

CONTROL NUMÉRICO DE UN BRAZO ROBÓTICO, COMO RESPUESTA A ALGORITMOS DE PROCESAMIENTO DE IMÁGENES Y SISTEMAS EXPERTOS

Calderón Carlos, Ing.
Ramirez Tayron ,Ing.
Mora Cristian , Ing.

Universidad Técnica Particular de Loja

RESUMEN

En el presente proyecto se integraron tres áreas de investigación: robótica, procesamiento de imágenes e inteligencia artificial. Para demostrar dicha integración, se diseñó e implementó un sistema robot que desempeñe el juego Tic-Tac-Toe con adversario. Este sistema consta de tres retos: Desempeñar los movimientos de las piezas del juego. Discernir la disposición de las fichas tanto las del sistema como aquellas del adversario. Planificar la jugada con la capacidad de ganar.

Para cumplir estos retos, se diseñó e implementó un sistema de control, basado en cinemática inversa, de un brazo robot de cinco grados de libertad, dotado de capacidad visual mediante una cámara y algoritmos de visión artificial y reconocimiento de patrones, por último, el sistema robot se complementa con un sistema experto que le permite desempeñar el juego utilizando inteligencia artificial. Los algoritmos de: cinética inversa, visión artificial y sistemas expertos, se diseñaron y desarrollaron bajo la plataforma de programación LabVIEW.

Como resultado, se llegó a obtener un sistema robusto a la variabilidad tanto de condiciones de iluminación como de orientación, así como también se obtuvo un tiempo de respuesta mínimo.

ABSTRACT

In the present project were integrated three areas of investigation: robotics, image processing and artificial intelligence. To demonstrate this integration, it was designed and it implemented a system robot that carries out the game Tic-Tac-Toe with opponent. This system consists of three challenges: To carry out the movements of the pieces of the game. To discern the disposition of the pieces of the system and of the opponent. To plan the play with the capacity to win.

To complete these challenges, it was designed and it implemented a control system, based on inverse kinematics, of an arm robot of five grades of freedom, endowed with visual capacity by means of a camera and algorithms of artificial vision and recognition of patterns, ultimately, the system robot is complemented with an expert system that allows him to carry out the game using artificial intelligence.

The algorithms of: inverse kinetics, artificial vision and expert systems, they were designed and they

developed under the programming platform LabVIEW.

As a result, was obtained a robust system to the variability of conditions of illumination and of orientation, as well as, was obtained a minimum time of response.

1. INTRODUCCIÓN

El presente proyecto se enfoca en la integración de tres áreas de investigación en el área de Electrónica como son: robótica, procesamiento digital de imágenes e inteligencia artificial.

La robótica estudia el modelamiento y construcción de máquinas capaces de desempeñar tareas, sea autónomamente o de forma controlada. El Procesamiento de imágenes, es el conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar la calidad o facilitar la búsqueda de información; Por último, la Inteligencia Artificial es la ciencia que intenta la creación de algoritmos orientados a automatización de sistemas que requieran un comportamiento inteligente.

Ahora bien, nuestro objetivo es demostrar que estas ramas de la ingeniería se pueden complementar para desempeñar una tarea de una manera integral, así como también demostramos la capacidad de una plataforma en específico (LabVIEW) para llevar a cabo las aplicaciones que desempeñaban las funciones antes mencionadas.

Cabe recalcar que la integración de estas áreas se encuentra dirigida hacia aplicaciones flexibles y adaptables de Control y Automatización Industrial, área que necesita nuestra atención en aquellos países que queremos incrementar la productividad y mejorar la calidad de los productos, pero que no poseen el presupuesto para invertir en grandes sistemas fijos de automatización, sino mas bien, invertir en sistemas flexibles y re-configurables.

El lenguaje utilizado para diseñar y desarrollar los algoritmos de cinética inversa, visón artificial y sistemas expertos se desarrollaron bajo la plataforma de programación LabVIEW. Estos algoritmos se comunican con dos dispositivos periféricos para el ingreso de datos y para la ejecución de respuestas.

