Human-like Robotic Handwriting and Drawing

A Thesis

Presented in Partial Fulfillment of the Requirements for the Degree Master of Science in the Graduate School of The Ohio State University

By

Boren Li, Bachelor of Science

Graduate Program in Electrical and Computer Engineering

The Ohio State University

2012

Master's Examination Committee:

Prof. Hooshang Hemami, Advisor

Prof. Yuan F. Zheng, Co-Advisor

Abstract

The method of human-like handwriting and drawing is addressed with a three-link arm. Three strategies of trajectory planning are considered: the basic stroke method, the Bezier Curve method and the non-gradient numerical optimization method. Scalar patterns are converted into vector form whose movement sequence and speed can be selected to imitate human handwriting and drawing. A nonlinear three-link three dimensional arm, similar to human arm, tracks the planned trajectories. The feasibility of these methods is demonstrated by simulation.

This work is dedicated to the ones I loved and loved me.

Acknowledgments

I would like to express my sincere thanks to those who helped me throughout this project. First and foremost, I would like to give my best gratitude to my advisor, Professor Hooshang Hemami. Without his sage advice and heartfelt encouragement, I would not have completed this project. I also send my sincere thanks to Professor Zheng, who gave me many valuable advices for the advancement of my project. The author is also grateful to Da Che and other graduate students in the ECE department who offered many help to the simulation efforts.

Vita

Sep. 2002 to Jul. 2005.....The Experimental High School Attached to Beijing Normal University, China

Sep. 2005 to Jul. 2009.....B.S. in Automation, Harbin Institute of Technology, China

Sep. 2010 to present .....M.S. Department of Electrical and Computer Engineering, The Ohio State University, USA

Fields of Study

Major Field: Electrical and Computer Engineering

Table of Contents

List of Tables

List of Figures

Chapter 1: INTRODUCTION

Writing and drawing [1, 2, 3], easily performed by humans, are among activities that make robots more human-like.

An algorithm based on character-segmentation is proposed by S. Yussof [1] to enable robots to write printed letters. The character information is divided and stored into segments in order to break a complicated problem into a sequence of simpler ones. G. Wang and others [4, 5] extract outline font information, directly or indirectly, from TureType Font (TTF) database under the Windows platform or Mac OS. TrueType is an outline font standard originally developed by Apple Computer in the late 1980s as a competitor to Adobe's Type 1 fonts used in PostScript, currently the most common format for fonts [6, 7, 8]. This method, referred to as TTF method, has the advantage of using previous font databases conveniently to plan reference trajectories, but not sufficiently flexible to generate an arbitrary curve, as needed in handwritings or drawings [9, 10].

Two methodologies for implementation of handwriting for robots are:

1) Method used by humans [11];
2) Imitation by a master-slave system.

From a human's perspective, it is evident that writing style is the same produced by arm, thumb or fingers [12]. There is constancy of the movement despite major changes in

motor implementation. This phenomenon, referred to as motor equivalence, is of theoretical importance because it suggests that actions are encoded in the central nervous system (CNS) in terms that are more abstract than commands to specific muscles [13, 14]. Handwriting may be represented in terms of ‘strokes’ that are encoded in terms of relative position and spatial direction, but without any specific motoric reference. Details of motor implementation, such as stroke size or speed, may be left unspecified until the effector is known. Once the effector is known, allowance can then be made for effector-specific complexities, such as gravity or joint segmental interaction torques, which distort trajectories in a way that depends non-linearly on movement speed [12]. Implementation of the human methods for the current generation of robots is a challenge but not realistic at this time.

A thorough review of the literature on human handwriting is beyond the scope of this paper. Prerequisites of a healthy subject and the motivational factors and/or interests are ignored. The next issues are issues of perception, recognition of characters [15, 16], learning and spatial attention [17]. We will not consider these issues here. Also, this paper is to give possible code formats for handwriting and drawing rather than how to generate these codes automatically. The comparison of the complexity issues will not be discussed, either.

A master-slave system could offer quick implementations for limited tasks [18], but will not have the potential to answer or implement human handwriting capabilities. Of course, part of the human learning of handwriting may be through observation of others.

The paper is organized as follows. In Section 2, the writing and drawing problem is posed as a control problem. Three trajectory planning methods are developed in Section 3. Section 4 contains the description of the control strategy. Computer simulations are presented in Section 5. Section 6 discusses the results and shows some possible extensions for the trajectory planning methods.

Chapter 2: ROBOTIC CONTROL FORMULATION

The robotic control formulation of handwriting and drawing is considered here. The process contains three major steps, referred to as encoding, decoding and motoric reference. In the encoding step, a static handwriting or drawing is given as visual input. Three trajectory planning methods are proposed to encode this handwriting or drawing into information with relative position and spatial direction. This information will be saved to memory and is effector-independent. When needed to reproduce these handwritings or drawings on a medium, the relative information is loaded and decoded using the mechanism of respective trajectory planning methods in the decoding step. In the motoric reference step, specific effectors are given. The decoded information should be located, rescaled and /or distorted on the medium plane. Writing speed is chosen based on effector performance. Finally, specific commanding signals are generated to control the effectors.

This method affords a modular and computer-adapted formulation [19] that is strictly functional and can be expanded systematically in the future to be closer to the human system. It would allow a gradual implementation of CNS structures as they become firmly known and it can accommodate incorporation of known central and peripheral nervous structures. In addition, computationally oriented models would serve as test beds for feasibility studies or physiologically and anatomically observed phenomena regarding

handwriting and drawing. It can also afford a complex structure that could test different hypotheses about the role of CNS in handwriting and drawing and in verifying the inter-relationships between different functional modules. Other perspectives and philosophies and alternative interpretation have also been espoused [20].

A second advantage of a modular and comprehensive computational model is that deficiencies, injuries and diseases [21] can be simulated. This may help diagnosis, development of relief and corrective mechanisms in the long run. A third advantage would be development of objective criteria for comparison and assessment purposes [22].

The next pertinent issue is how the issues of scale and speed [23, 24] are resolved and command signals are generated. A philosophical issue, at this point, is whether handwriting and drawing can be assumed to be analogous to locomotion, i.e., point to point movement with the additional dimension of pressure or force [25, 26, 27] applied to the medium. At the same time, new criteria of optimality such as information content, or readability, etc. become relevant [24, 28]. If the analogy with locomotion holds, the next questions are whether pattern generators are involved in handwriting and drawing, where in the CNS they may be located and the role of the basal ganglia and the cerebellum [29] in generation of central signals issued to the spinal cord [30,31]. This issue will be subject of future work.

Chapter 3: TRAJECTORY PLANNING METHODS

3.1 Overview

Three methods of trajectory planning (i.e. Basic stroke method, Bezier curve method and Non-gradient numerical optimization method) are considered. The objective is to transform scalar patterns with position information into “vector” form. The direction and speed are integrated in the vectors.

The idea of character-segmentation is used in all three methods. The stroke method is the basis for the other two algorithms; the Bezier Curve method is an expansion of the TTF method; the non-gradient numerical optimization method is a fresh new idea, inspired by the vehicle guidance problem.

All the derivations are presented based on a two-link planar robot for ease of communication. These methods are expanded to more complex musculoskeletal systems [25, 31, 32].

3.2 Basic Stroke Method

A stroke is a single unbroken movement or one of a series of repeated to-and-fro movements [33]. Alphanumeric characters consist of two main types of strokes, mainly a straight line and a curve [1]. For simplicity, circular arcs and line segments are enough to produce all different alphanumeric characters with key features [34].

In this section, two types of stroke (i.e. circular arcs and line segments) are discussed. The speed of writing is also considered.

3.2.1 Circular Arcs

Circular arcs are segments of the circumference of circles, the trajectory of the corresponding circles could be identified first and then, by specifying the time duration, the desired circular arcs could be constructed as reference inputs.

The trajectory of a circle is defined as: $(x-c)^2+(y-d)^2=r^2$, where the center of the circle is (c,d) and the radius is r.

The endpoint of the forearm is described by:

$$\begin{cases} x = l_1 \sin\theta_1 + l_2 \sin\theta_2 \\ y = l_1 \cos\theta_1 + l_2 \cos\theta_2 \end{cases}$$

Let $\omega(t)$ be the desired angular velocity and $x_0 = x(0), y_0 = y(0)$ is where the circular arc starts. Then, $\zeta = \int_0^t \omega(\tau)d\tau$ (in radians) which is the angular increment at each particular time, and $\begin{cases} x = c + r\cos(\theta_0 + \zeta) \\ y = d + r\sin(\theta_0 + \zeta) \end{cases}$.

Solving the above equations with the initial condition of $\theta_0 = \arctan\dfrac{y_0 - d}{x_0 - c}$ results in the following:

$$\begin{cases} \theta_1(t) = \arcsin\left(\dfrac{c + r\cos(\theta_0+\zeta)}{\sqrt{(l_1\sin\gamma(t))^2 + (l_1\cos\gamma(t) + l_2)^2}}\right) - \phi(t) + \gamma(t) \\ \theta_2(t) = \arcsin\left(\dfrac{c + r\cos(\theta_0+\zeta)}{\sqrt{(l_1\sin\gamma(t))^2 + (l_1\cos\gamma(t) + l_2)^2}}\right) - \phi(t) \end{cases} \quad (3.1)$$

Where $\zeta = \int_0^t \omega(\tau) d\tau$,

$$\gamma(t) = \arccos \frac{c^2 + d^2 + r^2 - l_1^2 - l_2^2 + 2r(c \cdot \cos(\theta_0 + \zeta) + d \cdot \sin(\theta_0 + \zeta))}{2l_1 l_2},$$

$$\phi(t) = \arctan \frac{l_1 \sin \gamma(t)}{l_1 \cos \gamma(t) + l_2}$$

At this point, the mapping between the position domain $(x(t), y(t))$ and the two-link arm angle domain $(\theta_1(t), \theta_2(t))$ is achieved. The positions $(x(t), y(t))$ generate a trajectory of circular arcs.

