Research Article

Adaptive Control Design for Human Handwriting Process Based on Electromyography Signals

Imane Mahmoud, Ines Chihi, and Afef Abdelkrim

Laboratory of Research in Automation (L.A.R.A), National School of Engineers of Tunis, BP 37, Le Belvédère, Tunis El Manar University, Tunis 1002, Tunisia

Correspondence should be addressed to Ines Chihi; chihi4ines@hotmail.fr

Received 20 August 2019; Revised 4 January 2020; Accepted 10 February 2020; Published 11 March 2020

Academic Editor: Mahardhika Pratama

Copyright © 2020 Imane Mahmoud et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The most used control approaches of hand prosthesis are based on the forearm muscle activities, named ElectroMyoGraphy signal (EMG). In this sense, researchers modeled the hand writing on the plane only from two EMG signals. Based on this analysis, we can consider the hand as a robot with two arms (two degrees of freedom) moving on (x, y) plane. However, these signals are very sensitive to many disturbances and are generally unpredictable in time, type, and level. Based on forearm EMG signals, this work aims to propose an adaptive hand–robot control design to generate handwriting. As a first step, we develop the application of the classic proportional integral structure (PI). The PI controller was applied to generate different essays of handwritten graphic traces in one-writer case and multiwriter case. Both cases have presented unsatisfactory results in generating cursive letters and forms. Indeed, we propose, as a second approach, an adaptive PI controller with varying Integral $K_i$ gain, according to EMG signals, in order to deal with operation changes.

1. Introduction

In the last decades, several investigators are interested in improving the quality of hand prosthesis robots to make them functional, more convenient to use. These improvements also concern movements requiring precision, such as typing text on a key board, fluffing an apple, or even handwriting with conserving the individual characteristics of the writer (roundness or sharpness, inclination, regular or irregular spacing between letters, etc.).

The increase in functionality of hand prosthesis is mainly based on the progression of the control strategies. The most used control approach is based on the amplifier electrical activity of the muscles, EMG signal, which allows encoding directly the orders generated by the brain [1–4].

The wealth of information of these biological signals leads many researchers to propose approaches based on the muscular activity control. For example, in order to control an articulated mechanism, Jacobsen et al. proposed a classical PID [5]. Kawanishi et al. developed in [6] a fuzzy logic controller for position control of the biomimetic robot finger designed by Hristu et al. [7]. Based on classification technique, in [8], EMG signals of ten muscles were used to control open and close movements of an artificial hand with four fingers, used to grasp objects. In [9], PID was developed to control upper-limb prosthesis in real-time EMG signal processing.

Comparing to other controls (fuzzy logic, sliding mode, and backstepping), PID- and PI-based techniques are considered as the preferable and the most used controllers in the industry due to the simplicity and the ease of implementation. However, they have some drawbacks. Indeed, parameters tuning these controllers are important to ensure their effectiveness [10]. In this sense, different methods were proposed to optimize the PI parameters such as Ghazali et al. improved in [11] the performance of an upper-limb prosthesis using a fuzzy-PID controller. Engeberg proposed in [12] a hybrid adaptive force-position controller of a prosthetic hand based on EMG information to control some simple hand movements.
In the literature, some researchers show that the handwriting on the plane \((x, y)\) can be modeled only from two forearm muscle activities, named “Abductor Pollicis Longus” and “Extensor Capri Ulnaris.” The first assures vertical movement, and the second assures horizontal displacement [10, 13, 14].

Based on this analysis, we can consider the hand as a robot modeled by two arms. One end of the first arm is connected to a first articulation for horizontal movements. The end of the second arm is linked to a second articulation, for vertical movements, and its other end to a pen moving on \((x, y)\) plane. This robot arm is dedicated to mimic human handwriting process from two forearm muscle activities in order to generate manuscript shapes like cursive letters and geometric forms.

However, EMGs are unpredictable in time, type, and level. The characteristics of the muscle activities are easily affected by many factors, such as recording over layers of muscles, fat and tissue, abrupt changing of the electrode positions, sweat of the prosthesis wearer at the recording site, changes of the impedance of the electrode, filtering method, noise of measure, disturbances, and user fatigue [15–17]. All these conditions lead to inaccurate identification of user intent and threaten the prosthesis control reliability [18–22].