Estos dispositivos periféricos son: en primera instancia, una cámara de video que suministra al sistema la capacidad visual del entorno de juego; y en

segunda instancia, un brazo robot que le permite al sistema desempeñar los movimientos planificados por el sistema experto.

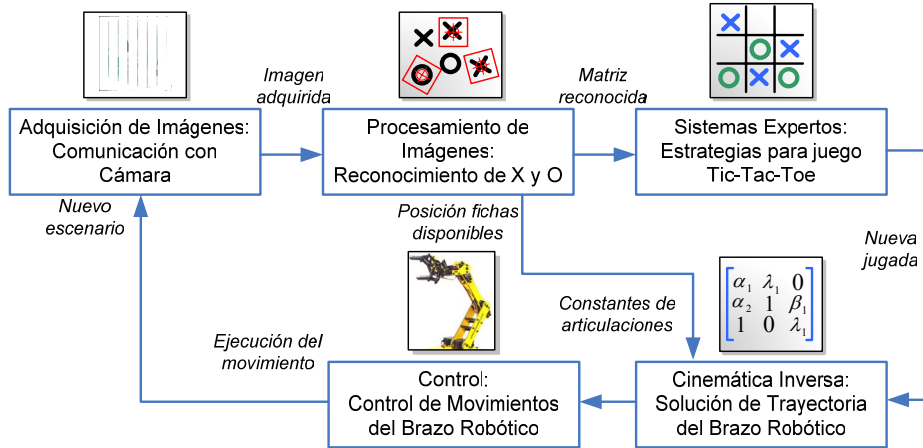


Figura 1. Esquema general del sistema. Integración de funciones entre Hardware y Software. Hardware: Brazo Robótico y Cámara de video; Software: Procesamiento de Imágenes, Cinemática inversa y Sistema Experto;

2. ESQUEMA GENERAL DEL SISTEMA

Se implementó el sistema, teniendo en cuenta dos segmentos claramente definidos: el software, el cual cumple las funciones de análisis y procesamiento de datos; y el hardware, el cual cumple las funciones de adquisición de la imagen y realización de movimientos de un punto a otro.

En cuanto al hardware. Para adquirir la imagen se utilizó una cámara de video iFire IEEE-1394 y para la realización de los movimientos requeridos se utilizó el brazo robot Lynx 5. En cuanto al software. Para analizar datos, procesar datos y comunicarse con el hardware utilizado, se utilizó el Lenguaje de Programación LabVIEW y su módulo IMAQ-Vision.

La forma de operabilidad de los componentes del sistema se esquematiza en la Figura 1. La imagen adquirida por la cámara es procesada con el objetivo de identificar el área de juego, identificar el lugar de ubicación de las fichas del juego y ensamblar una matriz en la que consten los tres estados: casilla vacía, casilla con equis (X) y casilla con círculos (O). Luego esta matriz numérica, se envía al algoritmo de sistemas expertos el cual la descompone con el objetivo de analizar las posiciones ocupadas por el sistema robot y aquellas ocupadas por el adversario y de allí deducir la siguiente jugada que mejor se adapte al estado actual del juego. Luego esta respuesta es interpretada por el algoritmo de cinemática inversa, el cual determina la configuración de los ángulos respectivos a las articulaciones del brazo robot, para desempeñar dos movimientos en cascada: el primero, dirigirse a recoger la ficha en lugar que se encuentre y el segundo, colocarla en la posición que ha determinado como respuesta el sistema experto.

Los algoritmos diseñados e implementados se describen en las secciones siguientes.

3. MÓDULO DE ADQUISICIÓN Y PROCESAMIENTO DE IMÁGENES

La tarea principal de este módulo es adquirir y analizar la señal de imagen, que es la que monitorea en tiempo real, tanto la matriz que nos sirve como escenario de juego como la posición de las fichas disponibles para la futura manipulación del brazo robótico.

Este sistema me arroja como respuesta la ubicación de la matriz de juego y sus respectivas posiciones vacías y llenas, así como también me arroja las posiciones en las que están las fichas disponibles para las futuras jugadas. En la Figura 2 se muestran los algoritmos que conforman la etapa de procesamiento de imágenes.

