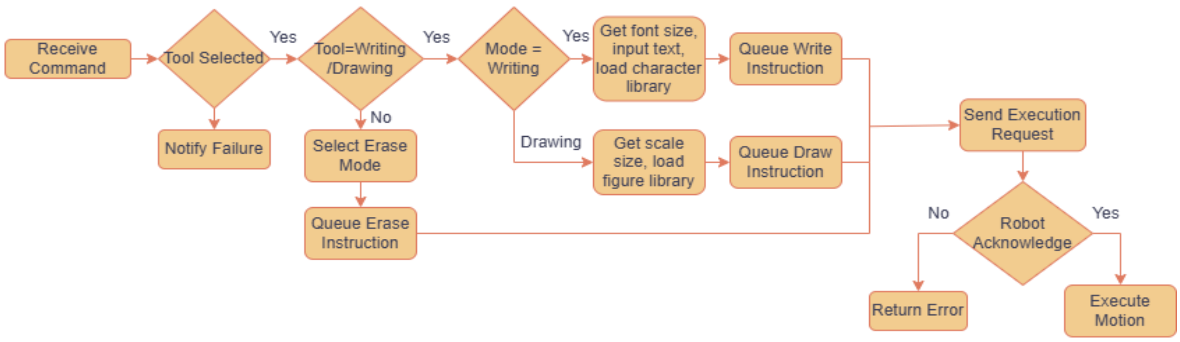


Figure 7. Control algorithm for writing, drawing and cleaning tasks using the ABB GoFa robot

Motion execution is performed by the OmniCore controller. For each trajectory segment, motion parameters such as velocity, acceleration, and blending zone are explicitly defined to ensure smooth and safe operation. Typical parameters include moderate linear velocities for writing and drawing tasks, reduced acceleration near surface contact, and fine blending zones to maintain continuous motion without abrupt stops.

Before execution, safety checks are applied to verify that all target points lie within the predefined workspace limits and that no tool collisions are expected. If a violation is detected, execution is aborted and a warning is sent to the HMI. Once validated, the robot executes the trajectory sequentially. Status feedback, including execution start, completion, or error conditions, is continuously transmitted back to the HMI, allowing real-time monitoring. The overall procedure of planning and execution is summarized in Algorithm 2.

Algorithm 2. Motion planning and execution procedure

```
Input: Command 𝑪 = {τ, θ}
Output: Status y ∈ {DONE, ERROR}
1.   Receive 𝑪
2.   If ToolSelected(τ) = false then return ERROR
3.   I ← (τ = ERASE) ? Erase(θ)
       : (τ = WRITE) ? Write(Parse(θ), Font(θ))
                     : Draw(Figure(θ), Scale(θ))
4.   SetStartPose(); SendExecute()
5.   If  Ack() = false then return ERROR
6.   Execute(I); return DONE
```

3. RESULTS AND DISCUSSION

This section presents several performances of the GoFa robot in writing and drawing tasks. After completing the library for letters, numbers and images, the robot is able to execute the given motions accurately and is able to write any text and draw any image.

3.1. Motion parameters and safety considerations

All experiments were conducted using a consistent set of motion parameters and experimental conditions; any deviation from these settings is explicitly stated where applicable. Cartesian linear motions were employed for all writing, drawing, and cleaning tasks to ensure accurate surface-following behavior. The maximum linear velocity was set to 1000 mm/s for most trajectories, while acceleration limits were automatically managed by the ABB OmniCore controller to maintain smooth contact with the writing surface. To maximize positioning accuracy and stroke fidelity, the transition (blending) zone was set to zero, forcing the robot to reach each target point precisely before moving to the next. Motion interpolation and velocity profile generation are handled internally by the OmniCore controller, which follows the standard trapezoidal velocity profile defined in ABB RAPID motion planning, consisting of acceleration, constant-velocity, and deceleration phases.

