UACJ INVESTIGACIÓN

Título del Proyecto de Investigación a que corresponde el Reporte Técnico:

Model-Based Control of a 6 DOF Robotic Arm with Visual Recurrent Bilayer ANN

Tipo de financiamiento

Sin financiamiento

Autores del reporte técnico:

Edgar Alonso Martínez García
Rafael Torres Córdoba
Ivan Carvajal Carlos (tesista maestría)

Model-Based Control of a 6 DOF Robotic Arm with Visual Recurrent Bilayer ANN

Resumen del reporte técnico en español **(**máximo 250 palabras)

Este trabajo presenta un modelo de control para un robot de 6 DOF que incluye generación y seguimiento de trayectorias para un robot manipulador de 6 grados de libertad. El sistema reconoce ataduras de ensamblaje de arnés mediante una cámara RGB que utiliza una red neuronal artificial Hopfield bicapa para procesar los datos de las placas de ensamblaje para el reconocimiento de las zonas de ensamblaje. Asimismo, se deduce el diseño cinemático de la estructura robótica para manipular y controlar el sistema. Este documento realiza un análisis matemático del modelo cinemático que describe la estructura, articulaciones, y eslabones lo que permite obtener las coordenadas xyz del efector final. Se deducen y prueban dos métodos de control. Se integra el método Tau-jerk con el método Newton-Rapson para obtener la cinemática inversa. Asimismo, se utiliza un método algebraico, adaptativo y variante en el tiempo cuyo control no requiere matrices cuadradas, así como la integración de un método de cinemática inversa.

Resumen del reporte técnico en inglés **(**máximo 250alabras):

This work presents a control model for a 6 DOF robot that includes trajectory generation and tracking. The system recognizes harness assembly ties using an RGB camera that uses a Hopfield bilayer artificial neural network to process data from the assembly plates for assembly zone recognition. Likewise, the kinematic design of the robotic structure to manipulate and control the system is deduced. This document performs a mathematical analysis of the kinematic model that describes the structure, joints, and links, which allows obtaining the xyz coordinates of the end effector. Two control methods are deduced and tested. The Tau-jerk method is integrated with the Newton-Rapson method to obtain the inverse kinematics. Likewise, an adaptive, time-varying algebraic method is used, the control of which does not require square matrices, as well as the integration of an inverse kinematics method.

Palabras clave: brazo-robot, linea-ensamble, ANN-recurrente, vision, control, tau-jerk

Usuarios potenciales **(**del proyecto de investigación**)**

Industria manufacturera de arneses

Reconocimientos

Al alumno de maestria en computo aplicado Ivan Carvajal Carlos por el desarrollo de este proyecto, y a la empresa Flextronics por el trabajo conjunto y planteamiento de problema y proveer la información requerida del proyecto. Así como también a los miembros del Laboratorio de Robótica que se involucraron parcialmente con algun tópico de este proyecto.

Productos generados

Capitulo de libro internacional

I. Carvjal Carlos, E.A. Martinez-Garcia, V.M. Carrillo Saucedo, R. Torres Cordoba, Chap: Bioinspired Robotic Arm Planning by Tau-Jerk Theory and Recurrent Multilayered ANN, Book:Deep Learning for Unmanned Systems, Springer LNCS, Germany, 2020

UACJ·IIT
LABORATORIO DE ROBÓTICA
http://robo-server.uacj.mx/en/
Cuerpo Académico de ΜεκατRοηικα
REPORTE TÉCNICO

RT-LR-13-20

Model-Based Control of a 6 DOF Robotic Arm with Visual Recurrent Bilayer ANN

Ivan Carvajal Carlos (M4)
Asesor: Edgar A. Martínez García, Rafael Torres Cordova
Laboratorio de Robótica
Instituto de Ingeniería y Tecnología
Universidad Autónoma de Ciudad Juárez

Resumen

This work presents an arranging model control for a 6 DOF robot that allows generation, monitoring, and control systems is developed for a robotic manipulator with six degrees of freedom that allows the manufacturing process to be carried out. The system must recognize a harness assembly tie utilizing an RGB camera that uses Hopfield's Artificial Neural Network to process the data from the assembly boards for the recognition of the assembly zones. Also, the kinematic design of the robotic structure is proposed to manipulate and control the system. This document performs a mathematical analysis of the kinematic model that describes the structure, joints, and links, which allows obtaining the x y and z coordinates of the end effector. Two control methods are deduced and tested. Tau-jerk method with Newton-Rapson method is integrated to obtain the inverse kinematics. Also, an algebraic method is used, which is adaptive and variant in time whose control does not require square matrices, as well as the integration of an inverse kinematics method.

