ISSN 1590-8844
International Journal of Mechanics and Control, Vol. 25, No. 01, 2024
https://doi.org/10.69076/jomac.2024.0016

PLANNING HANDWRITING FOR A 2-DOF PLANAR ROBOT ARM

Carlos Sánchez-López*

* Department of Electronics, Autonomous University of Tlaxcala, Mexico

ABSTRACT

The design of robotic arms with the ability to write words in handwritten format is a challenging task, involving image processing techniques, modeling and control algorithms. This paper introduces a methodology that facilitates handwriting planning for a two degree of freedom (2-DoF) planar robotic arm. From a handwritten word made with pencil on a sheet with white background, the image is acquired and processed in order to extract the word in (***x,y***) coordinates on the cartesian space. Afterwards, the collected data are used to compute each rotation angle related with each link of the 2-DoF planar robot arm by using its inverse kinematic model. A simple prototype is built with TowerPro MG995 servomotors and the Arduino DUE development board is used for controlling each servomotor. As a consequence, the 2-DoF planar robot arm can reproduce the handwriting on a blackboard. Experimental tests validate the proposed methodology, demonstrating its suitability, simplicity and effectiveness.

Keywords: Handwriting, 2-DoF planar robot arm, Forward/Inverse kinematic model, reconfigurable hardware, image acquisition

1 INTRODUCTION

Robotic arms have a wide range of applications in both: manufacturing and service industries [1, 2, 3]. In any case, robotic arms are replacing humans since they can work tirelessly and more efficiently [4, 5, 6]. Due to these reasons, universities, research institutes and companies develop robots with the ability to imitate some functions of humans [7, 8]. One main function that humans can easily perform, although still difficult for robots, is handwriting. In this sense, [9, 10, 11] describe this topic as a redundancy problem and offer the possibility of achieving human-like movements in a robot arm, applying the biomechanical principles that humans use. In [12, 13], a modification of the primitive formulation of dynamic motion is used to match the sequence of strokes associated with human handwriting. Although the proposed formulation is numerically efficient, no information was provided about of the robot used and its constraints. In [14], the Kinect device was used for detecting human gestures and hence, each stroke of a Chinese character is first captured through a human's straight right arm. Once the strokes were captured, they are processed to control the robotic arm [15]. However, the developed system was limited to detect simple human gestures and to reproduce a Chinese character, several strokes are necessary, making this task tedious and complex. In [16] the same methodology suggested in [14] was used, but here a database were used and, hence with less computational complexity. On the contrary, [17] uses the inverse kinematic model of a 5-DoF robot arm to reproduce the academic certificate with handwriting signature. Although the target was achieved, several custom CAD software must be used, which increases the manufacturing cost. Similarly in [18], the NAO humanoid robot was used as platform to test the derived forward kinematic algorithms. However, each handwritten letter not only is limited to 9 points, but the humanoid robot must also move to the right to write the next letter. Other studies use imitation learning [19] and other learning techniques have also been developed such that the robot learn from human writing [20] by using image processing techniques, stroke trajectory fitting algorithms or predefined font databases [21, 22, 23, 24, 25]. All in all, it can be seen that handwriting ability remains a challenging

Contact author: Carlos Sánchez-López*
*Clzda. Apizaquito S/N, km. 1.5, Apizaco, 90300, Tlaxcala, Mexico.
E-mail: carlsanmx@yahoo.com.mx

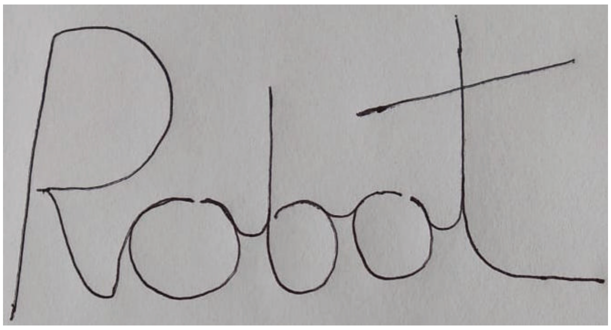


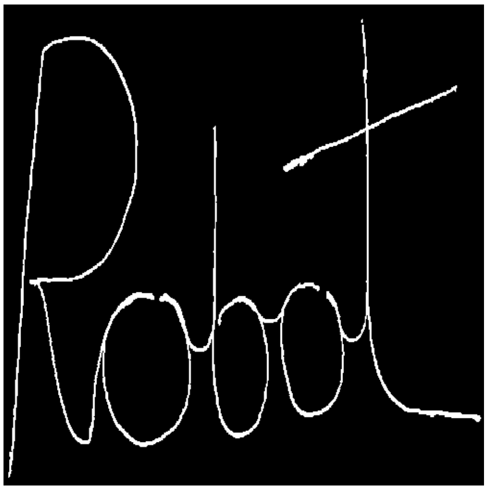
Figure 1 Image of a handwritten word to be extracted

task for robotic systems and research on this topic is still widely needed [26]. In response to this challenge, this work contributes to the development of a handwriting planning methodology for a 2-DoF planar robot arm, which uses its inverse kinematic model to control the end-effector along with image processing techniques [27, 28].