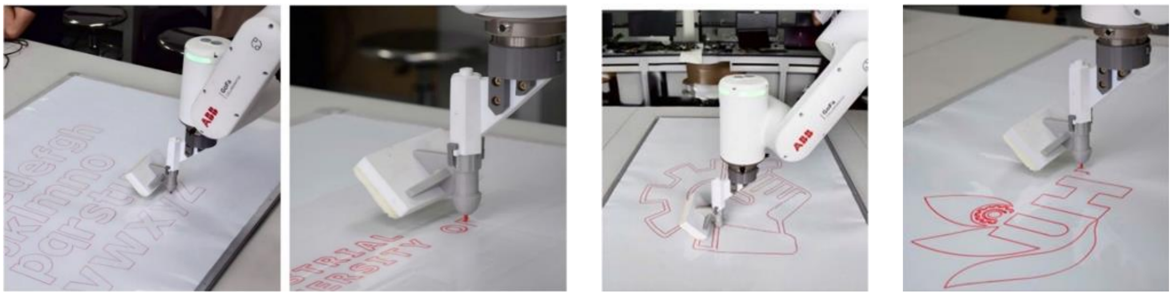
Safety checks are systematically applied prior to execution to ensure reliable and repeatable operation. All generated target points are verified to lie within predefined workspace boundaries corresponding to the physical writing surface. During all experiments, the commanded motion along the z-axis is kept constant. Vertical compliance is instead provided by the spring mechanism integrated into the Pentool, which allows a displacement of approximately ±5 mm along the z-direction. This compliance is sufficient to compensate for minor height inconsistencies of the writing surface while maintaining stable contact force.

To ensure reproducibility, all experiments were performed using the same hardware and software configuration. CAD models were created using SolidWorks, and path generation was carried out in ABB RobotStudio using the AutoPath feature. The robot was controlled by an ABB OmniCore controller running RobotWare v7.18. Prior to experimentation, the workspace and tool configuration were calibrated once and reused consistently across all trials.

3.1. Text writing performance and path generation

We developed a library containing all 26 letters of the English alphabet (both uppercase and lowercase), as well as digits from 0 to 9. Depending on the complexity of each character's shape, between 10 to 50 path points are typically required to accurately represent it. Simpler characters such as "i", "I", or "l", "L" require fewer points, whereas more complex characters like "g", "G", or "q", "Q" demand denser trajectories for precise rendering. Linear strokes are executed using straight-line motions, while curved segments, such as those in "c", "C", or "s", "S", are handled using circular motions. All path points are automatically generated using the procedure mentioned above, with only minor manual refinements. These refinements include adjusting the tool orientation to ensure proper alignment during writing, repositioning starting points to mimic typical human writing behavior, and fine-tuning path points to more closely follow the contours of certain characters. These improvements help enhance the natural appearance and accuracy of the robot's writing motions. For multi-line text input, the robot executes a line-by-line writing sequence, moving from left to right across each line, then returning to the left margin and shifting downward by a constant spacing before continuing with the next line. This motion pattern closely imitates human handwriting and ensures uniform character alignment across lines.

Through the HMI, users can enter custom text and specify desired font sizes. This input is then encoded and transmitted to the robot for execution. Figure 8 presents several examples of the ABB GoFa robot performing writing and drawing tasks. In particular, Figure 8(a) illustrates the robot executing text-writing operations with smooth and precise motion across different font sizes. While the present implementation supports only the English alphabet, the proposed approach is readily adaptable to complex font styles or multilingual inputs, as long as the characters can be represented as SolidWorks models for path point generation. It is also worth noting that the writing surface used in our experiments is not perfectly flat, with slight height variations across the board. However, thanks to the integrated spring mechanism in our pen tool, the robot can maintain consistent contact with the surface, enabling successful writing without excessive pressure or loss of contact.



(a) (b) (c)

Figure 8. Examples of the ABB GoFa robot performing writing and drawing tasks: (a) text writing in various font size, and drawing complex figures of (b) the simple "cooperative" symbol and (c) the university logo

3.2. Symbol and image drawing performance

For drawing tasks, although path points are automatically generated to follow the contours of a given symbol or image, additional refinement is often necessary, particularly for large, complex, or highly detailed figures. Compared to text writing, symbol and image drawing typically require more post-processing to ensure visual fidelity and natural motion. In cases where a figure consists of multiple layers or nested contours, manual selection of individual contours may be needed during the path generation process to achieve the desired level of accuracy.