1. Introduction

Robotic manipulators allow the integration of different features in workstations, this is known as flexible robotics manufacturing [1]. Manipulators can perform different tasks and even more if sensors are integrated that increase their capacities and even allow them autonomy, since many of these sensors currently operate "blindly", which is why vision systems have been integrated that allow feedback with the environment in which they operate.

Vision systems not only allow robots to recognize the environment, but it is also possible for them to recognize objects [9], which allows different industrial processes such as classifying objects, welding, or manipulating them different ways.

One of the challenges for a robot is the generation and tracking of trajectories, this not only implies manipulating the position of the robot. Furthermore, there may be obstacles that must be avoided. That is why various methods allow generating a trajectory between two points [3] who carried out a trajectory generation method based on geometric models, his method was used in Robocup, an international competition of robotics.

For the development of this project, a robotic manipulator is made to carry out the manufacturing process in a assembly plant. The process that requires generating and controlling the trajectory of the manipulator. Furthermore, it must be able to detect the area where the target is, as shown in figure 1. The detection of the area is obtained by employing a RGB camera with a Hopfield ANN. The generation and control of the trajectory are carried out from a mathematical model programmed (c ++) that is connected to the robotic manipulator and the vision sensor.

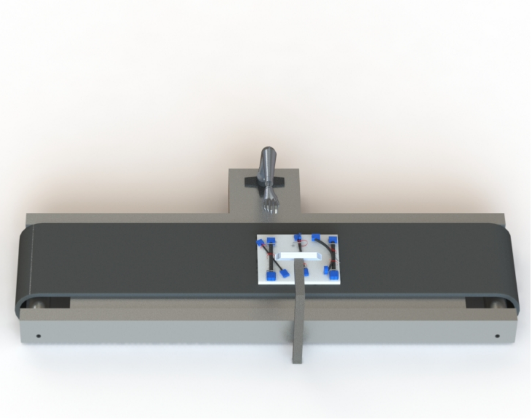


Figura 1: Trimetric view

1.1. Mechanical Design

The manipulator to be used is shown in 2, it is one of the most common manipulators in the industry because it resembles a human arm.

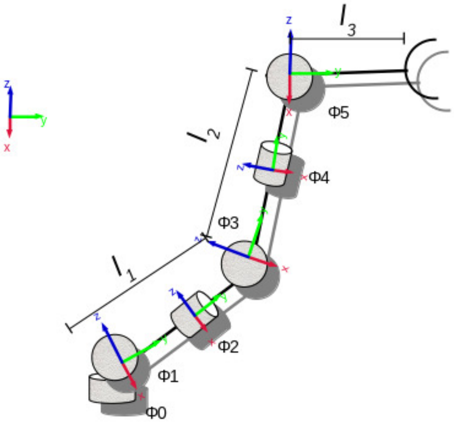


Figura 2: 6 DOF Manipulator

The manipulator's higher-order derivatives are deduced based on the arm's link-joints geometry in order to obtain the direct and inverse solutions.

$$\begin{aligned} x =& l_1 C_{(\phi_1} . S_{(\phi_0} + l_2 (C_{(\phi_1+\phi_3} . S_{(\phi_0} C_{(\phi_2} + S_{\phi_1+\phi_3} . S_{(\phi_2)} \\ &+ l_3 C_{\phi_1+\phi_3+\phi_5} . S_{(\phi_0} C_{(\phi_2} C_{(\phi_4} + S_{\phi_1+\phi_3} C_{(\phi_2} S_{(\phi_4} \\ &- C_{\phi_1+\phi_3+\phi_5} S_{(\phi_0} S_{(\phi_2} S_{(\phi_4} + S_{\phi_1+\phi_3+\phi_5} S_{(\phi 2} C_{(\phi_4)} \end{aligned} \tag{1}$$

$$y = (l_1 C_{\phi_1} + l_2 C_{\phi_1+\phi_3} + l_3 C_{\phi_1+\phi_3+\phi_5}) C_{\phi_0} \tag{2}$$

$$\begin{aligned} z =& l_1 S_1 + l_2 (-C_{\phi_1+\phi_3} S_{\phi_0} S_{\phi_2} + S_{\phi_1+\phi_3} C_{\phi_2}) \\ &+ l_3 (-C_{\phi_1+\phi_3+\phi_5} S_{\phi_0} S_{\phi_2} C_{\phi_4} + S_{\phi_1+\phi_3+\phi_5} S_{\phi_2} S_{\phi_4} \\ &- C_{\phi_1+\phi_3+\phi_5} S_{\phi_0} C_{\phi_2} S_{\phi_4} + S_{\phi_1+\phi_3+\phi_5} C_{\phi_2} C_{\phi_4}) \end{aligned} \tag{3}$$