For the integrity of the trajectory planning problem for any specific model, constraints are deserved to be considered to insure the feasibility of the planned trajectory because of the existence of the physical limitations for the model.

As for the two-link planar arm model, several constraints should be added in the following way:

All points on the circular arc are reachable, i.e.

$$\begin{cases} \sqrt{c^2 + d^2} + r < l_1 + l_2 \\ x \geq 0 \end{cases} \tag{3.2}$$

3.2.2 Line Segments

The start and end point of the line segments are given, which are (x_0, y_0) and (x_f, y_f), respectively. The relationship between the endpoint position of the forearm (x, y) and the angle (θ_1, θ_2) has been known, which is:

$$\begin{cases} x = l_1 \sin \theta_1 + l_2 \sin \theta_2 \\ y = l_1 \cos \theta_1 + l_2 \cos \theta_2 \end{cases}$$

The speed function of writing $v(t)$ is what should be chosen, then:

$$\begin{cases} x(t) = x_0 + \int_0^t v(\tau)d\tau \cdot \cos\alpha \\ y(t) = y_0 + \int_0^t v(\tau)d\tau \cdot \sin\alpha \end{cases}$$

Where:

If $x_f \neq x_0$, $\alpha = \arctan\dfrac{y_f - y_0}{x_f - x_0}$;

If $x_f = x_0$ and $y_f > y_0$, $\alpha = \dfrac{\pi}{2}$;

If $x_f = x_0$ and $y_f < y_0$, $\alpha = -\dfrac{\pi}{2}$.

With the same technique as the circular arcs, the following result can be derived:

$$\begin{cases} \theta_1(t) = \arcsin\left(\dfrac{x_0 + \zeta\cos\alpha}{\sqrt{(l_1\sin\gamma(t))^2 + (l_1\cos\gamma(t) + l_2)^2}}\right) - \phi(t) + \gamma(t) \\ \theta_2(t) = \arcsin\left(\dfrac{x_0 + \zeta\cos\alpha}{\sqrt{(l_1\sin\gamma(t))^2 + (l_1\cos\gamma(t) + l_2)^2}}\right) - \phi(t) \end{cases} \tag{3.3}$$

Where $\zeta = \int_0^t v(\tau)d\tau$, $\gamma(t) = \arccos\dfrac{{x_0}^2 + {y_0}^2 + \zeta^2 - l_1^2 - l_2^2 + 2\zeta(x_0\cos\alpha + y_0\sin\alpha)}{2l_1l_2}$,

$$\phi(t) = \arctan\frac{l_1\sin\gamma(t)}{l_1\cos\gamma(t) + l_2}$$

The result of the line segment is important and would be used in both the Bezier curve method and the non-gradient numerical optimization method later.

For the intactness of the trajectory planning problem, constraints should also be specified:

All points on the line segment should be reachable, i.e.

$$\begin{cases} x^2 + y^2 \le (l_1 + l_2)^2 \\ x \ge 0 \end{cases} \tag{3.4}$$

3.2.3 Speed of Writing

Before discussing the speed of writing, an example would show how to write alphanumerical characters using basic stroke method.

A letter 'P' would be segmented into two strokes, a line segment starting from the bottom to the top on the left and then a semi-circle from the top down to the middle of the line segment on the right.

Let the start and end point of the line segment be (x_0, y_0) and (x_f, y_f), then the end point of the semi-circle would be $(\frac{1}{2}(x_0 + x_f), \frac{1}{2}(y_0 + y_f))$.

For the simplest case, the velocity of writing the line segment and semi-circle would be chosen as constant.

Suppose the two segments are written in T_1 and T_2 seconds, respectively. Then, the two velocities are:

$$\begin{cases} v_0 = \dfrac{\sqrt{(x_f - x_0)^2 + (y_f - y_0)^2}}{T_1} \\ \omega_0 = \dfrac{\frac{1}{4}\pi\sqrt{(x_f - x_0)^2 + (y_f - y_0)^2}}{T_2} \end{cases} \tag{3.5}$$

This means two constant speeds are used. But large control effort is needed. Alternatively, triangular, trapezoidal or quadratic function of time could be used for speed.

Let the quadratic speed function be: $v = -K(t - t_0)^2 + v_0$ where (t_0, v_0) is the peak and K is a parameter of choice.

Let S be the length of the stroke:

$$S = \int_0^{t_f} (-K(t - t_0)^2 + v_0)dt$$

Where t_f is the time duration of the stroke, chosen based on performance.

For a given S and t_f, the velocity is:

$$v(t) = -\frac{6S}{t_f^3}(t - \frac{t_f}{2})^2 + \frac{3S}{2t_f} \tag{3.6}$$

The basic stroke method has the advantage of easy tuning for the parameters. But an arbitrary curve cannot be always decomposed into line segments and circular arcs without distortion. In other words, line segments and circular arcs are not enough to represent an arbitrary curve precisely for all time.

3.3 Bezier Curve Method

The Bezier curve method is introduced below. First, the basic principles and methods are discussed. Then, the mapping of speed to positions is given. Finally, the size and locations are considered.

3.3.1 Basic Principles and Methods

A Bezier curve is a parametric curve frequently used in computer graphics and related fields [35]. In vector graphics, Bezier curves are used for modeling smooth curves that could be scaled indefinitely. Bezier curves are also used in the time domain [36, 37],

particularly in animation and interface design, e.g., a Bezier curve could be used to specify the velocity over time.

A series of cubic Bezier functions can approximate an arbitrary trajectory with respect to time [38], modulating the speed in accordance with a specified speed function (trapezoidal function) and ultimately, interpolating to get a uniformly sampled data stream. Consequently, the Bezier curve method expands the TTF method from a fixed database to an arbitrary curve.

The cubic Bezier function is shown in Figure 3.1 [8]:

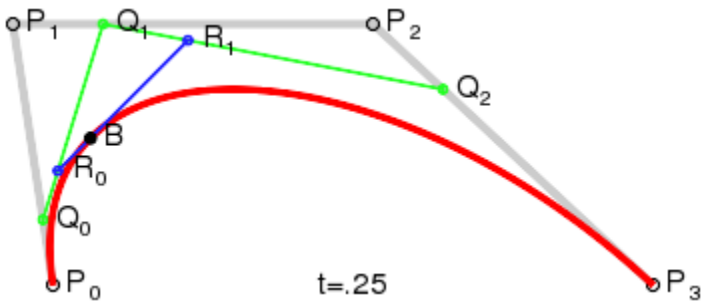

Figure 3.1: The cubic Bezier function

Four control points of P_0, P_1, P_2 and P_3 define the cubic Bezier curve. The curve starts at P_0, goes towards P_1 and arrives at P_3 coming from the direction of P_2. The curve does not pass through P_1 or P_2. These two points provide only the directional information. The distance between P_0 and P_1 determines "how long" the curve moves into direction P_2 before turning towards P_3. For some choices of P_1 and P_2, the curve may intersect itself, or contain a cusp.

The parametric form of the cubic Bezier curve is:

$$B(t) = (1-t)^3 P_0 + 3(1-t)^2 t P_1 + 3(1-t)t^2 P_2 + t^3 P_3, t \in [0,1]$$

The piecewise form for a series of cubic Bezier curves implemented in this algorithm is shown in the following way:

$$B(t)=\begin{cases}(1-t)^3P_0+3(1-t)^2tP_1+3(1-t)t^2P_2+t^3P_3, t\in[0,1]\\ (2-t)^3P_3+3(2-t)^2(t-1)P_4+3(2-t)(t-1)^2P_5+(t-1)^3P_6, t\in[1,2]\\ \vdots\\ (n+1-t)^3P_{3n}+3(n+1-t)^2(t-n)P_{3n+1}+3(n+1-t)(t-n)^2P_{3n+2}+(t-n)^3P_{3n+3}, t\in[n,n+1]\end{cases} \quad (3.7)$$

Each segment of the piecewise function (a sub-function) is defined as a sub-stroke.

The procedure of the Bezier curve method is given below:

Step 1: Encoding

1. *Visual input processing and coordinate system establishment;*
   - *1.1 Extraction of contour points of handwriting and cut the image to the focused area;*
   - *1.2 Memory coordinate establishment.*
2. *Parameter tuning of the piecewise Bezier function;*
   - *2.1 Choose start point of the handwriting;*
   - *2.2 Tune parameters of each sub-stroke to curve fit the skeleton of the handwriting.*

Step 2: Decoding

1. *Load the saved parameters;*
2. *Plug the parameters back to the piecewise Bezier function to reproduce the skeleton of the handwriting.*

Step 3: Motoric Reference

1. *Map between memory and medium coordinate;*
   - *1.1 Compute N values of points that are uniformly distributed with respect to time;*
   - *1.2 Locate, rescale and/or distort points to the medium plane.*
2. *Specify the speed function;*
   - *2.1 Build a time train for the N points;*
   - *2.2 Interpolate to get uniformly sampled data stream;*
3. *Generate specific commands to the effector.*

In the encoding stage, a visual input of handwriting is a grey-level image. This image contains only the static handwriting. The contour points from the grey-level images can

be extracted using Avrahami's algorithm [39]. This image will be then cut to a p by q matrix that contains only the focused area based on the contour points. The boundaries are determined by the outer outline of the handwriting. A memory coordinate can be established where the original is located on the left bottom corner of the focused image. The p by q matrix can be mapped on the coordinate to a 1 by q/p rectangular. After that, the parameters of the piecewise Bezier function should be tuned to curve fit the skeleton of the handwriting. This tuning is done manually here.

The procedure is given in Figure 3.2.