The individual characteristic of the handwriting process leads to think of a control design to represent and command this complex process that depends on the unpredictable characteristic of the muscle activities and different internal and external factors affecting the production of graphic traces [22]. Inspired from classical gain scheduling allowing controlling nonlinear and complex systems from its decomposition into different linear controllers, in this paper, we present an adaptive PI controller applied to the geometric models of an arm robot to generate handwriting from two EMG signals. The proposed controller is based on a decision function which is based on EMG signals, considered as a condition to choose the appropriate gains to force the system to follow a desired performance, even the unpredictable characteristic of the muscle activities, due to internal or external factors. Otherwise, scheduling of the proposed adaptive gain is based on averages of different electromyography signals; each one is associated to a specific handwriting model.

A database, collected from an experimental approach proposed by Sano in 2003, was considered to validate the present work [23]. From this experimentation, several nonconventional approaches, based on the concepts of artificial neural networks and fuzzy logic, are proposed to characterize the handwriting process [23, 24]. Other models based on the velocity profile are presented in [25, 26]. The complexity of the handwriting process explains the unsatisfactory results obtained in the previous research studies.

The remainder of this paper is organized as follows: the first section deals with the mostly used methods of experimental approaches in generating graphic traces. The second section studies the characterisation of the robot arm: direct and inverse geometric models. The handwriting control design is presented in the third section: this section deals with the application of the classic P controller as well as the PI controller and shows the results obtained for different essays of handwritten traces. While presenting the drawbacks of the classic PI controller, the proposed adaptive PI controller is presented. Conclusion and perspectives are in the last section.

## 2. Handwriting Data Acquisition

The generation of graphic shapes is regarded as a natural process of a complicated cognitive event. In fact, the brain processes the location information of the pen-tip, sent by eyes. This information is analysed and evaluated by an intelligent control system to send an order to the muscles of the forearm to move the hand to the desired new position [27].

While the handwriting process, the movements generated can be described like migration in the two-dimensional space of the writing plan. Numerous research studies, performed in the literature, proved that the natural constituent of the graphic trace coincides to space deportations of the pen during the creation of the trajectory [26–28].

Despite the complexity of the effector systems including the articulations of the shoulder, the elbow, the wrist, and the hand, for a total of forty-three muscles, the study conducted by Pirabakaran and Becerra [14] identified four principal muscles for the hand control. This paper is based on two forearm muscles to characterize the handwriting process. The first one, named “Abductor Pollicis Longus” and the second one “Extensor Capri Ulnaris,” Figure 1.

An experimental study was conducted at the University of Hiroshima City. These writers produce several Arabic letters and some basic geometric shapes written in different orientations, Table 1. The type of graphic traces is carefully chosen. Indeed, based on the analysis of Pirabakaran and Becerra who confirmed that when a person has to produce lines or forms in other nonpreferred directions, the performance would be less precise, more unstable, and is manifested particularly by the presence of shakiness [28].

In this experiment, the Japanese participants had to produce Arabic letters, which influence, inevitably, on the speed and preferential direction of writers. In this case, the production of geometric shapes is faster.

The muscles of the forearm, intervening in the act of handwriting, are located directly under the skin, allowing the use of surface electrodes to record the EMG signals. To characterize this biological process, Manabu proposed in [23] an experimental approach for recording, at the same time, the graphic traces’ coordinates in the \((x, y)\) plane and electromyography signals from the front arms involved in the production of handwriting movements. These signals are measured from surface electrodes to compute integrated muscle activity of two forearm muscles, IEMG1 and IEMG2. These signals are obtained by many signal processing techniques, Figure 2.

Figures 3 and 4 show the experimental setup and equipment used for the achievement of this approach. This experiment requires the following:

(i) Computer to record the \(x\) and \(y\) positions and the pen tip pressure on the writing plan

(ii) Numerical table: “WACOM, KT-0405-RN”, used as writing plan
3. Characterization of Handwriting Process

In this paper, the proposed control approach is applied to a MIMO handwriting model, developed in [28]. This model is based on multimodel approach to characterize the studied process. This section is devoted to present the handwriting model used in the proposed adaptive control.