This equations can be tested by making a graph. For example a graph of two rotations $\phi_3 = 90°$ and $\phi_5 = -90°$, as it can see in fig 3

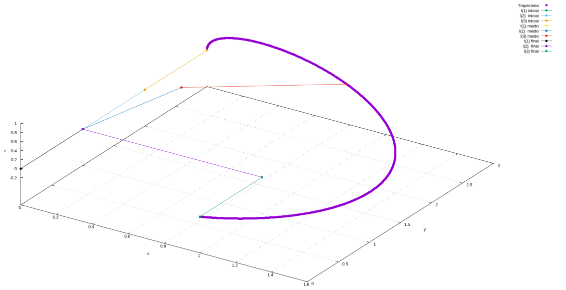


Figura 3: Trajectory

It follows from past position articulations that the principal request subordinates depicting the robot arm's Cartesian rates $\dot{p}$, as far as the joints turning speeds $\dot{\Phi}$, can be expressed in the direct kinematics equation $\dot{p} = J \cdot \dot{\Phi}$. The grid J is a nonfixed time-fluctuating Jacobian matrix.

$$J = \begin{bmatrix} \frac{\partial x}{\partial \phi_0} & \frac{\partial x}{\partial \phi_1} & \frac{\partial x}{\partial \phi_2} & \frac{\partial x}{\partial \phi_3} & \frac{\partial x}{\partial \phi_4} & \frac{\partial x}{\partial \phi_5} \\ \frac{\partial y}{\partial \phi_0} & \frac{\partial y}{\partial \phi_1} & \frac{\partial y}{\partial \phi_2} & \frac{\partial y}{\partial \phi_3} & \frac{\partial y}{\partial \phi_4} & \frac{\partial y}{\partial \phi_5} \\ \frac{\partial z}{\partial \phi_0} & \frac{\partial z}{\partial \phi_1} & \frac{\partial z}{\partial \phi_2} & \frac{\partial z}{\partial \phi_3} & \frac{\partial z}{\partial \phi_4} & \frac{\partial z}{\partial \phi_5} \end{bmatrix}$$

(4)

Then the angular position can be expressed in the linear matrix form $p = J\phi$

$$\vec{P}' = \mathbf{J}(\vec{\phi}).\vec{\phi}'$$

Inverse Jacobian matrix is used in order to find the approximate function of $\vec{\phi}$

$$\mathbf{J}(\vec{\phi})^{-1}.\vec{P}' = \mathbf{J}(\vec{\phi})^{-1}.\mathbf{J}(\vec{\phi}).\vec{\phi}'$$

. The Jacobian matrix is non stationary time-varying. Thus a integration is need it in order to find discrete values

$$\vec{\phi}' = \mathbf{J}(\vec{\phi})^{-1}.\vec{P}'$$

$$\int_{\phi_i}^{\phi_f} \frac{\vec{\phi}}{dt} = \mathbf{J}(\vec{\phi})^{-1}.\int_i^f \frac{\vec{P}'}{dt}$$

$$\phi_f - \phi_i = J^1\vec{\phi}.(\vec{P}_f - P_i) \tag{5}$$

2. Tau-Jerk Control

Tau theory is a bio-inspired method developed by David N Lee in 1976 [4] based on the perception of the human eye. Later, based on this theory, the movement of living beings is described calling the generalized theory of tau [5]. Zhen Zhang tested the general theory of tau using a robotic arm using the jerk of tau. In this papper this theory is recreated for a 3D manipulator with Newton Rapson's method for inverse kinematics. The equation that describes a Tau-jerk method to find Cartesian position 6 and angular position 7

$$p(t) = \frac{\triangle p}{T^{3/k_{p,q}}}(T^3 - t^3)^{1/k_{qp}} \tag{6}$$

$$\phi(t) = \frac{\triangle \phi}{T^{3/k_{p,q}}}(T^3 - t^3)^{1/k_{qp}} \tag{7}$$

However, if it is required to find the angles from a reference position, a method is necessary to find the inverse kinematics, because the Tau-Jerk method does not relate the position to its angles. Newton-Rapshon method can be used.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{8}$$

Following the previous step, a graph of the movement can be seen with an acceleration at the beginning and end of the movement.