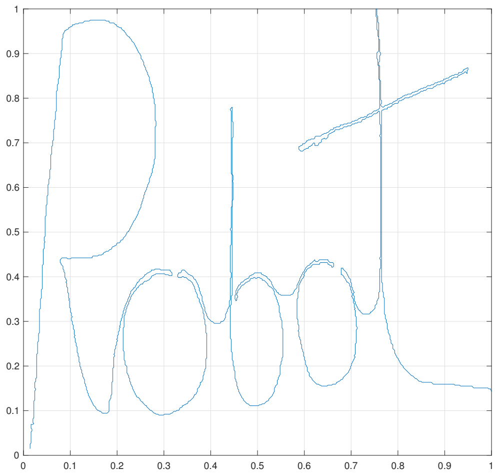
The rest of the paper is organized as follows. Section 2 describes step-by-step the process to extract from an image, the (***x,y***) coordinates of the handwritten word. The forward and inverse kinematic model of a 2-DoF planar robot arm are derived in Section 3. Section 4 focuses on the design and build of the robotic structure and its control scheme. Experimental results on the behavior of the robotic system in the time-domain are discussed in Section 5. Section 6 discuss advantages and limitations of the proposed methodology. Finally, some conclusions are summarized in Section 7.

2 DATA AQUISITION

An ability exclusive of the humans is the capacity of write words in handwriting format. Several methodologies have been reported in the literature for performing this challenging task with robots, and although this target has slightly been achieved, however, each of them has still disadvantages, such as those mentioned above [9]-[25]. Among all they, this paper use image processing techniques and hence, the handwritten word must be extract from an image. To do it, the handwritten word is first traced with pencil on a 1/8 letter-size sheet with white background and taking into account the following two constraints: 1) Do not join the internal parts of each letter and 2) The end point of the last letter should be a slightly away from the top of the letter itself, between 3mm and 5mm approximately. The camera used during the acquisition has 17Mp of resolution with aperture f/1.8 of illumination. Figure 1 shows a handwritten word at *jpg* format to be processed, where the required drawing conditions were taken into account. The image was acquired with artificial white lighting and at a distance of 10cm between the camera and the 1/8 letter-size sheet, approximately. The next step is convert the image to gray-scale and binarize the result. To filter the noise of the image



(a)



(b)

Figure 2 (a) Binarized and scaled handwritten word, (b) Upper contour extraction

and hold on the waveform information, a threshold th=0.5 was used, which was adjusted through experimentation. Results of this process are illustrated in Figure 2(a) and the image was scaled to 500×500 pixels using bi-cubic interpolation, where the black background is represented by a logical **0** and the white background is represented by a logical **1**. The scaling of the image was also adjusted through experimentation. From Figure 2(a), the contour of the binarized image is obtained and divided as upper and lower contour. For this case, all data of the upper contour are only required and must be horizontally and vertically normalized, as illustrated in Figure 2(b). Note that the upper

Table I - Matlab code to extract the handwritten word of Figure 1

```
clear all; close all; clc;
A1=rgb2gray(imread('Robot.jpg')); th=0.5;
A2=imresize(∼imbinarize(A1,th),[500 500]); figure; imshow(A2);
for i=1:size(A2)
    if find(A2(:,i),1)∼=0
        c=i;r=find(A2(:,c),1,'last');break;
    end
end
xy=bwtraceboundary(A2,[r,c],'N',8);
for i=size(A2):-1:1
    if find(A2(:,i),1)∼=0 r=i;break; end
end
for i=max(size(xy)):-1:1
    if xy(i,2)==r c=i;break; end
end
for i=1:c
    xyu(i,1)=xy(i,2);xyu(i,2)=max(size(A2))-xy(i,1);
end
Xoff=0;SizeX=1;Yoff=0;SizeY=1;
xys(:,1)=Xoff+SizeX*xyu(:,1)/max(xyu(:,1));
xys(:,2)=Yoff+SizeY*xyu(:,2)/max(xyu(:,2));
figure;plot(xys(:,1),xys(:,2));grid on
```

contour data are the (***x,y***) coordinates of the handwritten word on the Cartesian space. Here, ***x*** and ***y*** are data vectors. In order to improve the previous description, Table I shows the Matlab programming code used for generating of Figure 2.