In [28], a mathematical model was proposed to characterize the pen-tip movement during the handwriting act. As it is shown in Figure 5, this model is linear with two inputs (IEMG signals) and two-output pen-tip coordinates (x, y).

Multimodel approach is based on the combination of several simple submodels. The contribution of each one is determined through a normalized residual.

Equations (1) and (2) illustrate the linear handwriting model. This model is then used to develop the multimodel handwriting structure, (3) and (4). Recursive least squares algorithm (RLS) is used to estimate model’s parameters.

\[ x_i(k) = \sum_{i=1}^{4} \hat{a}_{ix} y_i(k - i) + \sum_{i=1}^{4} \hat{b}_{ix} x_i(k - i) \]
\[ + \sum_{i=1}^{5} \hat{c}_{ix} e_i(k - i + 1) \]
\[ + \sum_{i=1}^{5} \hat{d}_{ix} e_2(k - i + 1), \]

\[ y_i(k) = \sum_{i=1}^{4} \hat{a}_{iy} x_i(k - i) + \sum_{i=1}^{4} \hat{b}_{iy} y_i(k - i) \]
\[ + \sum_{i=1}^{5} \hat{c}_{iy} e_i(k - i + 1) \]
\[ + \sum_{i=1}^{5} \hat{d}_{iy} e_2(k - i + 1), \]

with \( x_i \) and \( y_i \): estimated pen-tip, coordinates, \( \hat{a}_{ix}, \hat{b}_{ix}, \hat{c}_{ix}, \hat{d}_{ix} \): estimated parameters of \( x_i \) vector, \( \hat{a}_{iy}, \hat{b}_{iy}, \hat{c}_{iy}, \hat{d}_{iy} \): estimated parameters of \( y_i \) vector, and \( e_1 \) and \( e_2 \): IEMG signals.

\[ x_d(k) = \sum_{i=1}^{n} \left( 1 - \frac{d_{x_i}(k)}{\sum_{j=1}^{n} d_{x_j}(k)} \right) x_i(k), \]
\[ y_d(k) = \sum_{i=1}^{n} \left( 1 - \frac{d_{y_i}(k)}{\sum_{j=1}^{n} d_{y_j}(k)} \right) y_i(k), \]

where \( d_{x_i} \) and \( d_{y_i} \) are the residues of each submodel of the multimodel structure, \( x_d \) and \( y_d \) are the handwriting multimodel outputs, and \( n \) is the number of submodels of the multimodel structure.

3.1. Handwriting Process. The handwriting process, Figure 6, generates the articulations \( \theta_1 \) and \( \theta_2 \) from two control laws \( U_x \) and \( U_y \), outputs of the API control technique, specific to each letter or form considered.

The model structure is given as follows:

<table>
<thead>
<tr>
<th>Name of the form</th>
<th>Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizontal line (left right left)</td>
<td>Horizontal line (right left right)</td>
</tr>
<tr>
<td>Vertical line (left right left)</td>
<td>Vertical line (right left right)</td>
</tr>
<tr>
<td>Circle (1) (clockwise)</td>
<td>Circle (2) (left)</td>
</tr>
<tr>
<td>Triangle (1) (clockwise)</td>
<td>Triangle (2) (left)</td>
</tr>
<tr>
<td>Arabic letter (HA)</td>
<td>Arabic letter (AYN)</td>
</tr>
</tbody>
</table>

[iii] Data logger: “TEAC AR-C2EMG1”

[iv] Surface electrodes: “Medico test, Blue Sensor N-00-S”

Figure 4 shows the positioning of the electrodes on the writer’s arm, and the electrodes, indicated by “ch 1,” are relative to the first muscle and those relating to the second muscle are indicated by “ch 2.”
Figure 2: Example of experimental recordings. (a) Arabic letter “HA.” (b) Displacement according to x- and y-axis EMG signals. (c) IEMG signals.

Figure 3: Experimental approach.

Figure 4: Positions of the surface electrodes ch1 and ch2 to measure EMG1 and EMG2 signals, respectively.
4. Handwriting Assistive Robot Arm Design

For a specified letter generated by the same writer in the same conditions, models proposed in the literature do not show a satisfactory concordance (for the same writer and the same type of character). These unsatisfying results led to several questions to be asked: What is the model or controller that can represent any kind of handwriting motion process? Can we find a single model to identify many writers?