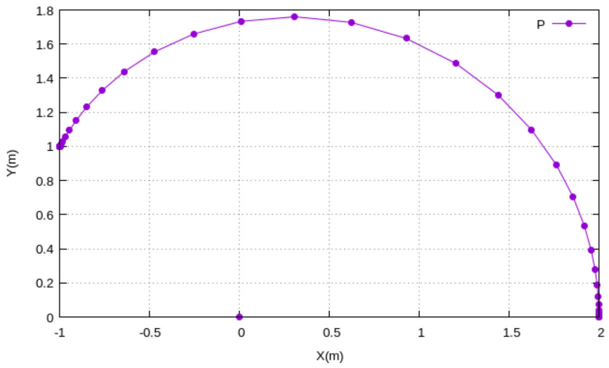


Figura 4: Tau-Jerk

3. Time variant Controller

According to algebra, solving an inverse kinematic equation 5 an adaptive control could be deducted like this:

$$\vec{\phi}_{t+1} = \vec{\phi_t} + \mathbf{J}\vec{\phi}^{-1}.(\vec{P}_{ref} - \vec{P}_t) \tag{9}$$

$$\vec{P}_{t+1} = \vec{P}_t + \mathbf{J}\vec{\phi}^{-1}.(\vec{\phi}_{t+1} - \vec{\phi}_t) \tag{10}$$

If what you have are the reference angles, you simply invert the equations. For the reduction of the error, it is given by the difference of the position, and in this experiment when it is less than 0.1. In addition, to make the simulation more similar to reality, it is multiplied by an attenuation constant, because a real motor cannot have such large changes. This constant is α and it can be from 0 to 1, with 1 being the highest value.

$$\vec{\phi}_{t+1} = \vec{\phi_t} + \alpha.\mathbf{J}\vec{\phi}^{-1}.(\vec{P}_{ref} - \vec{P}_t)$$

$$\vec{P}_{t+1} = \vec{P}_t + \alpha.\mathbf{J}\vec{\phi}^{-1}.(\vec{\phi}_{t+1} - \vec{\phi}_t)$$

Error is calculated with goal position

$$error = P_{goal} - P_t$$

Then a graph shows a linear behavior with more precision towards the end position

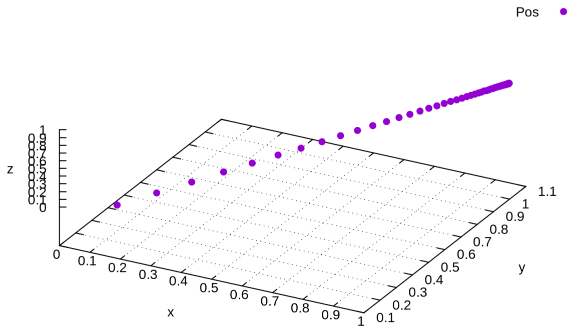


Figura 5: Algebraic adaptive and variant time method

Cartesian target positions are optimized, taking as a criteria the minimization of the distances of the roads and the number of turning points along a road. The smaller the number of inflection points, the more linear is a path, therefore closer than the other assembly target. where a polynomial with minimum and maximum values (inflection points). Set the roots through its derivative to detect the places of inflection points as roots. One second derivative establishes whether these roots are minimum or maximum values. For the top view, the polynomial assumes a Cartesian coordinate function y (x) processing a total of nT Cartesian measurement points,
Therefore, let $[y(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + ...a(n)x^n$ be a measure of the Cartesian trajectory for positions in the XY plane, the coefficients of the third degree polynomial are

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} nT & \sum_i x_i & \sum_i x_i^2 & \sum_i x_i^3 \\ \sum_i x_i & \sum_i x_i^2 & \sum_i x_i^3 & \sum_i x_i^4 \\ \sum_i x_i^2 & \sum_i x_i^3 & \sum_i x_i^4 & \sum_i x_i^5 \\ \sum_i x_i^3 & \sum_i x_i^4 & \sum_i x_i^5 & \sum_i x_i^6 \end{bmatrix} \cdot \begin{bmatrix} \sum_i y_i \\ \sum_i y_i x_i \\ \sum_i y_i x_i^2 \\ \sum_i y_i x_i^3 \end{bmatrix} \tag{11}$$

Y axis is calculated in function of x coordinate

$$y(x) = a_0 + a_1x + a_2x^2 + a_3x^3$$

First derivate

$$y'(x) = a_1 + 2a_2x + 3a_3x^2$$

Newton Rapson method

$$\hat{x}_{t+1} = \hat{x}_t - \frac{y'x}{y''(x)} = \hat{x}_t - \frac{a_1 + 2a_2x + 3a_3x^2}{2a_2 + 6a_3x} \tag{12}$$

Therefore, the optimized y is $y(\hat{x}) = a_0 + a_1x + a_2x^2 + a_3x^3$, to know the maximum and minimum we use the second derivative $y''(\hat{x}) = 2a_2 + 6a_3x$ therefore:

$$\begin{cases} max & \text{if } y''(\hat{x}) < 0 \\ min & \text{if } y''(\hat{x}) > 0 \end{cases} \tag{13}$$