The number of path points required to capture a figure depends largely on its complexity. For example, the simple "cooperative" symbol illustrated in Figure 8(b) requires approximately 243 path points, while the university logo in Figure 8(c) involves around 655 path points due to its intricate design and finer structural details. In our implementation, both symbols and images are stored in a predefined library and can be accessed via the HMI, where users can also freely adjust the scale of the drawing. Figures 8(b) and 8(c) showcase the GoFa robot successfully executing drawing tasks with varying levels of complexity, demonstrating the flexibility and precision of the proposed system.

3.3. Automated board cleaning with mounting pad integration

To enable autonomous board maintenance, the GoFa robot is equipped with a custom-designed mounting pad for cleaning operations. For this task, the system switches the Tool Center Point (TCP) to the one corresponding to the cleaning pad and executes a predefined motion sequence optimized for wiping the board surface. The cleaning trajectory maintains consistent pressure and full coverage, ensuring that the pad remains perpendicular to the board throughout the process. In this work, the cleaning procedure was experimentally verified to determine the optimal number of wiping passes and appropriate contact pressure during motion. Alternatively, a vision-based method could be employed to automatically assess surface cleanliness and adjust the number of cleaning sequences accordingly.

Figure 9 shows the robot configured with the cleaning pad attached, preparing to clean the board. This feature enhances the overall autonomy of the system by allowing it to clear previous content without human intervention, making it ready for subsequent writing or drawing tasks. Such functionality is especially useful in demonstration or classroom settings where repeated tasks are performed in sequence.

Figure 9. Configuration of the GoFa robot with the integrated cleaning pad mounted at the end-effector

3.4. Quantitative evaluation and system adaptability

Multiple writing and drawing trials were conducted to evaluate the accuracy and repeatability of the proposed system. Across repeated executions under identical conditions, consistent stroke placement and character geometry were observed, with no noticeable drift between trials. These results demonstrate that the robot can execute writing and drawing tasks with sufficient accuracy for clear handwriting and precise figure rendering. Execution time increased with task complexity: writing individual letters, as shown in Figure 8(a), typically required 3–10s depending on the selected scale, whereas drawing complex figures with several hundred path points, such as those in Figures 8(b) and 8(c), required approximately 50s and 145s, respectively. Compared to manual point-by-point programming, the proposed AutoPath-based approach significantly reduces setup time and improves repeatability.

The proposed framework is readily adaptable to other robotic platforms and task variations. Since the task logic and path generation are decoupled from robot-specific motion execution, the approach can be transferred to other industrial or collaborative robots (e.g., UR or KUKA systems) by modifying only the low-level motion interface. Multilingual writing can be supported by extending the character library with additional CAD models. Although experiments in this work were conducted with the writing surface placed horizontally on a table, the system can be adapted to boards mounted at arbitrary orientations by appropriately redefining the workobject frame. For curved surfaces with known geometric characteristics (e.g., spherical surfaces), the method can be extended with minor modifications. However, writing on unknown or irregular curved surfaces would require additional sensing, such as force–torque feedback or vision-based surface estimation.

4. CONCLUSION

In this work, we presented the development of a simple and easily integrable framework that enables a 6-DOF ABB GoFa robot to perform a variety of writing, drawing, and cleaning tasks. The system allows users to input custom text, numbers, and symbols through a user-friendly HMI, and automatically generates corresponding robot paths with minimal setup. Our framework has been successfully implemented on a real robot, demonstrating its potential for educational demonstrations, public exhibitions, and interactive robotics applications. The proposed solution supports an expandable library of characters and images, making it accessible to users without in-depth knowledge of robotic programming or configuration. The integration of a custom-designed writing tool and an automated cleaning pad further enhances the autonomy and versatility of the system.

However, there are still some limitations in our framework. First, a certain level of manual adjustment is required during the path generation process to ensure smooth and natural robot motion, particularly for complex or irregular shapes. This is crucial, as unnatural or jerky movements can negatively impact on the user experience; a key aspect of our user-oriented application. Additionally, the system currently lacks real-time feedback mechanisms, such as vision or force sensing, which restricts its ability to compensate for surface irregularities, tool slippage, or positioning errors.