In order to answer these questions, we propose geometric models (direct and inverse) to represent the handwriting motion and a novel approach to design an adaptive PI controller provided the IEMG signals of two forearm muscles. This new approach allows defining a variable gain controller adapted to different graphic traces (letters and geometric shapes) generated by the same writer.

4.1. Direct Geometric Modelling. Our manipulator robot is modeled by two arms in order to represent the human hand; one end of the arm is connected to a first articulation \( \theta_1 \). The end of the second arm is linked to \( \theta_2 \) and its other end to the effectors as shown in Figure 7.

(i) The joints are considered ideal (no mechanical play, rigid)

(ii) The length of the arms is a bit larger than other dimensions

(iii) The damping in the arms is considered negligible

We begin with forward kinematics. The manipulator will be able to sense its own position in some way using internal sensors (position encoders located at joints 1 and 2) that can determine directly the joint angles \( \theta_1 \) and \( \theta_2 \). We need therefore to express the beginning and end positions in function of these joint angles. This engenders the forward kinematics, which is to determine the position and orientation of the end-effectors or tool in terms of the joint variables.

It is acknowledged to establish a fixed coordinate system, called the world or base frame to which all objects counting the manipulator are referenced. In this case, we settle the base coordinate frame \( \mathbf{R}_0 (x_0, y_0) \) at the base of the robot, as represented in Figure 7. The coordinates \( (x, y) \) of the tool are declared in this coordinate frame as

\[
x = x_2 = \alpha_1 \cos \theta_1 + \alpha_2 \cos (\theta_1 + \theta_2),
\]

\[
y = y_2 = \alpha_1 \sin \theta_1 + \alpha_2 \sin (\theta_1 + \theta_2),
\]

in which \( \alpha_1 \) and \( \alpha_2 \) are the lengths of the two arms, respectively.

4.2. Inverse Geometric Modelling. Given the joint angles \( \theta_1 \) and \( \theta_2 \), we can fix the end-effector coordinates \( x \) and \( y \). In order to control the robot to move to the desired position, we
need the inverse geometric model; we need the joint variables $\theta_1$ and $\theta_2$ in function of the $x$ and $y$ coordinates of the desired location. This is the problem of inverse kinematics. In other words, provided $x$ and $y$ in the forward kinematic equations (7) and (8), we wish to solve for the joint angles.

Figure 7: From human hand to the modelling of the two-arm robot.

Figure 8: Inverse geometric modelling.

Figure 9: Application of the classic PI controller.
Given Figure 8, we can see that the $\theta_2$ angle is given by
\[
\cos \theta_2 = \frac{x^2 + y^2 - \alpha_1^2 - \alpha_2^2}{2 \alpha_1 \alpha_2} = D. \tag{9}
\]

We can now determine $\theta_2$ as follows:
\[
\theta_2 = \cos^{-1} (D). \tag{10}
\]

But, a better way to find $\theta_2$ is to notice that if $\cos \theta_2$ is given, then $\sin \theta_2$ can be given as
\[
\sin \theta_2 = \pm \sqrt{1 - D^2}. \tag{11}
\]

$\theta_2$ can be found by
\[
\theta_2 = \tan^{-1} \frac{\sqrt{1 - D^2}}{D}. \tag{12}
\]

The advantage of this approach is that both elbow-up and elbow-down solutions are recovered by choosing the positive and negative signs in equation (13), respectively. Then, it is easy to show that $\theta_1$ is now given as
\[
\theta_1 = \tan^{-1} \left( \frac{y}{x} \right) - \tan^{-1} \left( \frac{\alpha_2 \sin \theta_2}{\alpha_1 + \alpha_2 \cos \theta_2} \right). \tag{13}
\]

We have to notice that the angle $\theta_1$ depends on $\theta_2$. This makes sense physically since we would expect to require a different value for $\theta_1$, depending on which solution is chosen.

Figure 10: Simulation results for the P controller applied to (a) Arabic letter 'SIN' and (b) triangle.