The absolute minimum with the shortest magnitude among all polynomials is chosen as a function. And this model applies to xy or yz planes separately

4. Hopfield ANN

RGB camera is used to obtain the images of the assembly area. An RGB camera is a visual sensor that allows images to be obtained in their primary red, green and blue color values in a range of 8 bits (0-255) for each pixel (th).
An image can be represented as $I = I_R \cup I_G \cup I_B$, where I is a pixels matrix with size mxn. To segment a channel, for example red we would have to discriminate channels G and B so we have $I_c = I_R \cup (I_G \cap I_B)$, For white and black colors we have the same logical inference $I_R \cap I_G \cap I_B$, but white have a high RGB values th_{high} and black have low RGB values th_{low}. We consider a high threshold $th_{high} = 200$ and the low threshold $th_{low} = 50$ everything between that is gonna be gray. In the board we have four different types of objects to classify.

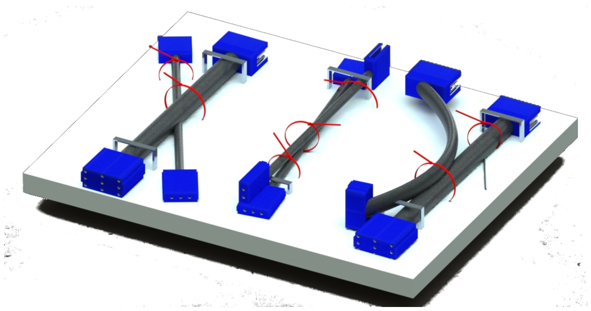


Figura 6: Example board for assembly

1. Board, $I_{white} = I_R \cap I_G \cap I_B$, RGB channel range th_{high}
2. Cable, $I_{black} = I_R \cap I_G \cap I_B$, RGB range th_{low}
3. Connector, $I_{blue} = I_B \cup (I_G \cap I_R)$, B range th_{high} and RG th_{low}

4. Cable Tie, $I_{red} = I_R \cup (I_G \cap I_B)$, R range th_{high} and GB th_{low}

The Hopfield network architecture is made up of a layer with all the neurons connected to each other, whose information travels back and forth. The inputs $x_i(t)$ are binary or polar ($\{0,1\}$ or $\{-1,1\}$). To Trains weights, we use a matrix W given by.

$$W_{ij} = \sum_{i=0}^{M-1} x_i^T \cdot x_j \tag{14}$$

such that, k = 1, 2, . . . , m

$$W = W_0 + W_1 + ... + W_m \tag{15}$$

Hence the total weights training matrix with vector input $xi = (x1, x2, x3, y1, y2)^T$ and three neurons with feedback of the first single layer Hopfield ANN is

$$w_i = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \end{bmatrix} (x_1, x_2, x_3, x_1, y_2) \tag{16}$$

Where, y_1 and y_2 are both of the other feedback neurons' output. When it's expected $y_a = 1 \Rightarrow y_1 = y_b, y_2 = y_c$ When it's expected $y_b = 1 \Rightarrow y_1 = y_a, y_2 = y_c$ and when expected $y_c = 1 \Rightarrow y_1 = y_a, y_2 = y_b$. Therefore,

$$w = \begin{bmatrix} W_{1,1}, W_{1,2}, W_{1,3}, W_{1,4}, W_{1,5} \\ W_{2,1}, W_{2,2}, W_{2,3}, W_{2,4}, W_{2,5} \\ W_{3,1}, W_{3,2}, W_{3,3}, W_{3,4}, W_{3,5} \\ W_{4,1}, W_{4,2}, W_{4,3}, W_{4,4}, W_{4,5} \\ W_{5,1}, W_{5,2}, W_{5,3}, W_{5,4}, W_{5,5} \end{bmatrix} \tag{17}$$

A neuron's inner vector si

$$s_i = \mathbf{W} - X_i \tag{18}$$

The neuron's output yi produces [+1, 1]. Any component sj of the vector si , either sj <0 will produce a negative output yj = 1, thus

$$y = \begin{cases} 1 & s_j > 0 \\ -1 & \text{otherwise} \end{cases} \tag{19}$$

To perform the once processed classification, the nxm image matrix is classified using the hopfield neural network where each class is a channel of the RGB image. x1 = R, x2 = G and x3 = B. The Hopfield network uses binary or polar values and the values in the RGB channels have a range of (0-255) so half 127.5 is used as threshold th. To improve results the Hopfield network will be used as a multilayer, having the network output as the input of the next layer in this way.