3 FORWARD AND INVERSE KINEMATIC MODEL

According to Figure 3, the position coordinate of the manipulator end-effector is given by (*x,y*) coordinates. For the *x*-abscissa, it is given by

$$x = a_1 cos(q_1) + a_2 cos(q_1 + q_2) \tag{1}$$

whereas for *y*-ordinate is approximated as

$$y = a_1 sin(q_1) + a_2 sin(q_1 + q_2) \tag{2}$$

where $a_{1,2}$ and $q_{1,2}$ are the lengths and rotation angles of first and second link of the robot arm, respectively. Applying Pythagoras theorem to Figure 3, one obtain

$$r^2 = x^2 + y^2 \tag{3}$$

and considering the cosine rule

$$r^2 = a_1^2 + a_2^2 - 2a_1a_2cos(\alpha) \tag{4}$$

hence

$$cos(\alpha) = \frac{a_1^2 + a_2^2 - r^2}{2a_1a_2} = \frac{a_1^2 + a_2^2 - x^2 - y^2}{2a_1a_2} \tag{5}$$

Since $\alpha = \pi - q_2$ and $cos(\pi - q_2)$=-$cos(q_2)$, we get

$$q_2 = \pm arccos\left(\frac{-a_1^2 - a_2^2 + x^2 + y^2}{2a_1a_2}\right) \tag{6}$$