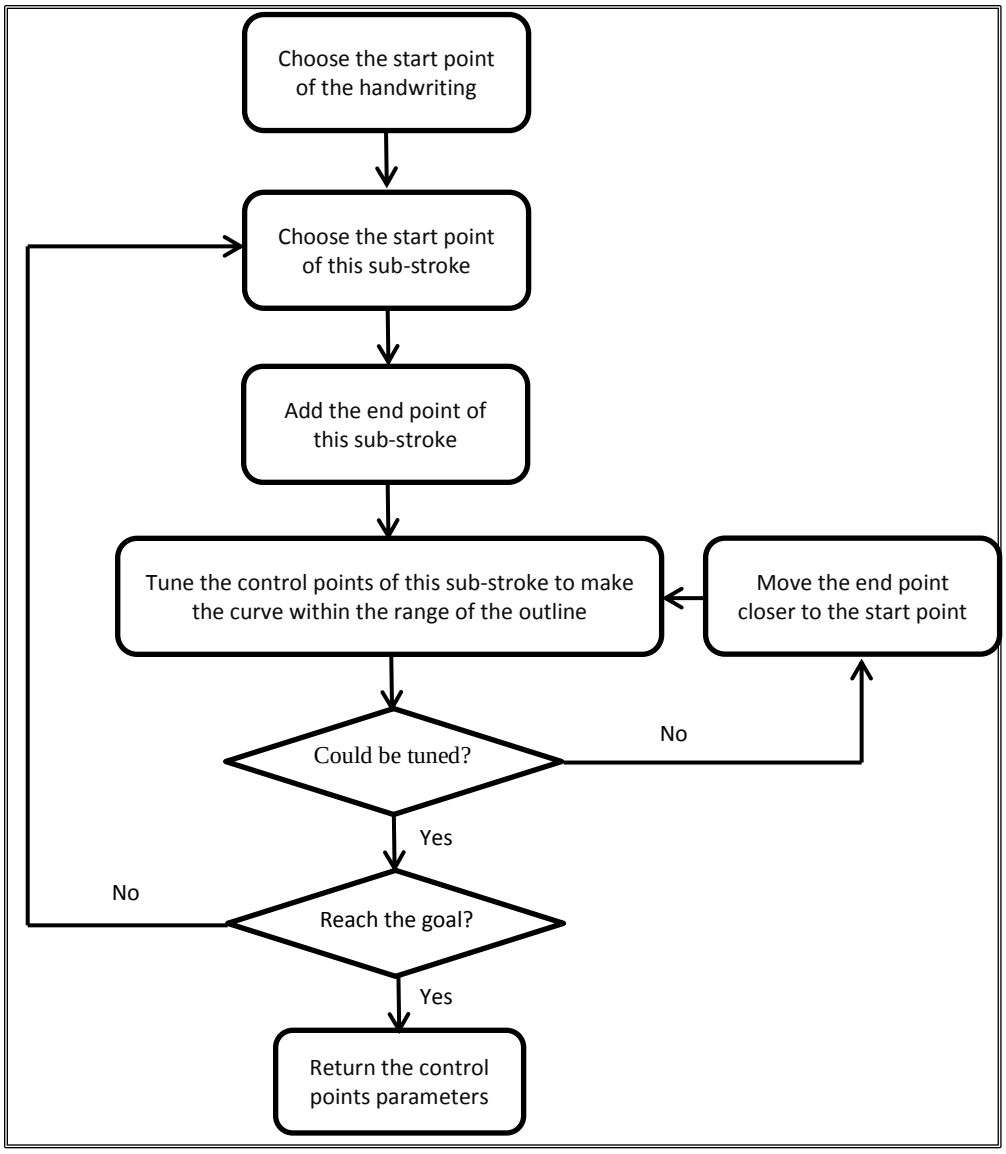

Figure 3.2: The procedure to tune the parameter of the piecewise Bezier curve

Note that:

1) The start point of the handwriting is chosen according to the habit of handwriting;
2) If it is the first sub-stroke, the start point of this sub-stroke coincides with the start point of the handwriting. Otherwise, the start point of this sub-stroke is the end point of the previous sub-stroke;

3) The end point of this sub-stroke is located at the nearest corner of the handwriting along the moving direction. The corner is where the curvature of the handwriting changes sharply. This end point can be chosen arbitrarily, though.
4) It is possible that the sub-stroke could not be tuned within the range of the outline. Then, the end point of this sub-stroke should be moved closer to the start point to see whether the criterion could be met;
5) The number of the sub-strokes should be continuously added until the end point of this sub-stroke coincides with the goal of the handwriting.

At this point, the static handwriting has been encoded and saved in memory in the format of Bezier curve parameters.

When the robot is commanded to reproduce the handwriting, the Bezier curve parameters should be loaded and simply plugged back to the piecewise Bezier function to reproduce the trajectory in the memory coordinate. This is referred as the decoding stage.

In the motoric reference stage, a mapping between memory and medium coordinate should be built, first. This is on the point-to-point mapping basis (See Section 3.3.2.). Then, the speed of writing should be considered (See Section 3.3.3). Finally, specific commands are generated to the effector. It turns to a tracking problem for control systems (See Chapter 4).

3.3.2 Mapping between Memory Coordinate to Medium Coordinate

Suppose the size of the medium is a P by Q (in meters) rectangular. The left bottom corner is defined as the original of the medium coordinate. The two axes coincide with the two sides of the rectangular.

The steps below should be followed to do the mapping:

1) Compute N values of points that are uniformly distributed with respect to time. N depends on how accurate the handwriting will be approximated by point to point movement;
2) Locate, rescale and distort points on the medium plane;

Suppose the value of the k[th] point is $(x(k), y(k))$ and $(\tilde{x}(k), \tilde{y}(k))$ on the memory and medium coordinate, respectively.

2.1) Rescaling and/or distorting

Suppose the desired size of the handwriting is within an m by n (in meters) rectangular boundaries. The value of the k[th] point in the memory coordinate is transformed using the following equation:

$$(\bar{x}(k), \bar{y}(k)) = (mx(k), \frac{np}{q} y(k)) \tag{3.8}$$

The handwriting will be distorted if $\frac{m}{n} = c\frac{p}{q}$ where c is a constant does not hold.

It is also possible to render the handwriting oblique (rotated by α degrees) using the following equation:

$$(\bar{x}(k), \bar{y}(k)) = (mx(k) + \tan\alpha \cdot \frac{np}{q} y(k), \frac{np}{q} y(k)) \tag{3.9}$$

2.2) Mapping the rescaled positions to the medium coordinate.

Suppose the original of the rescaled memory coordinate will be placed on (a,b) in the medium coordinate, where $a + \max(\bar{x}) \le P, b + \max(\bar{y}) \le Q$.

The transformed value of $(\tilde{x}(k), \tilde{y}(k))$ is derived by the following equation:

$$(\tilde{x}(k), \tilde{y}(k)) = (a + \bar{x}(k), b + \bar{y}(k)) \tag{3.10}$$

3.3.3 Speed of Writing

The speed of writing is achieved by a trapezoidal function. The computational complexity is lower than other functions like the quadratic function.

The trapezoidal speed function is shown in Figure 3.3.

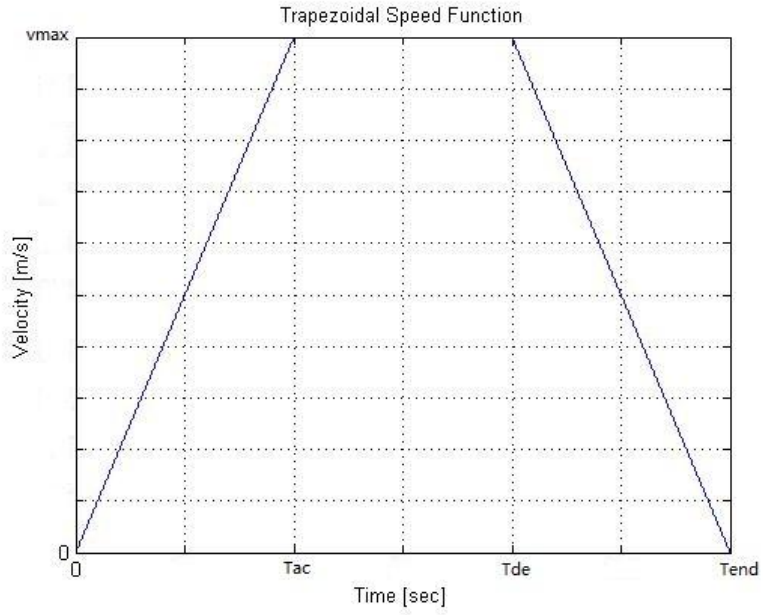

Figure 3.3: Trapezoidal speed function

The parametric form of the speed function is:

$$v(t) = \begin{cases} \dfrac{v_{\max}}{T_{ac}} t, 0 \le t < T_{ac} \\ v_{\max}, T_{ac} \le t < T_{de} \\ -\dfrac{v_{\max}}{T_{end} - T_{de}} (t - T_{de}) + v_{\max}, T_{de} \le t \le T_{end} \end{cases} \tag{3.11}$$

Where T_{de}, T_{ac} and T_{end} are chosen according to limitation of the control system.

The distance between every pair of successive points can be calculated using the following equation:

$$\lambda(k) = \sqrt{(\tilde{x}(k+1) - \tilde{x}(k))^2 + (\tilde{y}(k+1) - \tilde{y}(k))^2}, k = 1, 2...N-1 \tag{3.12}$$

Then, $v_{\max}$ can be calculated:

$$v_{\max} = \frac{2\sum_{k=1}^{N-1} \lambda(k)}{T_{de} - T_{ac} + T_{end}} \tag{3.13}$$

The problem of adding speed can be rewritten as: at which instant the movement would reach which position. Thus, a time train T(k) must be built corresponding to the positions.