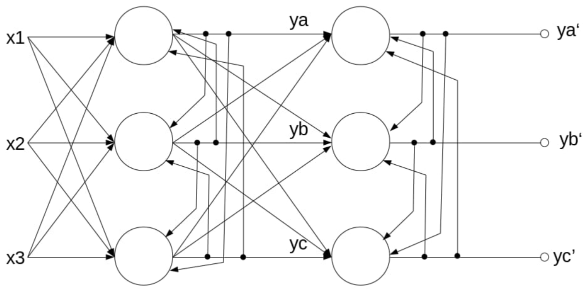


Figura 7: Hopfield Network

Class 1 Cables Ties

$$x_1 > th \cap x_2 <= th \cap x_3 <= th \tag{20}$$

Class 2 Connectors

$$x_1 <= th \cap x_2 < th \cap x_3 > th \tag{21}$$

Class 3 Board

$$x_1 > th \cap x_2 > th \cap x_3 > th \tag{22}$$

Class 4 Cables

$$x_1 <= th \cap x_2 <= th \cap x_3 <= th \tag{23}$$

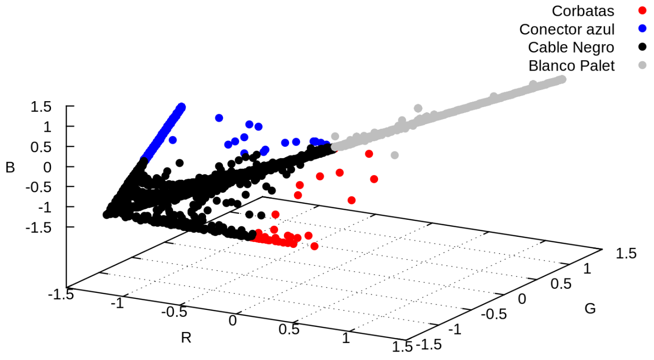


Figura 8: RGB Data at first layer

Then, we train the "w"matrix, and with results we make a truth table for the next clasify

ya	yb	yc	Class
1	1	1	Board
1	1	0	Fails
1	0	1	Fails
1	0	0	Target
0	1	1	Fails
0	1	0	Fails
0	1	0	Connector
0	0	0	Cables

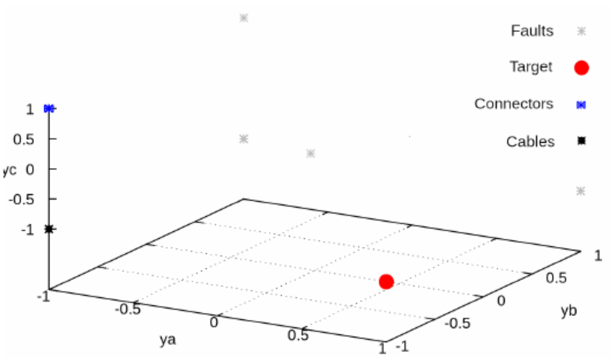


Figura 9: Hopfield Network graph

Then the position of ties is determined by the center of these points.