Future work will focus on extending the framework with real-time feedback and adaptive control mechanisms. In particular, the integration of vision systems can enable online surface tracking and error detection, allowing continuous monitoring of execution quality. Force or force–torque sensors could be incorporated to regulate contact pressure during writing and cleaning tasks, improving robustness on uneven or compliant surfaces. Furthermore, online trajectory correction strategies could be implemented to adapt motion paths dynamically in response to sensory feedback. Moreover, machine learning techniques could be employed to automatically extract object contours from images, enabling a fully autonomous path generation process. A more robust path optimization module could also be integrated to refine trajectories automatically and ensure smoother, more reliable execution.

ACKNOWLEDGMENTS

This work is supported by the Laboratory of Robotics and Intelligent Control, Faculty of Electrical Engineering Technology (FEET), Industrial University of Ho Chi Minh city (IUH), Vietnam.

FUNDING INFORMATION

Authors state there is no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Khoi Hoang Dinh	✓	✓	✓	✓	✓	✓			✓	✓		✓	✓	
Khanh Tran Duy		✓	✓			✓		✓	✓		✓	✓		
Thien Bui Thanh	✓		✓	✓			✓			✓	✓			✓
Quy Vo Quoc				✓	✓		✓	✓	✓		✓			✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

REFERENCES

[1] T. Brogårdh, "Present and future robot control development-An industrial perspective," *Annual Reviews in Control*, vol. 31, no. 1, pp. 69–79, 2007, doi: 10.1016/j.arcontrol.2007.01.002.

[2] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human–robot collaboration," *Autonomous Robots*, vol. 42, no. 5, pp. 957–975, 2018, doi: 10.1007/s10514-017-9677-2.

[3] D. Belanche, L. V. Casaló, C. Flavián, and J. Schepers, "Service robot implementation: a theoretical framework and research agenda," *Service Industries Journal*, vol. 40, no. 3–4, pp. 203–225, 2020, doi: 10.1080/02642069.2019.1672666.

[4] L. D. Evjemo, T. Gjerstad, E. I. Grøtli, and G. Sziebig, "Trends in smart manufacturing: role of humans and industrial robots in smart factories," *Current Robotics Reports*, vol. 1, no. 2, pp. 35–41, 2020, doi: 10.1007/s43154-020-00006-5.

[5] C. Di Napoli, G. Ercolano, and S. Rossi, "Personalized home-care support for the elderly: a field experience with a social robot at home," *User Modeling and User-Adapted Interaction*, vol. 33, no. 2, pp. 405–440, 2023, doi: 10.1007/s11257-022-09333-y.

[6] J. Liu *et al.*, "Robot cooking with stir-fry: bimanual non-prehensile manipulation of semi-fluid objects," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5159–5166, 2022, doi: 10.1109/LRA.2022.3153728.

[7] C. Matuszek *et al.*, "Gambit: An autonomous chess-playing robotic system," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2011, pp. 4291–4297, doi: 10.1109/ICRA.2011.5980528.

[8] Y. F. Li and C. Y. Lai, "Intelligent algorithm for music playing robot - Applied to the anthropomorphic piano robot control," in *IEEE International Symposium on Industrial Electronics*, 2014, pp. 1538–1543, doi: 10.1109/ISIE.2014.6864843.

[9] G. Jean-Pierre and Z. Saïd, "The artist robot: A robot drawing like a human artist," in *2012 IEEE International Conference on Industrial Technology, ICIT 2012, Proceedings*, 2012, pp. 486–491, doi: 10.1109/ICIT.2012.6209985.

[10] P. Tresset and F. Fol Leymarie, "Portrait drawing by Paul the robot," *Computers and Graphics (Pergamon)*, vol. 37, no. 5, pp. 348–363, 2013, doi: 10.1016/j.cag.2013.01.012.