On the other hand, from Figure 3 we also observe that

$$tan(\beta) = \frac{a_2 sin(q_2)}{a_1 + a_2 cos(q_2)} \tag{7}$$

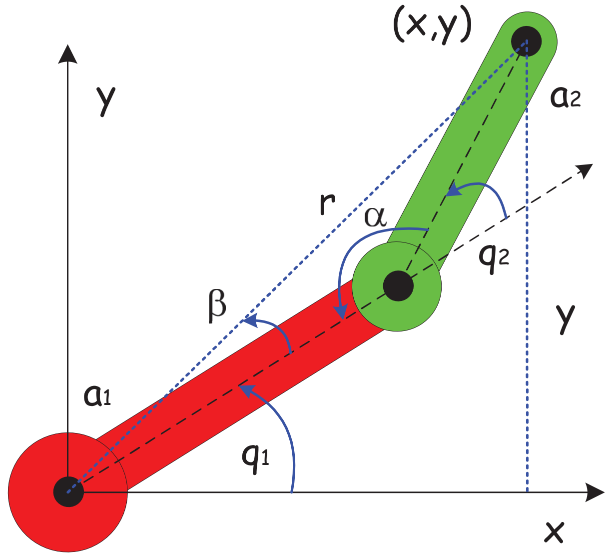


Figure 3 2-DoF planar robot arm geometry

and

$$\beta + q_1 = arctan\left(\frac{y}{x}\right) \tag{8}$$

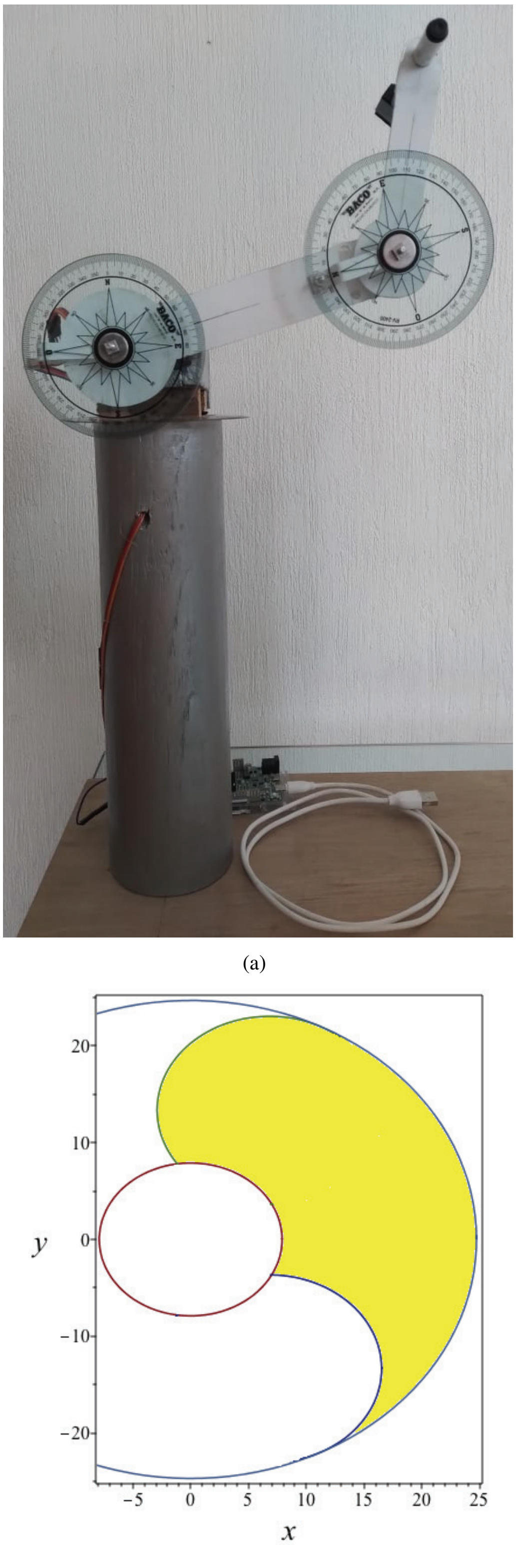
Merging (7) and (8), we obtain

$$q_1 = arctan\left(\frac{y}{x}\right) \mp arctan\left(\frac{a_2 sin(q_2)}{a_1 + a_2 cos(q_2)}\right) \tag{9}$$

Clearly (1) and (2) describe the forward kinematic model whereas (6) and (9) describe the inverse kinematic model [4, 15, 28]. The use of the positive sign in (6) along with the negative sign in (9) describes the *elbow-down* configuration, as depicted in Figure 3, whereas if the opposite signs are selected, then it gives the *elbow-up* configuration. It is worth mentioning that the rotation angles are given in rad/s and $a_{1,2}$ are dimensionless.

4 2-DOF PLANAR ROBOT ARM DESIGN

A prototype was built by using two Tower-Pro MG995 servomotors and following the scheme of Figure 3. The first link has a length of a_1=15cm, width=3cm and with a circle of radius=3cm on the left side, whereas the second link has a length of a_2=10cm, width=2cm and with a circle of radius=2cm also on the left side. All parts of the robot arm are mounted on a circular base of radius=4cm and high=30cm, as illustrated in Figure 4(a). According the characteristics of the servomotor, it has a rotation angle from 0° to 180°, with minimum resolution of 1°. However, due to the limitations of the mechanical system of the robotic arm built, the rotation angle for both links was limited from 30° to 160°. As a consequence, the workspace of the prototype is limited as shown in Figure 4(b), and the handwritten word to be reproduced must be included



(a)

(b)

Figure 4 (a) 2-DoF planar robot arm built, (b) Maximum workspace

into this workspace. In this sense, all (***x,y***) normalized coordinates obtained in Section 2 can be zoom-in, zoom-out, up or down, multiplied by a constant and add an offset to each data vector, respectively, as $\boldsymbol{x}_s$=X_{off}+$\boldsymbol{x}Size_x$ and $\boldsymbol{y}_s$=Y_{off}+$\boldsymbol{y}Size_y$. Moreover, from (6), (9) and taking into account the physical characteristics of the prototype shown in Figure 4 at *elbow-down* configuration, (6) and (9) were adjusted to

$$q_{2d} = 180° - \frac{180°}{\pi}q_2 \tag{10}$$

$$q_{1d} = 90° - \frac{180°}{\pi}q_1 \tag{11}$$

The units for both rotation angles are now degrees and whereas q_{1d} varies from 30° (starting from the top) to 160° (ending at the bottom), q_{2d} varies from 30° (starting from the left side) to 160° (ending on the right side). According to Section 2, all (***x,y***) normalized coordinates of the handwritten word can be extracted from an image and using (6), (9), (10) and (11), the rotation angle for each link can now be computed according to each pair of ($\boldsymbol{x}_s$,$\boldsymbol{y}_s$) coordinates. However, to control each servomotor by using the Arduino DUE development board, so that its repeatability and accuracy are hold on, the rotation angles in degree must be converted to microseconds [29]. To overcome this issue, surrogate functions are used. This approximation technique establishes a relation between the rotation angle, considered as design variable, and its numerical value in microseconds, i.e., $q_{1\mu}(q_{1d})$ and $q_{2\mu}(q_{2d})$. Since there are several servomotors on the market with particular characteristics, a library of surrogate functions associated with each of them can be generated and calibrated from measurement data. As a consequence, they not only able to be used at any time, but the use of surrogate functions is also computationally cheap and it is not needed to recompute $q_{1\mu}(q_{1d})$ and $q_{2\mu}(q_{2d})$. Thus, by applying cubic spline interpolation to the measured data, the surrogate functions are generated as illustrated in Figure 5. Once all minimum requirements have been satisfied, we are now in conditions to verify the proposed planing methodology, which consists of extracting a handwritten word from an image until its reproduction through a 2-DoF planar robot arm.