For the trapezoidal velocity profile, the following recursive equation is derived:

1) If $0 \le T(k-1) \le T_{ac}$, then $T(k) = \sqrt{\frac{2\lambda(k)T_{ac}}{v_{\max}} + T(k-1)^2}$; (3.14)

2) If $T_{ac} < T(k-1) \le T_{de}$, then $T(k) = \frac{\lambda(k)}{v_{\max}} + T(k-1)$; (3.15)

3) If $T_{de} < T(k-1) \le T_{end}$, then $T(k) = T(k-1) + \frac{A - \sqrt{A^2 - 2B\lambda(k)}}{B}$ (3.16)

Where $A = v_{\max} - \frac{v_{\max}}{T_{end} - T_{de}}(T(k-1) - T_{de})$ and $B = \frac{v_{\max}}{T_{end} - T_{de}}$.

After the time train T(k) relative to position $(\tilde{x}(k), \tilde{y}(k))$ has been constructed, linear interpolation should be applied to get a uniformly sampled data stream.

3.4 Non-gradient Numerical Optimization Method

The Non-gradient numerical optimization method is a novel method to encode handwriting and drawings. This method is inspired by the problem of vehicle guidance [40].

The procedure of the Non-gradient numerical optimization method is given below:

Step 1: Encoding

1. *Visual input processing and coordinate system establishment;*
2. *Parameter tuning in the Non-gradient numerical optimization method;*
3. *Return and save the parameters.*

Step 2: Decoding

1. *Load the saved parameters;*
2. *Plug the parameters back to the Non-gradient numerical optimization method to reproduce the handwriting.*

Step 3: Motoric Reference

1. *Map between memory and medium coordinate;*
2. *Specify the speed function;*
3. *Generate specific commands to the effector.*

The general procedure is similar to that of the Bezier curve method. The difference is the algorithm to code the handwriting.

The core idea of this algorithm is to construct a series of maps and a moving strategy to achieve the desired trajectory. These maps are defined by cost functions $J_k(x, y)$. k refers to the k^{th} map, which corresponds to the k^{th} sub-stroke. The handwriting is comprised of sub-strokes. The start point of the 1^{st} sub-stroke is the start point of the handwriting, which is chosen according to the habit of handwriting. The goal point of the k^{th} sub-stroke is the start point of the $(k+1)^{th}$ sub-stroke unless it coincides with the end point of the handwriting. The goal point of each sub-stroke is located around the corner and arbitrarily chosen. For the k^{th} sub-stroke, the trajectory moves along the points that try to minimize $J_k(x, y)$ and reach the goal point $[x_{kgoal}, y_{kgoal}]$.

The cost function for the k^{th} sub-stroke $J_k(x, y)$ is defined as:

$$J_k(x,y) = w_{k1}J_{ko}(x,y) + w_{k2}J_{kg}(x,y) \tag{3.17}$$

Where w_{k1}, w_{k2} are the weights, J_{ko} is the cost function for the obstacles and J_{kg} is the cost function for the goals

A straight line can be easily bent into ellipse arc due to the tails of the Gaussian function. So it is used as the obstacle function, which is:

$$J_{ko} = Ae^{-(\frac{(x-x_{k0})^2}{2\sigma_{kx}^2}+\frac{(y-y_{k0})^2}{2\sigma_{ky}^2})} \tag{3.18}$$

Where $A, x_{k0}, y_{k0}, \sigma_{kx}$ and σ_{ky} are the parameters of the Gaussian function

Quadratic (bowl) function is used as the goal function, which is:

$$J_{kg} = ([x,y]^T - [x_{kgoal}, y_{kgoal}]^T)^T([x,y]^T - [x_{kgoal}, y_{kgoal}]^T) \tag{3.19}$$

The goal point is at the bottom of the bowl.

$w_{k1}, w_{k2}, A, x_{k0}, y_{k0}, \sigma_{kx}$ and σ_{ky} are referred to as local parameters.

Next, specific moving strategy is considered. Suppose the current position is the mth point on the kth sub-stroke, which is $[x_k[m], y_k[m]]$. There are Q planned angles that are uniformly distributed on $[-\pi, \pi]$ of the "sensing" circle. The circle is centered on the current position and the "sensing" radius is λ. The sensing position is given by the following equation:

$$\begin{bmatrix} x_k[m+1] \\ y_k[m+1] \end{bmatrix}_q = \begin{bmatrix} x_k[m] \\ y_k[m] \end{bmatrix} + \lambda \begin{bmatrix} \cos(\frac{2\pi q}{Q}) \\ \sin(\frac{2\pi q}{Q}) \end{bmatrix}, q = 1, 2...Q \tag{3.20}$$

It will compute the cost function values $J_k(\begin{bmatrix} x_k[m+1] \\ y_k[m+1] \end{bmatrix}_q)$ of the q points, find the minimum and move along that direction for a distance of λ to the next position $\begin{bmatrix} x_k[m+1] \\ y_k[m+1] \end{bmatrix}$. If $\begin{bmatrix} x_k[m+1] \\ y_k[m+1] \end{bmatrix}$ falls in the circle centered on $\begin{bmatrix} x_{kgoal} \\ y_{kgoal} \end{bmatrix}$ with a radius of ε $(\varepsilon > 0)$, then it terminates the kth sub-stroke and set the goal point be $\begin{bmatrix} x_k[m+1] \\ y_k[m+1] \end{bmatrix}$. The step size λ, the number of planned angles Q and the terminate condition ε are the global parameter of this algorithm and should be chosen properly.

At this point, the set-up of the algorithm has been given. In brief, the whole process approximates the "steepest descent approach" (a numerical optimization method). The cost function gives the idea of where the trajectory is heading for, which determines the general shape of the handwriting. The whole trajectory is decomposed into several sub-strokes. Each sub-stroke consists of multiple small steps. Each step is a line segment. These line segments are used to approximate the skeleton of the handwriting.

The methodology of tuning the parameters to curve fit the handwriting is given in Figure 3.4.

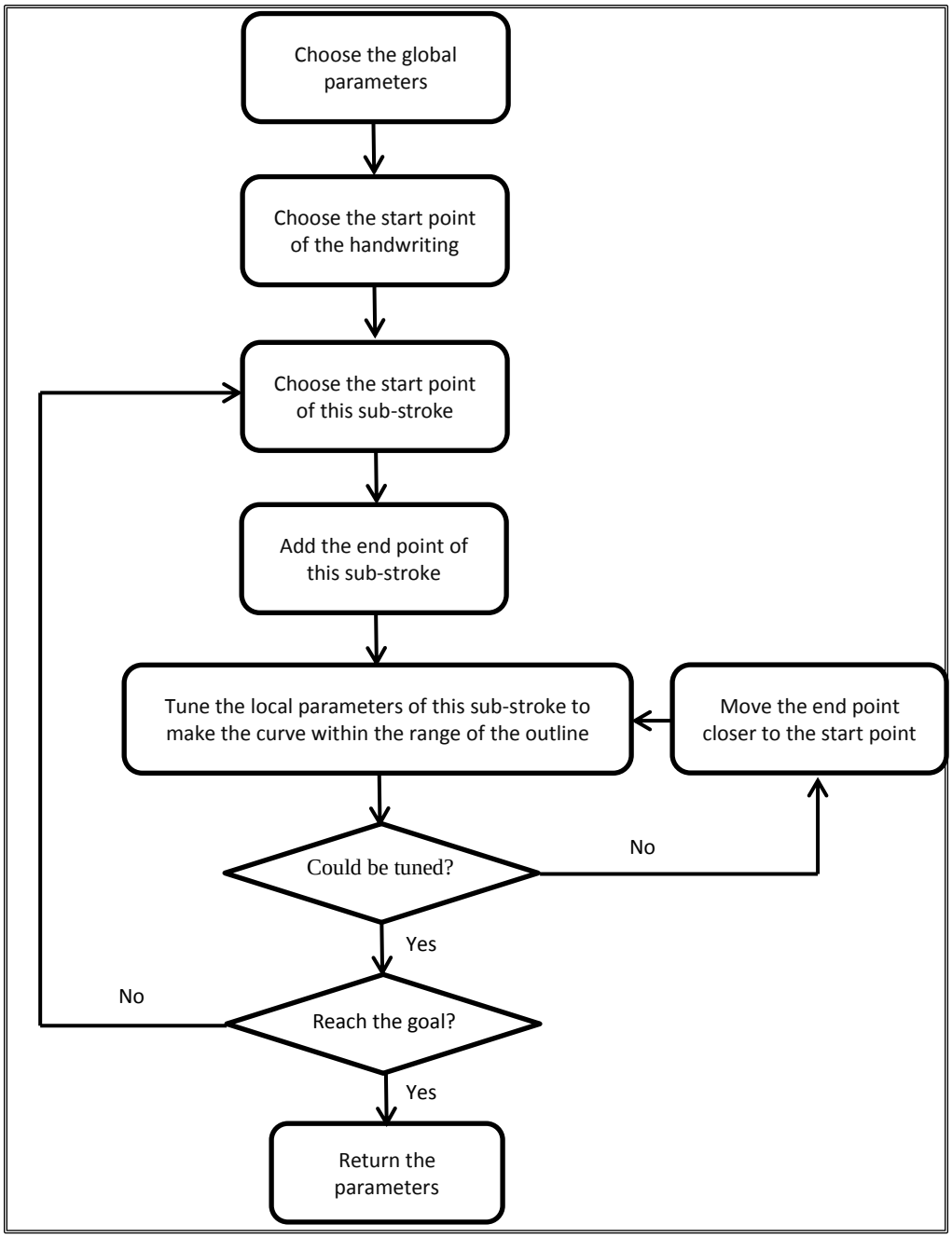

Figure 3.4: The procedure to tune the parameters of the Non-gradient numerical optimization method to curve fit the handwriting or drawing

The procedure is similar to that of the Bezier curve method. This method is not suitable to curve fit ellipse arcs with more than 180 degrees. Under these circumstances, it is necessary to divide the ellipse arcs into two sub-strokes or more.