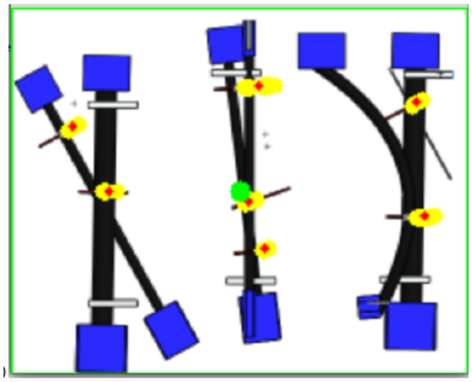


Figura 10: Board with goal points

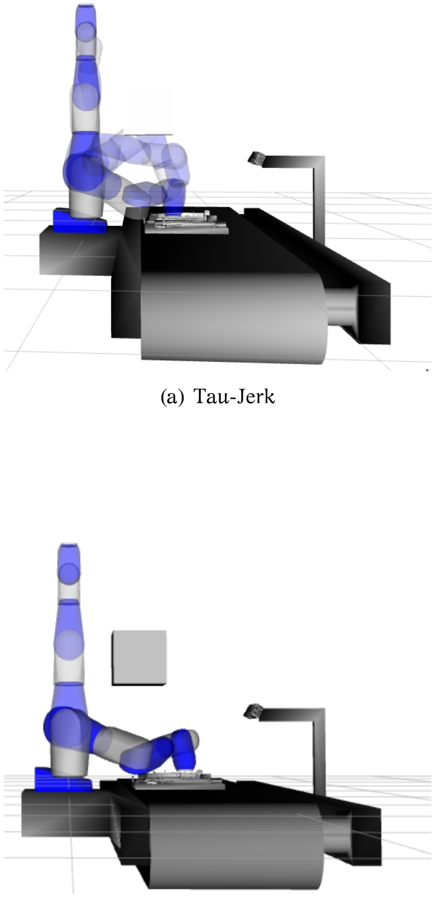
5. Simulation and Results

There are different simulators for robotics that are regularly based on physics engines that allow simulating the kinematics and dynamics of the robot to simulate the behavior of the robot in the most realistic way. Several simulators were considered, such as Project Chrono, which uses a real-time 3D engine that is programmed in c ++. In addition to taking advantage of the boost Chrono libraries, among other physics simulation packages. Also, this gazebo is a simulator specialized in robotics and player stage evolution that includes the controllers for several real robotic systems, allowing them to test them after the simulation. Finally, ROS was considered a package that unites several tools including gazebo. In addition to other packages that allow the design, simulation, and testing of robotic systems. A Yaskawa montoman manipulator was used, which is available for download on the MoveIt oficial web page.

To obtain the target areas, the image is first processed, in this work a png image is used simulating the taking of a real camera, and through a Hopfield network the objects are classified to obtain the desired points. For this, a new module called ANN was created within the ROS work-space where moveit is installed. This ANN network uses Armadillo, openCV, and various boost libraries which were appended to the cmake of the new ANN module together with its package.xml to run. When it obtains the destination zones, it creates the .txt file and communicates through a ROS node to the move group Taujerk / algebraic to notify it that the .txt file is already updated.

Then Tau-jerk module and the algebraic module solve the equations to obtain the position. Each new position obtained is sent to the manipulator with RVIz and this is in charge of performing the inverse kinematics and reaching the desired position, until reaching the desired destination area. In addition, an obstacle is simulated, which must be evaded by the manipulator. A T = 20 was used, so there were 20 iterations for each destination zone but only 10 positions since the first 10 are unreachable for the manipulator (close to 0). The target zones are obtained by communicating with the Hopfield neural network.

Robotic projects integrate different areas of knowledge to solve a complex problem, in this case, the trajectory was generated by classifying destination areas in a Hopfield network. Obtained from RGB image processing. The Hopfield network is a recurring single layer. However, the process is carried out twice to use it as if it were multilayer. The network turned out to be able to detect the ties by means of their colors. In addition, failures are reduced thanks to its second layer, although the results show some repeated points, the layers and the number of neurons can be increased to improve the results. You can also look for manufacturing solutions such as using white ties and red markers to mark specific areas.



(a) Tau-Jerk

(b) Time Variant Controller

Figura 11: ROS Simulation

Due to the complexity of multiple operations and the amount of data in the image processing, the use of parallel computing was required, which was implemented with a shared memory that runs in c++ code and the boost libraries.
The manipulator positions were calculated with the algebraic method of successive approximations. The monitoring of the positions in a linear way and control of the variables over time is a better option than the Tau-Jerk method in redundant manipulators.
For the simulation, both control methods were used by means of the Tau-jerk method, they find the positions and ROS is in charge of finding the inverse kinematics and the trajectory to be executed. Through the algebraic system we find the angular position of the robot and ros simulates the proposed path.

6. Conclusions

An algorithm was developed that allows the generation of trajectories through areas detected by a Hopfield network, which classifies different objects based on the pattern of colors obtained by RGB images. The data is transmitted through a shared memory, and trajectory monitoring and control for a robotic arm with 6 degrees of freedom are achieved with the method of successive approximations, feedback by the simulation of an RGB camera. The results are simulations based on the mathematical abstraction of the problem in question, and it is possible to theoretically prove that a 6-degree-of-freedom manipulator can perform a task in manufacturing autonomously through feedback from a vision system.
It is intended to implement the system in a real way for a company in Cd. Juarez, Chihuahua. In the Hopfield model, the recognition of the target areas by patterns formed by the shape of the object and adding it to a third layer of the neural network is proposed. Besides, the processes will be transferred to a prototype card such as parallel, cerebot, or to a process in the cloud.

Referencias

[1] A. Yalcin and T. O. Boucher, "Deadlock avoidance in flexible manufacturing systems using finite automata,ïn IEEE Transactions on Robotics and Automation, vol. 16, no. 4, pp. 424-429, Aug. 2000. doi: 10.1109/70.864237

[2] R. C. Luo and C. Kuo, Ïntelligent Seven-DoF Robot With Dynamic Obstacle Avoidance and 3-D Object Recognition for Industrial Cyber–Physical Systems in Manufacturing Automation,ïn Proceedings of

the IEEE, vol. 104, no. 5, pp. 1102-1113, May 2016. doi: 10.1109/JPROC.2015.2508598

[3] J. C. Restrepo, J. Villegas, A. Arias, S. Serna and C. Madrigal, "Trajectory generation for a robotic in a robocup test scenery using Kalman filter and B-spline curves,"2012 XVII Symposium of Image, Signal Processing, and Artificial Vision (STSIVA), Antioquia, 2012, pp. 110-115, doi: 10.1109/STSIVA.2012.6340566.