5 EXPERIMENTAL TESTS

Experimental tests were carried out using the built prototype and executing on Matlab environment, the data acquisition code gives in Table I, (6), (9), (10), (11) along with the surrogate functions described in Figure 5. It is important to emphasize that after the data acquisition from an image, all (***x,y***) normalized coordinates must be updated and stored as ($\boldsymbol{x}_s$,$\boldsymbol{y}_s$) coordinates, so that they are into the workspace depicted in Figure 4(b). As a consequence, two variables of the type *array* are generated, i.e., q_{1u} and q_{2u}, which

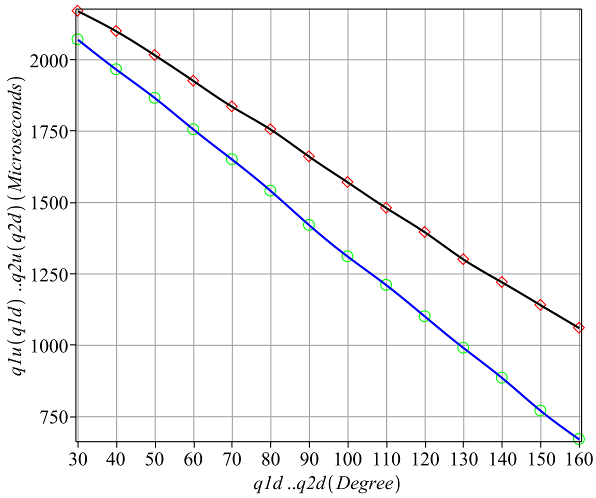


Figure 5 Surrogate functions used to convert the rotation angle in degree to microseconds. Green circles are the $(\boldsymbol{x}_s,\boldsymbol{y}_s)$ coordinates of q_{1d} whereas solid blue line is $q_{1\mu}(q_{1d})$. Red diamonds are the $(\boldsymbol{x}_s,\boldsymbol{y}_s)$ coordinates of q_{2d} whereas solid black line is $q_{2\mu}(q_{2d})$

have all the information on the rotation angles of each servomotor in microseconds, according to the handwritten word extracted from an image and to the valid workspace due to the physical limitations of the mechanical system of the 2-DoF planar robot arm. For flexibility of programming, build and verification, the Arduino DUE development board was used to control the rotation angle of each servomotor [30]. Nevertheless, other reconfigurable hardware development platforms with better features can be used to take advantage of their full capacity. Thus, whereas Table II shows the programming code on the Arduino board, Figure 6 displays the handwritten word reproduced by the 2-DoF planar robot arm on a blackboard. For this example, X_{off}=8, $Size_x$=15, Y_{off}=2 and $Size_y$=9 were used and Table II shows only the first five and last four data for q_{1u} and q_{2u}, out of a total of 214 data. Nonetheless, Figure 7 shows the full behavior of q_{1u} and q_{2u} used to get Figure 6. As one can observe, the proposed handwriting planning methodology is simple, effective and can not only be used to reproduce another handwritten word, but can also be used for path planning and tracking [31, 32] .

6 DISCUSSION

The proposed handwriting planning methodology described herein is relatively simple and easy to code on reconfigurable hardware. Based on obtained results, it may be claimed that all numerical simulations done on Matlab environment and experimental tests conducted by Arduino DUE development board give satisfactory results. Unlike other methodologies reported in the literature [9]-[25], image processing techniques were used for extracting, from an image at *jpg*

Table II - Programming code for Arduino DUE development board

```
#include <Servo.h>
Servo Q1,Q2;
float q1u[214]={1159.05948874914, 1179.59280130950, 1196.65477149230, 1216.01344658085,
                1234.14062872014, ..., 1269.45978582690, 1287.26066048404, 1308.26605166436};
float q2u[214]={2153.02893433695, 2146.63753054631, 2140.22655890366, 2134.02177099947,
                2127.27685696610, ..., 1386.86306283123, 1331.89159993661, 1275.25297697264};
int Sq1=sizeof(q1u)/sizeof(q1u[0]);
void setup()
{
        Serial.begin(9600);
        Q1.attach(9,670,2070);          // Min=670µs⇒160º, Max=2070µs⇒30º
        Q1.writeMicroseconds(1420); // 1420µs⇒90º
        Q2.attach(10,1060,2170);        // Min=1060µs⇒160º, Max=2170µs⇒30º
        Q2.writeMicroseconds(1660); // 1660µs⇒90º
        delay(2000);
}
void loop()
{
        Q2.writeMicroseconds(q2u[0]);
        Q1.writeMicroseconds(q1u[0]);
        delay(1000);
        for (int i=0;i<Sq1;i++)
        {
                Q2.writeMicroseconds(q2u[i]);
                delay(20);
                Q1.writeMicroseconds(q1u[i]);
                delay(20);
        }
}
```