Chapter 4: STABILITY AND CONTROL

A simple control strategy is proposed and applied to the three-link three-dimensional humanoid system (See Appendix A) that has abundant sensory and actuation channels as in living systems. The peripheral controller is a proportional and derivative (PD) controller. It can be envisioned as a high gain state feedback system. The high gain attribute imitates the agonist-antagonist co-activation in the natural system. The central controller produces the reference signals that are the desired angular positions of the upper arm, lower arm and hand with respect to time. These reference signals are generated using the three trajectory planning methods.

The block diagram of the system is shown in Figure 4.1.

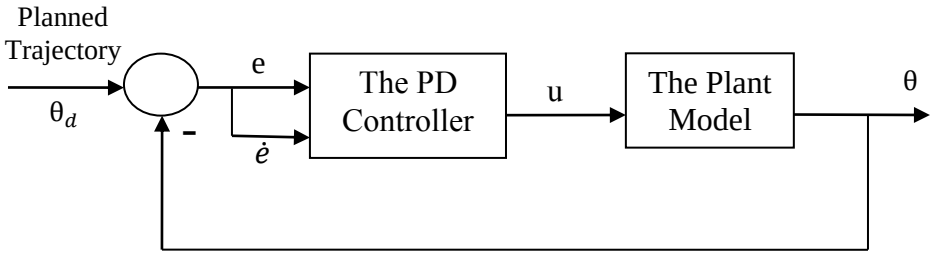

Figure 4.1: The block diagram for the system

Chapter 5: SIMULATION

In this section, a simulation for the three-link three-dimensional humanoid model is given to prove the feasibility and show the potential of the trajectory planning methods and the control strategy. This simulation is done using the Bezier curve method. Similar results could also be achieved by using the Non-gradient numerical optimization method. The simulation is to draw the logo of 'FC Barcelona' [41] and it contains the 'lift pen' process.

The results are shown in Figure 5.1 to Figure 5.9.

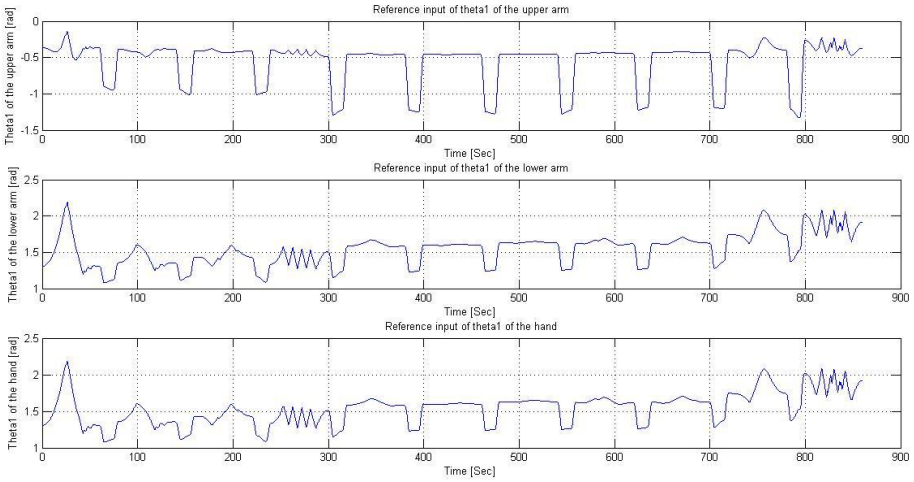

Figure 5.1: The input to the system from the central controller: the Euler roll angles in the inertial coordinate system (ics) for the upper arm, lower arm and hand as functions of time.

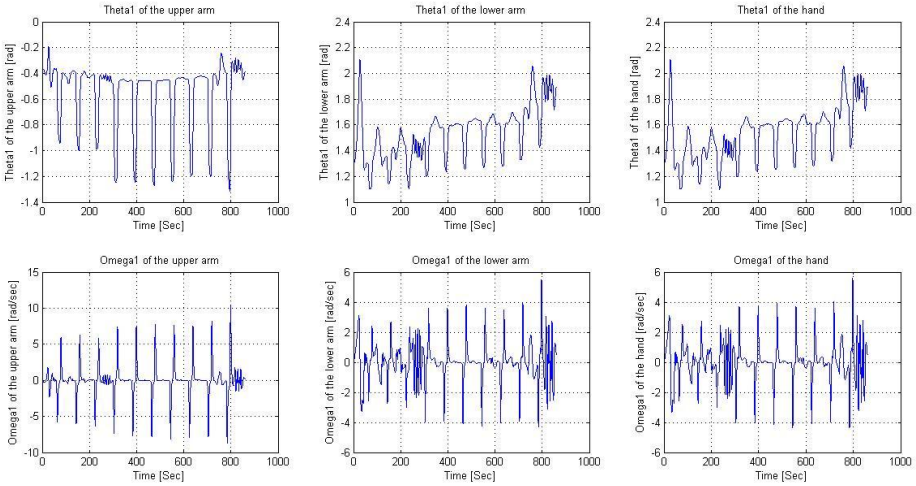

Figure 5.2: The trajectories of the response states of the system: the Euler roll angular positions in the ics and the roll angular velocities in the body coordinate system (bcs) of the upper arm, lower arm and hand as functions of time.

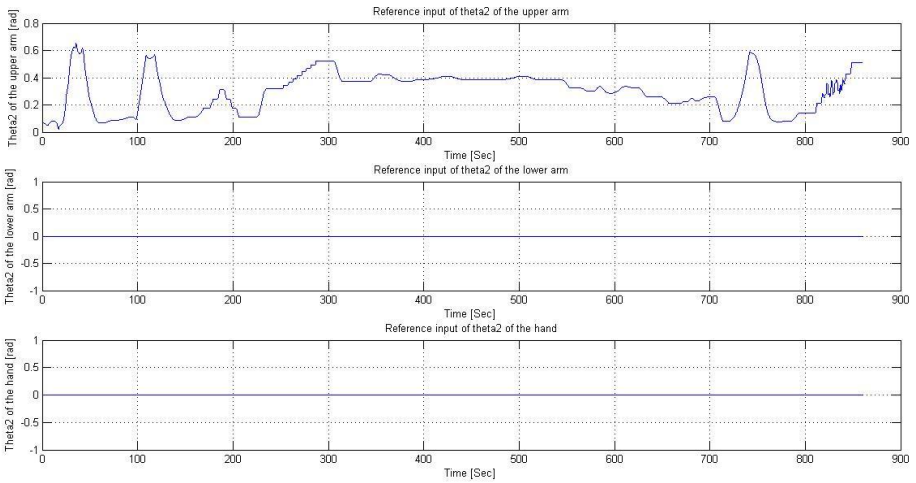

Figure 5.3: The input to the system from the central controller: the Euler pitch angles in the ics for the upper arm, lower arm and hand as functions of time.

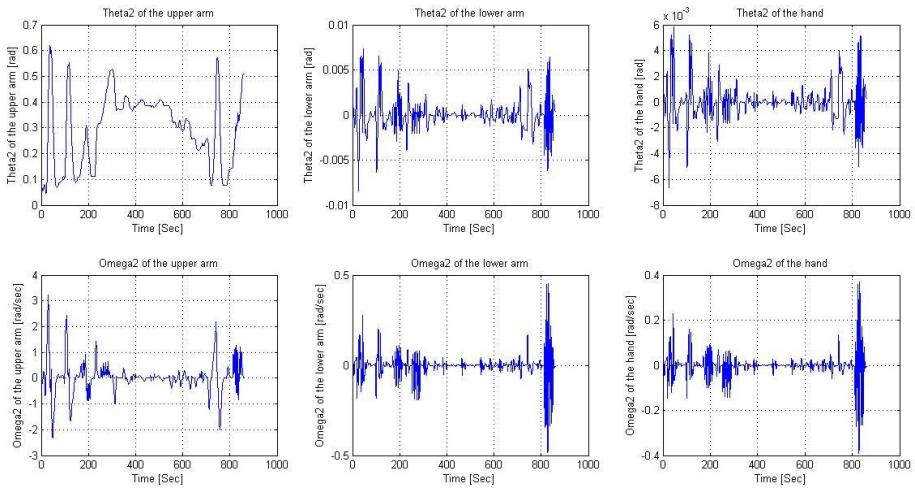

Figure 5.4: The trajectories of the response states of the system: the Euler pitch angular positions in the ics and the roll angular velocities in the bcs of the upper arm, lower arm and hand as functions of time.

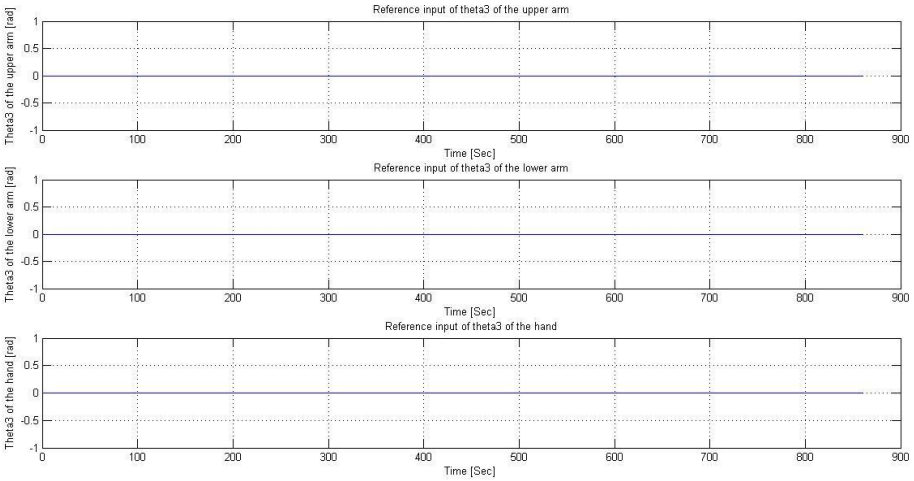

Figure 5.5: The input to the system from the central controller: the Euler yaw angles in the ics for the upper arm, lower arm and hand as functions of time.