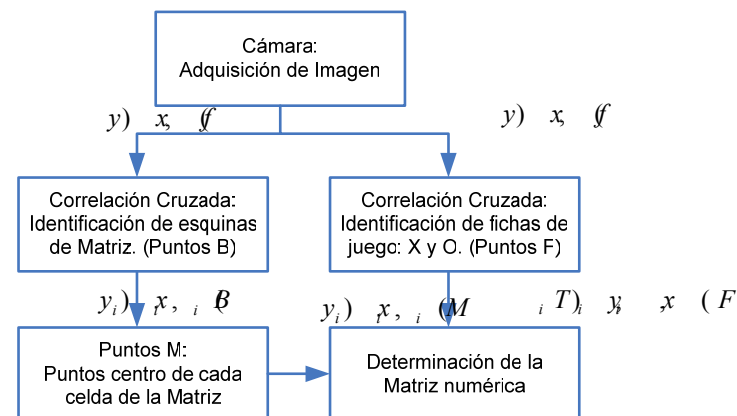


Figura 2. Secuencia de Algoritmos que conforman el Módulo de Procesamiento Digital de Imágenes.

3.1 Buscar los Puntos B

Los Puntos B, se los definió como aquellas coordenadas pertenecientes a las esquinas de la matriz de juego. El objetivo de identificarlos es para obtener la región de interés en la cual realizar la búsqueda de las fichas en juego, así como también para definir las fronteras de la matriz y sus respectivas celdas.

Para obtener los puntos B, nos basamos en la búsqueda de patrones en donde el modelo (esquina) es buscado alrededor de la imagen. El algoritmo de búsqueda utiliza el método de correlación cruzada, el cual se describe a continuación:

3.1.1 Método de búsqueda basado en Correlación Cruzada

Considere el patrón a buscar (subimagen) $w(x,y)$ de tamaño $K \times L$, en una imagen $f(x,y)$ de tamaño $M \times N$, donde $K \leq M$ y $L \leq N$. La correlación C entre $w(x,y)$ y $f(x,y)$ en el punto (i,j) está dada por la ecuación (1):

$$C(i,j) = \sum_{x=0}^{L-1} \sum_{y=0}^{K-1} w(x,y) f(x+i, y+j) \quad (1)$$

Donde $i = 0, 1, \dots, M-L$, $j = 0, 1, \dots, N-L$, y la sumatoria es tomada como la región en la imagen donde w y f se traslapan. La Figura 3 ilustra geoméricamente el procedimiento de correlación.

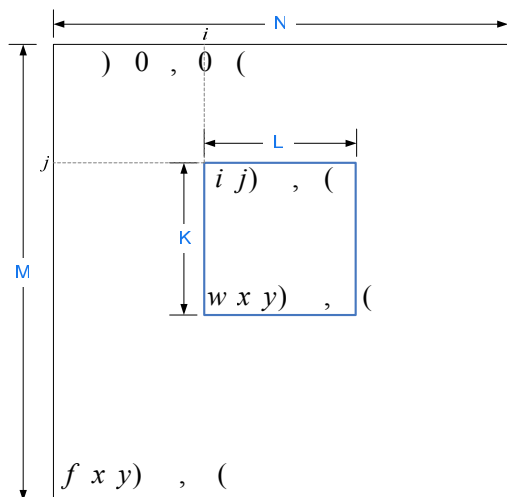


Figura 3. Procedimiento del método de búsqueda de patrones basado en el algoritmo de correlación cruzada.

De la Figura 3, se deduce que la correlación es el proceso de mover la sub-imagen w a través del área de la imagen f , y calcular el valor de C en ésta área. Esto se obtiene multiplicando cada píxel de la plantilla con el píxel de la imagen que se traslapa y posteriormente sumar los resultados. El máximo

valor de C indica la posición donde w se iguala de mejor manera con f .

En base a lo anterior, el patrón $w(x,y)$ a buscar es el modelo que corresponde a las cuatro esquinas. En la Figura 4 se muestra el modelo a buscar $w(x,y)$.