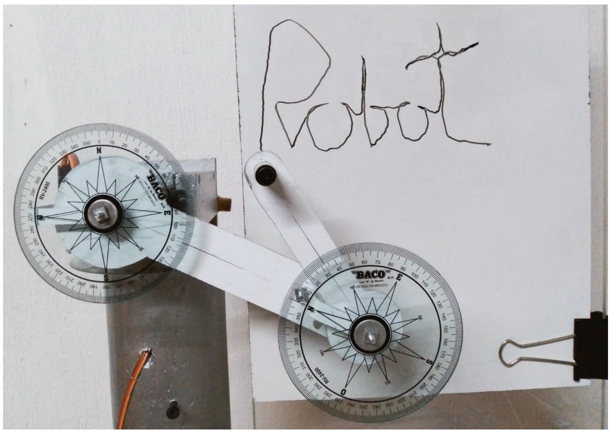


Figure 6 Handwritten word reproduced with the 2-DoF planar robot arm of Figure 4(a)

format, the waveform of the handwritten word. Furthermore, the inverse kinematic model of the robot arm along with surrogate functions were used to compute the rotation angle, given in microseconds, of each link. However, one must take care of that the handwritten word to be reproduced on the blackboard must be into the valid workspace. This last issue can be overcome by updating the X_{off}, $Size_x$, Y_{off} and $Size_y$ variables. On the other hand, the proposed handwritten planning methodology for robot arms offers several advantages. First, the methodology is very simple and we assure that any handwritten word, extracted from an image, can be reproduced by the 2-DoF planar robot arm on a blackboard. Second, (10) and (11) can easily be adjusted according to the workspace defined by the user and taken into account the physical limitations of the mechanical system of the robotic arm. Third, experimental tests were shown, validating the theory. Limitations of the proposed planning methodology are: a) It is necessary to develop a technique

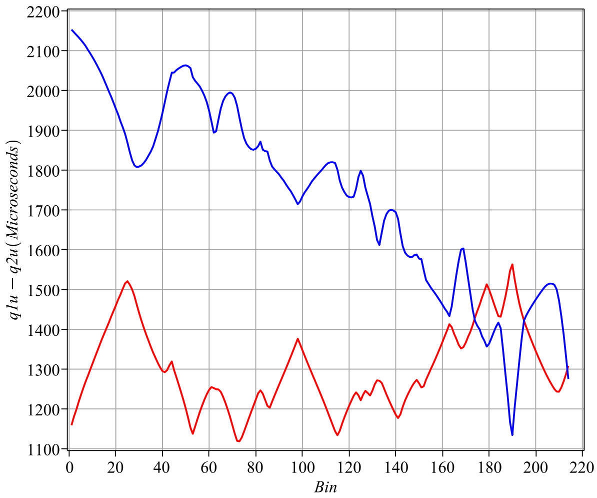


Figure 7 Behavior of q_{1u} (red solid line) and q_{2u} (blue solid line) to reproduce Figure 6

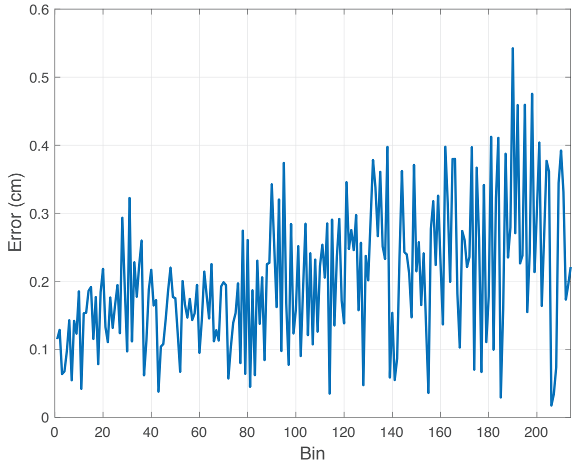


Figure 8 Computed errors for each (*x*,*y*) denormalized coordinate between Figure 2(b) and Figure 6.

for the automatic selection of the threshold associated with each image to be processed. b) It is necessary to develop an automatic technique to place the ($\boldsymbol{x}_s$,$\boldsymbol{y}_s$) coordinates into the valid workspace. c) Small size handwritten words cannot be reproduced. This issue is related to the minimum resolution of 1° by each servomotor. Furthermore, this is also the main reason why the experimental results depicted in Figure 6 have a deviation compared to the numerical results shown in Figure 2(b). This is because there are small angles less than 1° in each data vector q_{1u} and q_{2u} that the servomotors cannot reproduce. Therefore, it is advisable to use servomotors with high resolution or stepper motors, even if the writing speed decreases. Therefore, the error for each (*x*,*y*) denormalized coordinate were computed between Figure 2(b) and Figure 6, as shown in Figure 8. In this figure, the maximum errors found are in the range of 0.4cm-0.6cm, which is related to the letter ***t***.