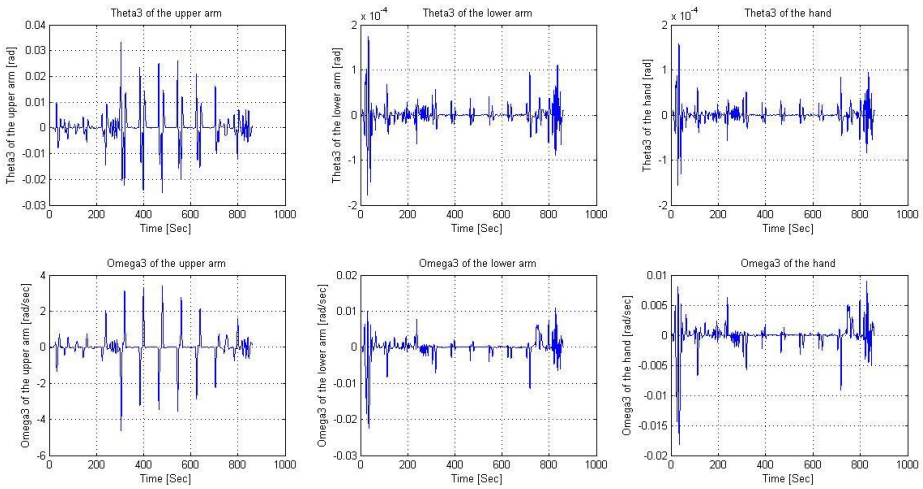

Figure 5.6: The trajectories of the response states of the system: the Euler yaw angular positions in the ics and the yaw angular velocities in the bcs of the upper arm, lower arm and hand as functions of time.

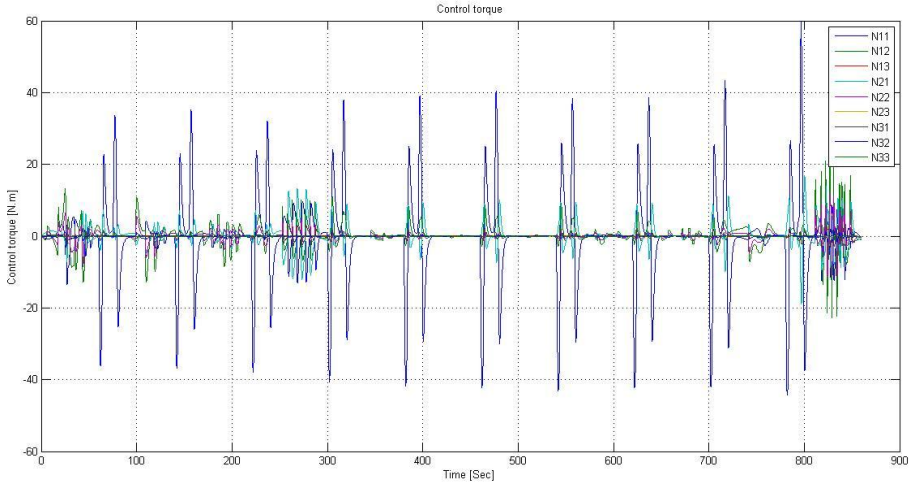

Figure 5.7: The control torques as functions of time.

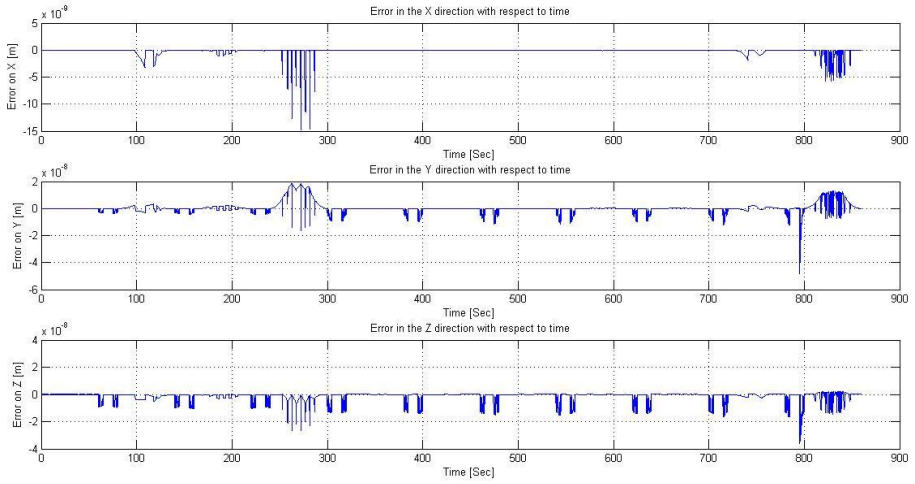

Figure 5.8: The positional error between the output and the reference with respect to time

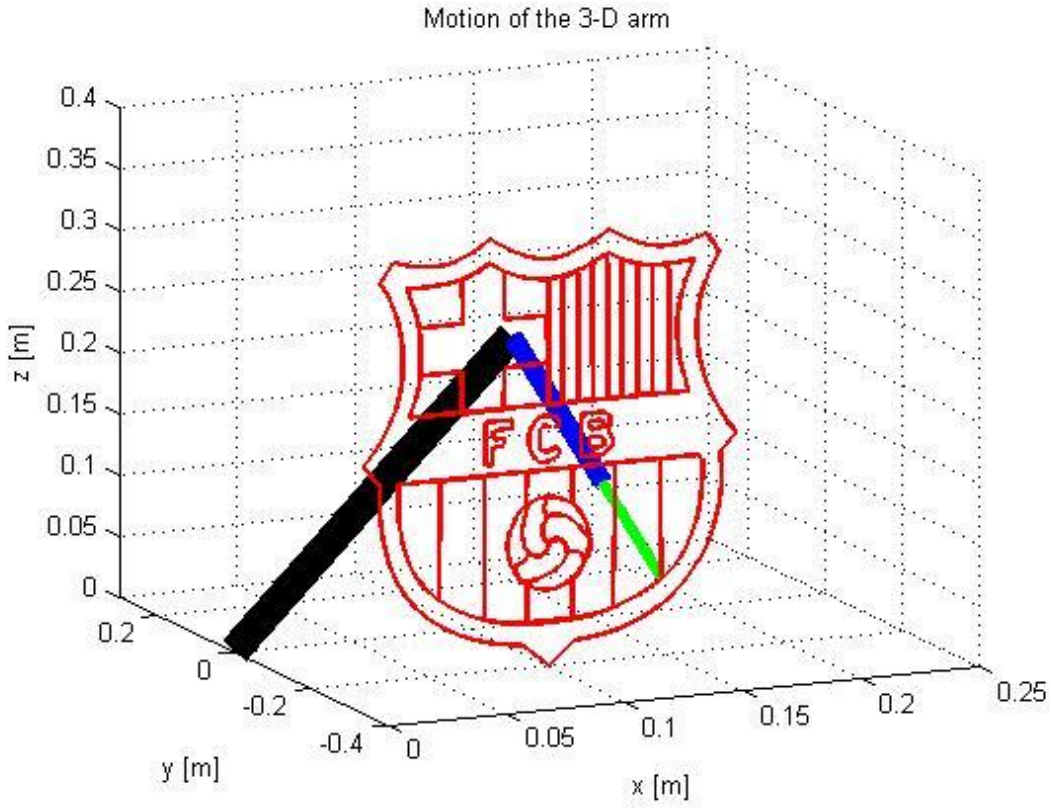

Figure 5.9: The final result in the position (x,y,z) domain

Chapter 6: CONCLUSION AND FUTURE WORK

Three trajectory planning strategies: the basic stroke method, the Bezier curve method and the Non-gradient numerical optimization method were described, designed and applied successfully to encode handwriting and drawings. A high-gain state feedback controller was proposed and implemented to control the three-link three-dimensional humanoid model.

Future works may concentrate on how to tune the parameters in the three trajectory planning methods automatically by computers to approximate a given handwriting or drawing. In addition, how to implement the trajectory planning methods on the three-dimensional three-link humanoid model with contact force at the end point is also worth to be considered.