[11] M. Pichkalev, R. Lavrenov, R. Safin, and K. H. Hsia, "Face drawing by KUKA 6 axis robot manipulator," in *Proceedings - International Conference on Developments in eSystems Engineering, DeSE*, 2019, vol. October-2019, pp. 709–714, doi: 10.1109/DeSE.2019.00132.

[12] D. Song, T. Lee, and Y. J. Kim, "Artistic pen drawing on an arbitrary surface using an impedance-controlled robot," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2018, pp. 4085–4090, doi: 10.1109/ICRA.2018.8461084.

[13] S. Jain, P. Gupta, V. Kumar, and K. Sharma, "A force-controlled portrait drawing robot," in *Proceedings of the IEEE International Conference on Industrial Technology*, 2015, vol. 2015-June, no. June, pp. 3160–3165, doi: 10.1109/ICIT.2015.7125564.

[14] J. H. M. Lam, K. W. Lo, and Y. Yam, "Robot drawing techniques for contoured surface using an automated sketching platform," in *2007 IEEE International Conference on Automation Science and Engineering*, Sep. 2007, pp. 735–740, doi: 10.1109/COASE.2007.4341738.

[15] J. M. Gülzow, P. Paetzold, and O. Deussen, "Recent developments regarding painting robots for research in automatic painting, artificial creativity, and machine learning," *Applied Sciences (Switzerland)*, vol. 10, no. 10, p. 3396, 2020, doi: 10.3390/APP10103396.

[16] L. Scalera, S. Seriani, A. Gasparetto, and P. Gallina, "Non-photorealistic rendering techniques for artistic robotic painting," *Robotics*, vol. 8, no. 1, p. 10, 2019, doi: 10.3390/robotics8010010.
[17] O. Igno-Rosario, C. Hernandez-Aguilar, A. Cruz-Orea, and A. Dominguez-Pacheco, "Interactive system for painting artworks by regions using a robot," *Robotics and Autonomous Systems*, vol. 121, p. 103263, 2019, doi: 10.1016/j.robot.2019.103263.
[18] C. Guo *et al.*, "Skywork-daVinci: A novel CPSS-based painting support system," in *IEEE International Conference on Automation Science and Engineering*, 2020, vol. 2020-Augus, pp. 673–678, doi: 10.1109/CASE48305.2020.9216814.
[19] S. Yussof, A. Anuar, and K. Fernandez, "Algorithm for robot writing using character segmentation," in *Third International Conference on Information Technology and Applications (ICITA'05)*, 2005, vol. 2, pp. 21–24, doi: 10.1109/ICITA.2005.58.
[20] M. Abdelaal, "A study of robot control programing for an industrial robotic Arm," in *ACCS/PEIT 2019 - 2019 6th International Conference on Advanced Control Circuits and Systems and 2019 5th International Conference on New Paradigms in Electronics and Information Technology*, 2019, pp. 23–28, doi: 10.1109/ACCS-PEIT48329.2019.9062878.
[21] Y. Sun and Y. Xu, "A calligraphy robot - Callibot: Design, analysis and applications," in *2013 IEEE International Conference on Robotics and Biomimetics, ROBIO 2013*, 2013, pp. 185–190, doi: 10.1109/ROBIO.2013.6739456.
[22] H. I. Lin, X. Chen, and T. T. Lin, "Calligraphy brush trajectory control of by a robotic arm," *Applied Sciences (Switzerland)*, vol. 10, no. 23, pp. 1–21, 2020, doi: 10.3390/app10238694.
[23] F. Chao *et al.*, "Robotic free writing of chinese characters via human-robot interactions," *International Journal of Humanoid Robotics*, vol. 11, no. 1, 2014, doi: 10.1142/S0219843614500078.
[24] D. Zhang, J. Chen, M. Zhang, and H. Ji, "Real time writing reproduction by robot Arm," in *8th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, CYBER 2018*, 2019, pp. 1030–1035, doi: 10.1109/CYBER.2018.8688073.
[25] M. Balaganesh, E. Logashanmugam, C. S. Aadhitya, and R. Manikandan, "Robotic Arm showing writing skills by speech recognition," in *International Conference on "Emerging Trends in Robotics and Communication Technologies", INTERACT-2010*, 2010, pp. 12–15, doi: 10.1109/INTERACT.2010.5706158.
[26] D. Pérez-Marcos, J. A. Buitrago, and F. D. G. Velásquez, "Writing through a robot: A proof of concept for a brain-machine interface," *Medical Engineering and Physics*, vol. 33, no. 10, pp. 1314–1317, 2011, doi: 10.1016/j.medengphy.2011.06.005.
[27] A. El-Barkouky, A. Mahmoud, J. Graham, and A. Farag, "An interactive educational drawing system using a humanoid robot and light polarization," in *2013 IEEE International Conference on Image Processing, ICIP 2013 - Proceedings*, 2013, pp. 3407–3411, doi: 10.1109/ICIP.2013.6738703.
[28] C. F. Hsu, W. H. Kao, W. Y. Chen, and K. Y. Wong, "Motion planning and control of a picture-based drawing robot system," *IFSA-SCIS 2017 - Joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems*, 2017, doi: 10.1109/IFSA-SCIS.2017.8023232.
[29] S. Nasrat, T. Kang, J. Park, J. Kim, and S. J. Yi, "High-speed, high-quality robotic portrait drawing system," in *IEEE International Workshop on Robot and Human Communication, RO-MAN*, 2023, pp. 1042–1047, doi: 10.1109/RO-MAN57019.2023.10309559.
[30] J. Pan, L. Zhang, and D. Manocha, "Collision-free and curvature-continuous path smoothing in cluttered environments," in *Robotics*, MIT Press, 2019, pp. 233–240.
[31] T. Kunz and M. Stilman, "Time-optimal trajectory generation for path following with bounded acceleration and velocity," in *Robotics: Science and Systems*, 2013, vol. 8, pp. 209–216, doi: 10.15607/rss.2012.viii.027.
[32] J. Kim and E. A. Croft, "Online near time-optimal trajectory planning for industrial robots," *Robotics and Computer-Integrated Manufacturing*, vol. 58, pp. 158–171, 2019, doi: 10.1016/j.rcim.2019.02.009.
[33] Thingiverse, "Dynamic marker and eraser tool for industrial robot ABB IRB140." Accessed: June 2, 2025. [Online]. Available: https://www.thingiverse.com/thing:1338629.