7 CONCLUSIONS

A handwriting planning methodology for a 2-DoF planar robot arm has been presented. Basically, the handwritten word is first traced with pencil on a 1/8 letter-size sheet with white background and afterwards reproduced with the 2-DoF robot arm. The Matlab script to extract the handwritten word from an image is described in Table I and the Arduino code to control the servomotors of the robot arm is described in Table II. Experimental results were gathered, showing good performance, robustness and effectiveness. Finally, as can be seen throughout this paper, the developed handwriting planning methodology provides a useful tool to reproduce any handwritten word, from an image at *jpg* format until its experimental verification on a blackboard, using reconfigurable hardware and the 2-DoF planar robot arm depicted in Figure 4. On the other hand, to reduce the error in Figure 8, high-resolution servomotors or stepper motors with multiple steps should be used. Other future work is related to the use of the proposed platform to perform other activities with the robot arm, such as those described in [26] together with collaborative robots [6].

REFERENCES

[1] Callegari M., Carbonari L., Palmieri G. and Palpacelli M., Functional Design of a Manipulator for the Automation of Laboratory Precision Tasks,' *International Journal of Mechanics and Control*, Vol. 21, No. 02, pp. 29–37, 2020.

[2] Manuello Bertetto A. and Prete A., Perspective on Saffron Spice Separation Based on Controlled Fluid Dynamic System and Computer Vision, *International Journal of Mechanics and Control*, Vol. 22, No. 01, pp. 161–166, 2021.

[3] Pagano S., Russo R. and Savino S., A gluing process driven by a vision guided robotic system for footwear industry, *International Journal of Mechanics and Control*, Vol. 22, No. 02, pp. 43–52, 2021.

[4] Krastev R., Motion Path Control of Redundant Robot Arms with Jacobian Singularities, *International Journal of Mechanics and Control*, Vol. 20, No. 01, pp. 15–22, 2019.

[5] Valsamos C., Wolniakowski A., Miatliuk K. and Moulianitis V.C., Optimal Placement of a Kinematic Robotic Task for the Minimization of Required Joint Velocities, *International Journal of Mechanics and Control*, Vol. 20, No. 01, pp. 3–14, 2019.

[6] Boschetti G., Faccio M. and Minto R., A framework for the integration of traditional and collaborative robotics, *International Journal of Mechanics and Control*, Vol. 22, No. 02, pp. pp. 3–14, 2021.

[7] S. Schaal, Is Imitation Learning the Route to Humanoid Robots?, *Trends Cogn. Sci.*, Vol. 3, No. 6, pp. 233—242, 1999.

[8] R. Dillmann, Teaching and learning of robot tasks via observation of human performance, *Robot. Auton. Syst.*, Vol. 47, No. 2-3, pp. 109–116, 2004.

[9] V. Potkonjak, M. Popovic, M. Lazarevic and J. Sinanovic, Redundancy problem in writing: From human to anthropomorphic robot arm, *Trans. Syst. Man & Ciber. Part B*, Vol. 28, No. 6, pp. 790–805, 1998.

[10] V. Potkonjak, D. Kostic, S. Tzafestas, M. Popovic, M. Lazarevic and G. Djordjevic, "Human-like behavior of robot arms: general considerations and the handwriting task-Part II: the robot arm in handwriting," *Robotics and Comp. Intgr. Manu.*, Vol. 17, No. 4, pp. 317–327 (2001).

[11] V. Potkonjak, "Robotic handwriting," *Int. J. Hum. Robotics*, Vol. 2, No. 1, pp. 105–124 (2005).

[12] T. Kulvicius, K. Ning, M. Tamosiunaite and F. Wörgötter, "Joining movement sequences: Modified dynamic movement primitives for robotics applications exemplified on handwriting," *Tran. Robotics*, Vol. 28, No. 1, pp. 145–157 (2012).

[13] W. Si, N. Wang and C. Yang, "Composite dynamic movement primitives based on neural networks for human–robot skill transfer," *Neural Comput. & Applic.*, (2021).