[4] David N Lee. A theory of visual control of braking based on information about time-to-collision.Perception, 5(4):437–459, 1976.

[5] David N Lee. General tau theory: evolution to date.Perception, 38(6):837, 2009.

[6] Zhen Zhang and Xu Yang. Bio-inspired motion planning for reaching movement of amanipulator based on intrinsic tau jerk guidance. Advances in Manufacturing, pages1–11

[7] S. S. Young, P. D. Scott and N. M. Nasrabadi, .ºbject recognition using multilayer Hopfield neural network,",Industrial Robot, abril 2020, Manuscrit id: IR-04-2020-0079.

[8] E. Salari and S. Zhang, Ïntegrated recurrent neural network for image resolution enhancement from multiple image frames,ïn IEE Proceedings - Vision, Image and Signal Processing, vol. 150, no. 5, pp. 299-, 22 Oct. 2003, doi: 10.1049/ip-vis:20030524.

[9] R. C. Luo and C. Kuo, Ïntelligent Seven-DoF Robot With Dynamic Obstacle Avoidance and 3-D Object Recognition for Industrial Cyber–Physical Systems in Manufacturing Automation,ïn Proceedings of the IEEE, vol. 104, no. 5, pp. 1102-1113, May 2016. doi: 10.1109/JPROC.2015.2508598

[10] J. C. Restrepo, J. Villegas, A. Arias, S. Serna and C. Madrigal, "Trajectory generation for a robotic in a robocup test scenery using Kalman filter and B-spline curves,"2012 XVII Symposium of Image, Signal Processing, and Artificial Vision (STSIVA), Antioquia, 2012, pp. 110-115, doi: 10.1109/STSIVA.2012.6340566.

[11] J. Kofman, Xianghai Wu, T. J. Luu and S. Verma, "Teleoperation of a robot manipulator using a vision-based human-robot interface,ïn IEEE Transactions on Industrial Electronics, vol. 52, no. 5, pp. 1206-1219, Oct. 2005. doi: 10.1109/TIE.2005.855696

[12] Y. Yamamoto, N. Maekawa, M. Hida, X. Yang, K. Aoyama, T. Kataoka, Y. He, K. Tatsuno, "Task performance tests on inserting the bolts by an experimental system for power distribution line maintenance - grope action under compliance control," Proc. 2012 Int. Symp.

[13] D. Prattichizzo, Robotics in second life,ïn IEEE Robotics Automation Magazine, vol. 16, no. 1, pp. 99-102, March 2009. doi: 10.1109/MRA.2009.932131

[14] H. I. Christensen, "Formulation of a U.S. National Strategy for Robotics [Industrial Activities],ïn IEEE Robotics Automation Magazine, vol. 19, no. 2, pp. 10-14, June 2012. doi: 10.1109/MRA.2012.2193931

[15] B. C. Purba, M. Diva Pasha and Y. Bandung, "Design and Implementation of WebRTC-Based Video Conference System in Odroid Board,"2018 12th International Conference on Telecommunication Systems, Services, and Applications (TSSA), Yogyakarta, Indonesia, 2018, pp. 1-6. doi: 10.1109/TSSA.2018.8708747

[16] H. Su, C. Yang, G. Ferrigno and E. De Momi, Ïmproved Human–Robot Collaborative Control of Redundant Robot for Teleoperated Minimally Invasive Surgery,ïn IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 1447-1453, April 2019, doi: 10.1109/LRA.2019.2897145.

[17] Barrientos, A.; Peñín, L.F,; Balaguer, C. Aracil, R. Fundamentos de Robótica 2ª Ed. McGraw-Hill, 2007

[18] Rodriguez-Jorge, Ricardo, et al. "Weight Adaptation Stability of Linear and Higher-Order Neural Units for Prediction Applications.Ïnternational Conference on Multimedia and Network Information System. Springer, Cham, 2018.

[19] Lee, David N. "General Tau Theory: evolution to date."Perception 38.6 (2009): 837.

Ivan Carvajal Carlos: Egresado del programa de ing. Mecatrónica por la Universidad Autónoma de Cd. Juaréz y maestrante de la Maestría de Cómputo Aplicado de la misma universidad. Actualmente desarrolla su investigación en el área de manufactura robótica flexible con manipuladores.