Figure 11: Simulation results for the PI controller applied to (a) Arabic letter “SIN” and (b) triangle.
for $\theta_2$. Provided the geometric modelling, we can design an adaptive controller to cope with every changing in the system’s conditions.

5. Assistive Robot Arm Control

5.1. Classic Handwriting Control Design. In this paper, we present a specific controller for each desired form model:
As a first approach, an application of the classic PI controller, Figure 9, applied to the geometric model presented in the previous paragraph.

As a second approach, an adaptive PI controller using a new tuning technique to choose the PI gains in order to cope with the change of characters or writers.
PI (proportional integral) control has a simple control structure which was understood by plant operators and which they found relatively easy to tune. Since many control systems using the PI control have proved satisfactory results, it still has a wide range of applications in industrial control.

The control parameters are as follows:

(i) The proportional term: providing an overall control action proportional to the error signal through the constant gain factor

(ii) The integral term: the action is to reduce steady-state errors through low-frequency compensation by an integrator

In the literature, several works have been describing the PI structure [14, 29]. The structure is also known as the parallel form and is represented by

\[ u(t) = K_p e(t) + K_i \int_0^t e(t)dt, \]  
\[ e(t) = y_d(t) - y_e(t), \]

with \( K_p \): proportional gain, \( K_i \): integral gain, \( y_d \): desired position, and \( y_e \): estimated position.

Different equations used in this work for the PI controller applied to the handwriting system process are as follows:

\[ e_x(t) = x_d(t) - x_e(t), \]  
\[ e_y(t) = y_d(t) - y_e(t), \]

where \( e_x \) and \( e_y \) are the errors relative to \( x \) and \( y \).

**Figure 12:** Principle of the adaptive control approach. *Direct geometric model. **Inverse geometric model.*
5.1.1. P Controller. We have applied the classic PI controller with a proportional gain \( K_p \), a high value of \( K_i \) gain in order to get a P controller, and the command becomes

\[
\begin{align*}
    u_{xp}(t) &= K_p \times e_x(t), \\
    u_{yp}(t) &= K_p \times e_y(t),
\end{align*}
\]

where we multiply the error \( e(t) \) by a gain \( K_i \) specifically chosen so that the simulation results obtained will be the nearest to the database shape. Figure 10 shows simulation results for an Arabic letter ‘SIN’ and a ‘triangle.’

The obtained shapes show unsatisfactory concordance between the controller result and the traces generated by the writer. The P controller cannot give good results when it is used to mimic a handwritten cursive shape.

5.1.2. PI Controller. We apply the PI controller to the same shapes provided the same \( K_p \) and considerable \( K_i \):

\[
\begin{align*}
    u_{xpi}(t) &= K_p \times e_x(t) + K_i \times \int_0^t e_x(t), \\
    u_{ypi}(t) &= K_p \times e_y(t) + K_i \times \int_0^t e_y(t).
\end{align*}
\]

Figure 11 shows satisfactory results. A good correspondence is revealed in the following of the two geometric traces.

The PI controller can be used in the generation of followed traces to mimic handwritten shapes.

We are opting to introducing of a PI controller to control our robot and to generate geometric shapes similar to those produced by the writers.

PI equations are as follows:

\[
\begin{align*}
    u_{xpi}(t) &= K_p \times e_x(t) + K_i \times \int_0^t e_x(t), \\
    u_{ypi}(t) &= K_p \times e_y(t) + K_i \times \int_0^t e_y(t).
\end{align*}
\]

5.1.3. Drawbacks of the Classic Controller Design. The classic PI controller proposed has generated graphic handwritten traces. In fact, each form has its own \( K_p \) and \( K_i \) gains; in other words, two forms of the same type and written by the same writer have two different gains. It is important to note that the values of the integral \( K_i \) gain values and the \( K_p \) gain values of several graphic traces are very close.

To validate the classical PI controller results and to test its limits, we propose in this part the following validation strategy:

(i) We maintain the same proportional gain \( K_p \) and we set just the integral gain \( K_i \).

(ii) Writer-1 writes the Arabic letter “HA” three times. In this case, we define three models, model-11, model-12, and model-13, relating to each example of the “HA” letter. The PI-1 controller, developed for model-11, is applied to model-12 and model-13, Table 2.