BIOGRAPHIES OF AUTHORS

Khoi Hoang Dinh is a lecturer at the Faculty of Electrical Engineering Technology (FEET), Industrial University of Ho Chi Minh City (IUH), Vietnam. He received both his bachelor's and master's degrees from Ho Chi Minh City University of Technology, Vietnam. He later earned his Ph.D. in Robotics and Control from the Technical University of Munich (TUM), Germany, in 2024. His research interests span various areas within robotics and control, including human-robot collaboration, optimal control, model predictive control, motion planning, and autonomous driving. He can be contacted at hoangdinhkhoi@iuh.edu.vn.

Khanh Tran Duy is an undergraduate student at the Faculty of Electrical Engineering Technology (FEET), Industrial University of Ho Chi Minh City (IUH), Vietnam. He is currently pursuing studies in Robotics and Intelligent Control. His interests include human-robot collaboration, as well as the research and design of mechanical tools and grippers for articulated robots in various applications. He is enthusiastic about robotics and its practical integration into real-world systems. He can be contacted at khanh371554@gmail.com.

Thien Bui Thanh is an undergraduate student at the Faculty of Electrical Engineering Technology (FEET), Industrial University of Ho Chi Minh City (IUH), Vietnam. He is currently pursuing studies in Robotics and Intelligent Control. His academic interests focus on optimal control and model predictive control, with a strong emphasis on their practical applications in robotics. He can be contacted at 23657051.thien@student.iuh.edu.vn.

Quy Vo Quoc is an undergraduate student at the Faculty of Electrical Engineering Technology (FEET), Industrial University of Ho Chi Minh City (IUH), Vietnam. He is currently studying in the field of Automatic Control Engineering. His interest spans across the field of industrial automation, robotics in industrial settings as well as mobile robots. He can be contacted at 23732001.quy@student.iuh.edu.vn.