[14] F. Chao, F. Chen, Y. Shen, W. He, Y. Sun, Z. Wang, C. Zhou and M. Jiang, "Robotic free writing of Chinese characters via human–robot interactions," *Int. J. Hum. Robotics*, Vol. 11, No. 1, 1450007 (2014).

[15] A. Botta, P. Cavallone, L. Tagliavini, G. Colucci, L. Carbonari and G. Quaglia, "Modelling and simulation of articulated mobile robots," *International Journal of Mechanics and Control*, Vol. 22, No. 02, pp. 15–26 (2021).

[16] R. Wu, F. Chao, C. Zhou, Y. Huang, L. Yang, C-M. Lin, X. Chang, Q. Shen and C. Shang, "A developmental evolutionary learning framework for robotic Chinese stroke writing," *IEEE Trans. on Cog. Develop. Syst.*, Early Access, (2021).

[17] M.F. Crainic, S. Preteil, L.A. Sandru and V. Dolga, "Secure handwriting using a robot arm for educational purpose," *Int. Confer. Method & Modules Auto. & Robotics*, Vol. 1, pp. 1–6 (2014).

[18] S. Sayem and F. Ahmed, "A forward kinematics approach towards humanoid robot handwriting," *Int. Confer. Electrical Electronic Eng.*, Vol. 1, pp. 1–4 (2017).

[19] N. García, J. Rosell and R. Suárez,"Motion planning by demonstration with human-likeness evaluation for dual-arm robots," *Trans. on Systems, Man and Cyber.: Systems*, Vol. 49, No. 11, pp. 2298—2307 (2019).

[20] F. Xiong, B. Sun, X. Yang, H. Qiao, K. Huang, A. Hussain and Z. Liu, "Guided policy search for sequential multitask learning," *Transactions on Systems, Man and Cyber.: Systems*, Vol. 49, No. 1, pp. 216—226 (2019).

[21] Y. Man, C. Bian, H. Zhao, C. Xu and S. Ren, "A kind of calligraphy robot," *Int. Confer. on Inf. Sc. Inter. Sc.*, Vol. 1, pp. 635—638 (2010).

[22] X. Ma, Q. Kong, W. Ma and X. Zhang, "4-DOF lettering robot's trajectory planning," *Mech. Eng. Automa*, Vol. 165, No. 10, pp. 161—163 (2010).

[23] H.I. Lin and Y.C. Huang, "Visual matching of stroke order in robotic calligraphy," *Int. Confer. on Advanced Robotics*, Vol. 1, pp. 459–464 (2015).

[24] N.S. Zamani, M.N. Mohammed, M.I. Abdullah and S. Al-Zubadai, "A new developed technique for handwriting robot," *Int. Confe. Aut. Control & Intelligent Syst.*, Vol. 1, pp. 264–267 (2019).

[25] X. Wu, C. Yang, Y. Zhu, W. Wu and Q. Wei, "An integrated vision-based system for efficient robot arm teleoperation," *Industrial Robot*, Vol. 48, No. 2, pp. 199–210 (2020).

[26] Scalera L., Maset E., Seriani S., Gasparetto A. and P. Gallina, Performance evaluation of a robotic architecture for drawing with eyes, *International Journal of Mechanics and Control*, Vol. 22, No. 02, pp. 53–60, 2021.

[27] D. Kostic, B. De Jager, M. Steinbuch and R. Hensen, Modeling and identification for high-performance robot control: An RRR-robotic arm case study, *Tran. Control, Syst. Tech.*, Vol. 12, No. 6, pp. 904–919 (2004).

[28] Martorelli M., Rossi C., Savino S. and Staiano G., A Contactless Robot Kinematic Calibration Method by Digital Photogrammetry, *International Journal of Mechanics and Control*, Vol. 16, No. 02, pp. 9–16, 2015.

[29] Colombo F., Marino A., Mazza L. and Raparelli T., Inverted pendulum stabilization based on Arduino drive control," *International Journal of Mechanics and Control*, Vol. 23, No. 01, pp. 105–114, 2022.

[30] C. Sánchez-López, An experimental synthesis methodology of fractional-order chaotic attractors, *Nonlinear Dyn.* Vol. 100, No. 4, pp. 3907–3923, 2020.

[31] K. Althöfer, D.A. Fraser and G. Bugmann, "Rapid path planning for robotic manipulators using an emulated resistive grid," *Electronics Letters*, Vol. 31, No. 22, pp. 1960–1961, 1995.

[32] C. Hernández-Mejía, H. Vázquez-Leal and D. Torres-Muñoz, "A Novel Collision-Free Path Planning Modeling and Simulation Methodology for Robotical Arms Using Resistive Grids," *Robotica*, Vol. 38, No. 7, pp. 1176–1190, 2019.