(iii) Writer-2 writes the Arabic letter “AYN” three times. In this case, we define three models, model-21, model-22, and model-23, relating to each instance of the letter “AYN.” The PI-2 controller, developed for model-21, is applied to model-22 and model-23, Table 3.

The validation results of the classic PI controller are shown in Tables 4 and 5. The red line shows the experimental data, and the blue line represents the response of the proposed command structure.

Validating the classic PI controller gave unsatisfactory results when the same \( K_i \) gain value is considered for various examples written by the same person. This shows the limitations of the classic PI controller, that is, only capable of generating the model with which it was developed.

The same validation strategy has been applied to other types of graphic forms, namely, a circle and a triangle. The results of this validation are shown in Tables 6 and 7 which present unsatisfactory results when using the same \( K_i \) gain for various examples, even if they are written by the same writer.

In this sense, we define the following:

(i) Writer-3 wrote the geometric shape “Circle” three times. In this case, we define three models, model-31, model-32, and model-33, relating to each instance of the “Circle,” Table 8. The PI-3 controller, developed for model-31, is applied to model-32 and model-33, Table 6.

(ii) Writer-4 wrote the geometric shape “Triangle” three times. In this case, we define three models, model-41, model-42, and model-43, relating to each example of the “Triangle,” Table 9. PI-4 controller, developed for model-41 is applied to model-42 and model-43, Table 7.

When the same form is produced and when written by the same person, the proposed PI controller gives...
unsatisfactory results. Indeed, even when the controller has to recognize the type of graphic trace, the error between its response, and experimental recordings are considerable. This leads us to think about the design of an adaptive PI controller (API) to cope with the changing conditions and data that can have the system studied.
6. Adaptive Handwriting Control Design

In this part, an adaptive PI controller is proposed to overcome the limitations of the conventional PI controller described above. For this, we opted for the use of IEMG signals that go into the calculation of the appropriate gains to generate the desired writing.

Figure 12 illustrates the basic principle of the adaptive correction approach proposed. In this case, the integrated electromyographic signals are injected at this corrector that consists of the following:

(i) Block 1 is to calculate the average of the IEMG1 and IEMG2 signals
(ii) Block 2 allows, from the average of the electromyographic signals related to the manuscript shapes, to select optimal PI gains
(iii) Block 3 contains a base of PI controllers, and the decision function selects the optimal PI from this library

6.1. Decision Function. For the decision function, we have opted to the use of the IEMG signals. These signals are a condition to choose the appropriate gains to force the system to follow a desired performance even with the existence of some changes in system characteristics.

Using the average of the Integrated ElectroMyoGraphic signals (IEMG1 and IEMG2) provided for each geometric shape from the database of the experimental approach results, we have chosen to attribute to each shape a Ki gain so that we can interpolate them by a "spline" function. Provided the average of the IEMG of one shape and having the previous function, our system can estimate the Ki gain for the related shape. So, our model becomes general and applicable to different letters and geometric shapes, Figures 12 and 13.

The results in Figure 14 show different results of the control technique used to obtain an adaptive design considered as unpredictable; each one is associated to a specific handwriting model.

In order to compute a specific gain for each written shape, we have worked on a tuning technique for choosing a specific integral Ki gain of an adaptive PI controller. The proposed controller is inspired by gain schedule that is used for the control of nonlinear systems and based on interpolating between different linear controllers associated with several operating points of the studied nonlinear system.

The scheduling of the proposed adaptive gain is based on averages of different electromyography signals, considered as unpredictable; each one is associated to a specific handwriting model.

Regarding the simulation results, we remark that the adaptive PI (API) controller is giving satisfactory results when it comes to cursive letters and forms. That is to say, cross validation shows a negligible error between real data and the estimated shape even when we consider data written by different writers. In summary, compared to the classical controller, cursive Arabic letters and geometric forms are better followed by the adaptive PI, even in the most complicated areas, characterized by an abrupt change in orientation, direction, and even the velocity.

However, the reliability of this approach depends on the number of PI correctors of the developed base. In this sense is a part of our future work, which consists in overcoming this problem and improving the interpolation function by using nonconventional approaches such as artificial neural networks.

Data Availability

The EMG signals and the corresponding letter coordinates data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References