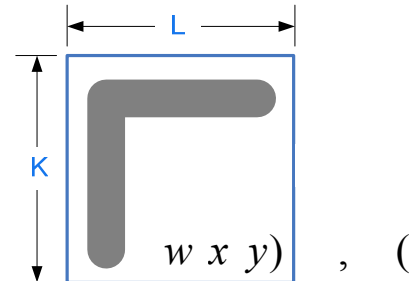


Figura 4. Patrón a buscar $w(x,y)$. Este patrón se correlacionará progresivamente con la imagen en tiempo real $f(x,y)$, para obtener los Puntos B.

La imagen en la cual se va a buscar el patrón mostrado en la Figura 4, es la imagen en tiempo real recibida desde la cámara de video. Ahora bien, teniendo listo tanto el patrón como la imagen en tiempo real, y aplicando el método de búsqueda basado en Correlación Cruzada, obtenemos los 4 puntos B con sus respectivas coordenadas. En la Figura 5 se muestra la ubicación de éstos puntos.

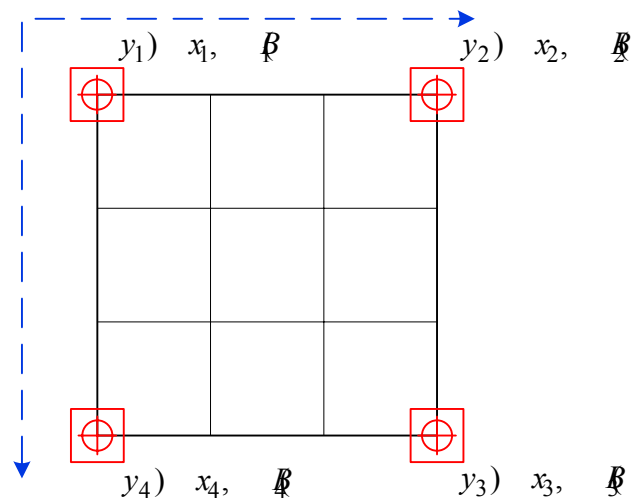


Figura 5. Dentro de la imagen $f(x,y)$, se muestra la ubicación de los Puntos B, con sus respectivas coordenadas.

3.2 Determinar los Puntos M

Una vez encontrados los Puntos B, se procede a encontrar la constante A , la cual es la longitud del lado de la matriz de juego. La ecuación (2) muestra la substracción de coordenadas, de las cuales resulta la constante A .

$$A = |x_2 - x_1| = |y_3 - y_4| \quad (2)$$

A partir de la constante A obtenemos la longitud de lado de cada una de las nueve celdas que forman la matriz, a esta longitud la denominamos constante $?$. La ecuación (3) indica el cálculo de esta longitud, y por último con ayuda de la constante $?$, determinamos las coordenadas de los Puntos M. Los Puntos M son los puntos centrales de cada una de las celdas de la matriz de juego. Las ecuaciones (4) muestran el cálculo de las coordenadas de los Puntos M.

$$? \approx \frac{A}{3} \quad (3)$$

$$\left\{ \begin{array}{l} a \quad x_1 \approx \frac{?}{2} x \\ b \quad x_1 \approx 3 \frac{?}{2} x \\ c \quad x_1 \approx 5 \frac{?}{2} x \\ a \quad y_1 \approx \frac{?}{2} y \\ b \quad y_1 \approx 3 \frac{?}{2} y \\ c \quad y_1 \approx 5 \frac{?}{2} y \end{array} \right. \quad (4)$$

En la Figura 6 se muestra la ubicación de los Puntos M, con sus respectivas coordenadas dentro de la imagen $f(x,y)$.