Appendix A: THREE-LINK THREE-DIMENSIONAL HUMANOID ARM

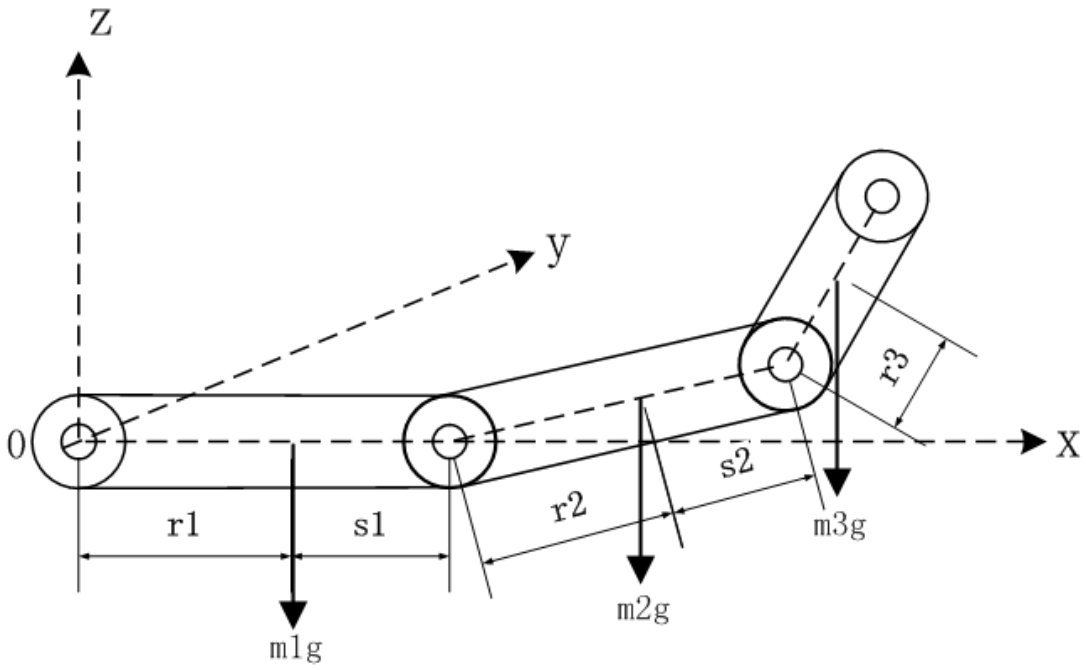

Figure A.1: The three-link three-dimensional humanoid arm

Let X, V, Θ and Ω be, respectively, the translational and rotational states of a free rigid body [42, 43]. Let Λ be 3-vector of force acting on a point of the body whose coordinates are vector r in body coordinate system (bcs). In connection with vector r, we define the skew symmetric 3 by 3 matrix $\breve{r}$ [44]. The vectors of force G_i and N_i are, respectively, the gravity vector and the vector of control inputs for the i^{th} rigid body. The equations of motion for the three-link three-dimensional humanoid model can be derived as follows.

Let the first rigid body with index 1 be connected to a platform at point a_1, and to the rigid body with index 2 at point b_1. Similarly, the points of contact of body 2 to body 1 and body 3 are, respectively, designated as a_2 and b_2. The point of contact of body 3 to body 2 is designated as a_3. Let the origin of the inertial coordinate system (ics) be at 0, and the point a_1, where the arm connected to the platform.

Let the coordinates of points a_1 and b_1 in the rigid body coordinate system be r_1 and s_1, respectively. Let the coordinates of points a_2 and b_2 be, respectively, r_2 and s_2 in the second body's coordinate system bcs2. Let the force Γ_1 act on body 1 at point a_1, and force $(-\Gamma_2)$ act on body 1 at point b_1. It follows that force Γ_2 acts on body 2 at point a_2.

The equations of motion for the three-link rigid body are:

The first body corresponds to the upper arm:

$$\begin{cases} \dot{X}_1 = V_1 \\ M_1\dot{V}_1 = G_1 + \Gamma_1 - \Gamma_2 \\ \dot{\Theta}_1 = B_1(\Theta_1)\Omega_1 \\ J_1\dot{\Omega}_1 = f_1(\Omega_1) + N_1 + \breve{r}_1 A_1^T(\Theta_1)\Gamma_1 - \breve{s}_1 A_1^T(\Theta_1)\Gamma_2 \end{cases} \tag{A.1}$$

In the above formulation

$$f_1 = -\breve{\Omega}_1 J_1 \Omega_1, \; M_1 = m_1 \times I$$

Where I is the 3×3 identity matrix. The 3×3 matrix J_1 is the positive definite symmetric moment of inertia matrix in the body coordinate system. The matrices $A_1(\Theta_1)$ and $B_1(\Theta_1)$ that appear above are given below [45]. Matrix A_1 is the transformation from

the bcs to ics and depends on the choice of Euler angles. A roll, pitch, yaw sequence is adopted here.

Let $A_{11}(\theta_{11})$, $A_{12}(\theta_{12})$ and $A_{13}(\theta_{13})$ be defined by

$$A_{11}(\theta_{11}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_{11} & -\sin\theta_{11} \\ 0 & \sin\theta_{11} & \cos\theta_{11} \end{bmatrix}$$

$$A_{12}(\theta_{12}) = \begin{bmatrix} \cos\theta_{12} & 0 & \sin\theta_{12} \\ 0 & 1 & 0 \\ -\sin\theta_{12} & 0 & \cos\theta_{12} \end{bmatrix}$$

$$A_{13}(\theta_{13}) = \begin{bmatrix} \cos\theta_{13} & -\sin\theta_{13} & 0 \\ \sin\theta_{13} & \cos\theta_{13} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Now $A_1(\Theta_1)$ can be defined:

$$A_1(\Theta_1) = A_{11}(\theta_{11})A_{12}(\theta_{12})A_{13}(\theta_{13})$$

The matrix $B_1(\Theta_1)$ is given by:

$$B_1(\Theta_1) = \begin{bmatrix} \dfrac{\cos\theta_{13}}{\cos\theta_{12}} & \dfrac{-\sin\theta_{13}}{\cos\theta_{12}} & 0 \\ \sin\theta_{13} & \cos\theta_{13} & 0 \\ \dfrac{-\sin\theta_{12}\cos\theta_{13}}{\cos\theta_{12}} & \dfrac{\sin\theta_{12}\sin\theta_{13}}{\cos\theta_{12}} & 1 \end{bmatrix}$$

By the same reasoning, the second body, the forearm, can be represented by the equations below:

$$\begin{cases} \dot{X}_2 = V_2 \\ M_2\dot{V}_2 = G_2 + \Gamma_2 - \Gamma_3 \\ \dot{\Theta}_2 = B_2(\Theta_2)\Omega_2 \\ J_2\dot{\Omega}_2 = f(\Omega_2) + N_2 + \breve{r}_2 A_2^T \Gamma_2 - \breve{s}_2 A_2^T \Gamma_3 \end{cases} \tag{A.2}$$

Finally, the third body represents the hand:

$$\begin{cases} \dot{X}_3 = V_3 \\ M_3\dot{V}_3 = G_3 + \Gamma_3 \\ \dot{\Theta}_3 = B_3(\Theta_3)\Omega_3 \\ J_3\dot{\Omega}_3 = f(\Omega_3) + N_3 + \breve{r}_3 A_3^T \Gamma_3 \end{cases} \tag{A.3}$$

The 9 connection constraints are given by the following equations:

$$\begin{cases} X_1 + A_1 r_1 = 0_{3\times 3} \\ X_1 + A_1 s_1 - A_2 r_2 = X_2 \\ X_2 + A_2 s_2 - A_3 r_3 = X_3 \end{cases}$$

$$\text{Or} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} I_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} \\ -I_{3\times 3} & I_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & -I_{3\times 3} & I_{3\times 3} \end{bmatrix}^{-1} \begin{bmatrix} -A_1 r_1 \\ -A_2 r_2 + A_1 s_1 \\ -A_3 r_3 + A_2 s_2 \end{bmatrix} \tag{A.4}$$

Take the 2nd derivative for equation (A.4), the following results can be achieved:

$$\dot{V} = H_{cc}(-H_{41}^T)\dot{\Omega} + H_{cc}H_{55} \tag{A.5}$$

$$\text{Where } H_{cc} = \begin{bmatrix} I_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} \\ -I_{3\times 3} & I_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & -I_{3\times 3} & I_{3\times 3} \end{bmatrix}^{-1}, H_{41} = \begin{bmatrix} \breve{r}_1 A_1^T & -\breve{s}_1 A_1^T & 0 \\ 0 & \breve{r}_2 A_2^T & -\breve{s}_2 A_2^T \\ 0 & 0 & \breve{r}_3 A_3^T \end{bmatrix},$$

$$H_{55} = \begin{bmatrix} -A_1 \breve{\Omega}_1^2 r_1 \\ -A_2 \breve{\Omega}_2^2 r_2 + A_1 \breve{\Omega}_1^2 s_1 \\ A_2 \breve{\Omega}_2^2 s_2 - A_3 \breve{\Omega}_3^2 r_3 \end{bmatrix},$$

$$\Omega = \begin{bmatrix} \Omega_{11} & \Omega_{12} & \Omega_{13} & \Omega_{21} & \Omega_{22} & \Omega_{23} & \Omega_{31} & \Omega_{32} & \Omega_{33} \end{bmatrix}^T$$

From equation (A.1), (A.2) and (A.3), the following results can be derived:

$$\begin{cases} M\dot{V} = -Mg + (H_{cc}^T)^{-1}\Gamma \\ J\dot{\Omega} = f + N + H_{41}\Gamma \end{cases} \tag{A.6}$$

Where $\Gamma = \begin{bmatrix}\Gamma_1 & \Gamma_2 & \Gamma_3\end{bmatrix}^T$, $M = diag(\begin{bmatrix} m_1 & m_1 & m_1 & m_2 & m_2 & m_2 & m_3 & m_3 & m_3 \end{bmatrix})$

$$g = grav \times \begin{bmatrix}0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1\end{bmatrix}^T,\ f = \begin{bmatrix} -\breve{\Omega}_1 J_1 \Omega_1 \\ -\breve{\Omega}_2 J_2 \Omega_2 \\ -\breve{\Omega}_3 J_3 \Omega_3 \end{bmatrix}, N = \begin{bmatrix} N_1 \\ N_2 \\ N_3 \end{bmatrix}$$

$$J = \begin{bmatrix} J_1 & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & J_2 & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & J_3 \end{bmatrix}$$

Eliminate Γ in equation (A.6) and combine it to equation (A.5), the following result can be achieved:

$$(J + H_{41}H_{cc}^T M H_{cc} H_{41}^T)\dot{\Omega} = f + N + H_{41}H_{cc}^T M H_{cc} H_{55} + H_{41}H_{cc}^T Mg \tag{A.7}$$

Finally, the dynamics of the model is given below:

$$\begin{cases} \dot{\Omega} = (J + H_{41}(\Theta)H_{cc}^T M H_{cc} H_{41}^T(\Theta))^{-1}(f(\Omega) + N + H_{41}(\Theta)H_{cc}^T M H_{cc} H_{55}(\Omega,\Theta) + H_{41}(\Theta)H_{cc}^T Mg) \\ \dot{\Theta} = B(\Theta)\Omega \end{cases} \tag{A.8}$$

Where $B(\Theta) = \begin{bmatrix} B_1(\Theta) & 0 & 0 \\ 0 & B_2(\Theta) & 0 \\ 0 & 0 & B_3(\Theta) \end{bmatrix}$

Appendix B: PHYSICAL PARAMETERS USED IN SIMULATION

The physical parameters of the three-link three-dimensional humanoid arm used in simulation are given in Table B.1.

Segment	**Mass**	**Dim. 1**	**Dim. 2**	**m. of i., j_1**	**m. of i., j_2**	**m. of i., j_3**
Symbol	m_i	r_i	s_i	j_1	j_2	j_3
Unit	Kg	m	m	Kg· m^2	Kg· m^2	Kg· m^2
Upper arm	2.4	-0.14	0.13	0.008	0.008	0.001
Lower arm	2.2	-0.13	0.12	0.007	0.007	0.001
hand	1.2	-0.07	0.09	0.007	0.007	0.002

Table B.1: Definition, Numerical values for the three-link arm

Bibliography

[1] S. Yussof, A. Anuar and K. Fernandez, "Algorithm for Robot Writing using Character Segmentation", Third International Conference on Information Technology and Applications (ICITA'05), vol.2, pp.21-24, 2005.

[2] A.M. Wing, F. Watts and V. Sharma, "Developmental Dynamics of Handwriting: Appraising the Relation between Handwriting and Personality", Development of Graphic Skills, Edited by J. Wann, A.M. Wing and N. Sovik, London: Academic Press, 1991.

[3] M.A. Eldridge, I. Nimmo-Smith, A.M. Wing and R.N. Totty, "The Variability of Selected Features in Cursive Handwriting: Categorical Measures", J Forensic Sci Soc, vol. 24, pp. 179-219, 1984.

[4] G. Wang, Z. Liao and X. Chen, "Robot Manipulator Lettering Technology and Motion", Journal of Chongqing University, vol.26 No.12, Dec. 2003.

[5] X. Ma, Q. Kong, W. Ma and X. Zhang, "4-DOF Lettering Robot's Trajectory Planning", Mechanical Engineering & Automation, vol.162 No.5, Oct. 2010.

[6] http://en.wikipedia.org/wiki/TrueType.

[7] Microsoft Typography: http://www.microsoft.com/typography/truetypefonts.mspx.

[8] TrueType Specifications: http://developer.apple.com/fonts/TTRefMan/index.html.

[9] J.S. McDonald, "Experimental Studies in Handwriting Signals", pp. 45-53, MIT RLE Tech. Rep.443, 1966.

[10] P. Marmelstein and M. Eden, “Experiments on Computer Recognition of Connected Handwritten Words”, Information and Control, vol. 7, pp. 255-270, 1964.

[11] D. Bullock, S. Grossberg and C. Mannes, “A Neural Network Model for Cursive Script Production”, Biological Cybernetics, Vol. 70, pp. 15-28, 1993.

[12] A.M. Wing, “Motor Control: Mechanisms of Motor Equivalence in Handwriting”, Current Biology, vol. 10, Issue 6, pp. 245-248, 2000.

[13] S.W. Keele, “Behavioral Analysis of Movement”, Handbook of Physiology, vol. 2, pp. 1391-1414, 1981.

[14] C.E. Wright, “Generalised Motor Programmes: Re-examining Claims of Effector Independence in Timing”, In Attention and Performance XIII Motor representation and control, pp. 294-320, 1990.

[15] N.V. Atteveldt, L. Blomert and J. Schwarzbach, “Different Neural Routes for Reading Handwriting and Print”, 9th Annual Meeting of the Cognitive Neuroscience Society, April 14-16, San Francisco, 2002.

[16] Y. Wada, “A Computational Theory for Movement Pattern Recognition Based on Optimal Movement Pattern Generation”, Biology Cybernetics, vol. 73, pp. 15-25, 1995.

[17] S. Grossberg, “Neural Dynamics of Motion Perception, Recognition, Learning and Spatial Attention”, MIT Press, Cambridge, 1995.

[18] S. Masui and T. Terano, “Calligraphic Robot with Fuzzy Logic”, on Proc. The Third IEEE Conference on Fuzzy System, vol.3, pp. 1598-1603, 1994.

[19] E. Bizzi, F.A. Mussa-Ivaldi and S. Giszter, “Computations Underlying the Execution of Movement: a Biological Perspective”, Science, vol.253, pp. 287-291, 1991.

[20] M. Rucci, D. Bullock and F. Santini, “Integrating Robotics and Neuroscience: Brains for Robots, Bodies for Brains”, Advanced Robotics, vol. 21, pp. 1115-1129, 2007.

[21] S.Z. Rapcsak, “Disorders of Writing”, In Apraxia: The Neuropsychology of Action, Psychology Press, 1997.

[22] V. Taler and N.A. Phillips, “Language Performance in Alzheimer’s Disease and Mild Cognitive Impairment: A Comparative Review”, pp. 501-556, 2008.

[23] M.M. Churchland, G. Santhanam and K.V. Shenoy, “Preparatory Activity in Premotor and Motor Cortex Reflects the Speed of the Upcoming Reach”, J Neurophysiol., vol.6, pp. 2842-2843, 2006.

[24] D. Bullock and S. Grossberg, “Neural Dynamics of Planned Arm Movements: Emergent Invariants and Speed-accuracy Properties during Trajectory Formation”, Psychol Rev, vol. 95, pp. 49-90, 1988.

[25] A.P. Georgopoulos, J.F. Kalaska, R. Caminiti and J.T. Massey, “On the Relationship between the Direction of Two-dimensional Arm Movements and Cell Discharge in Primate Motor Cortex”, J Neurosci, vol.2, pp. 1527-1537, 1982.

[26] A.P. Georgopoulos, J. Ashe, N. Smyrnis and M. Taira, “The Motor Cortex and the Coding of Force”, Science, vol. 256, pp. 1692-1695, 1992.

[27] D. Bullock, P. Cisek and S. Grossberg, “Cortical Network for Control of Voluntary Arm Movement under Variable Force Conditions”, Cerebral Cortex, vol.8, pp. 48-62, 1998.

[28] M.K. Babcock, J.J. Freyd, “Perception of Dynamic Information in Static Handwriting Forms”, Am. J. Psychol., vol. 101, pp. 111-130, 1988.

[29] J.L. Contreras-Vidal, S. Grossberg and D. Bullock, “A Neural Model of Cerebellar Learning for Arm Movement Control: Cortico-spino-cerebellar Dynamics”, Learning and Memory, vol. 3, pp. 475-502, 1997.

[30] P. Morasso, “Spatial Control of Arm Movement”, Exp Brain Res, vol.42, pp. 223-227, 1981.

[31] A. Shah, A.H. Fagg and A.G. Barto, “Cortical Involvement in the Recruitment of Wrist Muscles”, J Neurophysiol, vol. 6, pp. 2445-2456, 2004.

[32] J.M. Hollerbach, T. Flash, “Dynamic Interactions between Limb Segments during Planar Arm Movement”, Biology Cybernetics, vol. 44, pp. 67-77, 1982.

[33] Merriam Webster: http://www.merriam-webster.com/dictionary/stroke.

[34] http://en.wikipedia.org/wiki/Vector_graphics.

[35] Farin and Gerald, “Curves and Surfaces for Computer-aided Geometric Design (4 ed.), Ch.1, pp.2-5, Elsevier Science & Technology Books, 1997.

[36] J. Choi, R.E. Curry and G.H. Elkaim, “Path Planning Based on Bezier Curve for Autonomous Ground Vehicles”, IEEE Computer Society, Ch. 19, pp. 158-166, 2009.

[37] I. Skrjanc and G. Klancar, “Cooperative Collision Avoidance between Multiple Robots Based on Bezier Curves”, The Information Technology Interfaces (ITI), pp. 451-456, 2007.

[38] D. Salomon, “Curves and Surfaces for Computer Graphics”, Ch.1, pp.7-14, Springer, 2006.

[39] G. Avrahami and V. Pratt, “Sub-pixel Edge Detection in Character Digitization’, Raster Imaging and Digital Typography Ⅱ, eds. R. Morris and J. Andre, pp. 54-64. Cambridge University Press, 1991.

[40] K.M. Passino, “Biomimicry for Optimization, Control, and Automation”, Ch.5-6, pp. 155-178,231-238, Springer-Verlag London, Apr. 2004.

[41] http://www.fcbarcelona.com/.

[42] H. Hemami and B. Dariush, “Single rigid body representation, control and stability for robotic applications,” in Proc. IEEE 2000 International Conference on Robotics and Automation, Institute of Electronic and Electrical Engineers, Apr. 2000.

[43] H. Hemami and B. Dariush, “A mathematical representation of biorobots and humanoids for performance assessment, computer simulation and motion animation,” in Proc. 6th International Workshop on Advanced Motion Control, pp. 531,536, IEEE. Japan, 2000.

[44] H. Hemami and A. Katbab, “Constrained inverted pendulum for evaluating upright stability,” Journal of Dynamic Systems, Measurement and Control, pp. 343-349, 1982.

[45] H. Hemami, “Towards a Compact and Computer-Adapted Formulation of the Dynamics and Stability of Multi Rigid Body Systems”, Journal of Automatic Control, vol. 12, pp. 64-70, 2002.