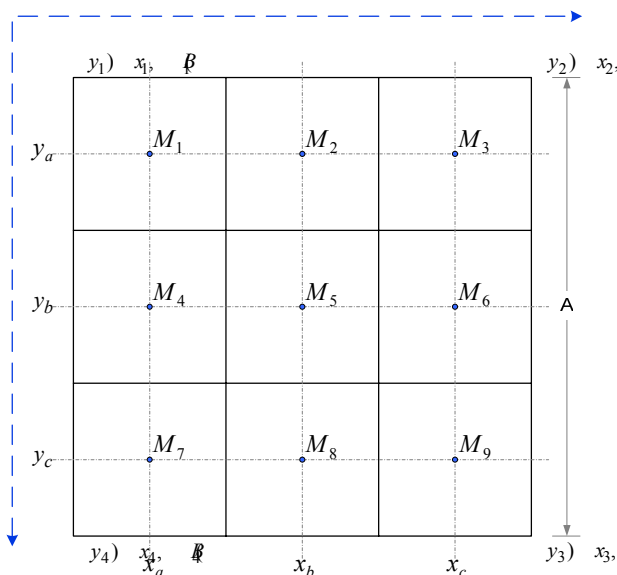


Figura 6. Dentro de la imagen $f(x,y)$, se muestra la ubicación de los Puntos M, con sus respectivas coordenadas.

La ubicación de estos puntos es muy importante ya que en cada una de las celdas se determina un área de validación para las fichas que se sitúan dentro de la matriz. Dicha área es un círculo que cubre el 75 % del área de la celda. En la Figura 7 se muestra el

esquema del establecimiento del área de validación que se formula para cada una de las celdas.

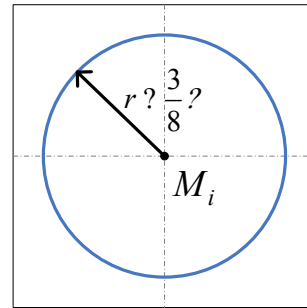


Figura 7. Área de validación, establecido para cada una de las celdas de la matriz de juego.

El área de validación nos sirve para determinar a qué celda pertenecen las fichas encontradas sobre la matriz de juego. En la sub-sección siguiente mostraremos como se encuentran los puntos centrales de las fichas (Puntos F).

3.3 Buscar los Puntos F

En esta sección mostraremos cómo encontrar las coordenadas de los Puntos F. Los Puntos F son los puntos centro de las fichas que se sitúan en la matriz de juego. También mostraremos como validar la pertenencia de una ficha a determinada celda, a través del área de validación.

Para obtener los Puntos F, nos basamos en la búsqueda de patrones en donde el modelo (ficha X y ficha O) es buscado alrededor de la imagen. El algoritmo de búsqueda utiliza el método de correlación cruzada, el cual se describió en la sección 3.1.1.

En la Figura 8 se muestra los patrones $w(x,y)$ a buscar en la imagen en tiempo real.

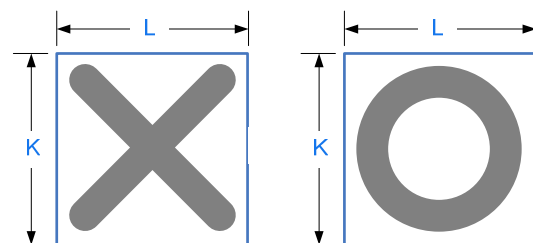


Figura 8. Patrones de las fichas que se correlacionarán progresivamente con la imagen en tiempo real $f(x,y)$, para obtener los Puntos F.

Los Puntos F son importantes ya que indican la coordenada en la que está ubicada la ficha, así como también su tipo. Los Puntos F son de la forma: $F_i(x_i, y_i, T)$, donde $T=1$ si la ficha es cruz (X) y $T=2$ si la ficha es círculo (O).

La etapa final del módulo de visión artificial es reconstruir la matriz numérica. Para llevar a cabo este objetivo, se siguió el siguiente proceso:

Para cada celda, se encuentran las distancias desde su Punto M hasta todo el conjunto de los Puntos F. Y luego se evalúa cuál de los Puntos F pertenece a la celda, y para determinarlo se comparan las distancias $M_i F_i$ con r . Si esta distancia es menor a r , la ficha pertenece a la celda, caso contrario, no pertenece a la celda analizada. En la Figura 9, se muestra el análisis de las distancias entre Puntos M - Puntos F y su relación con el radio r .

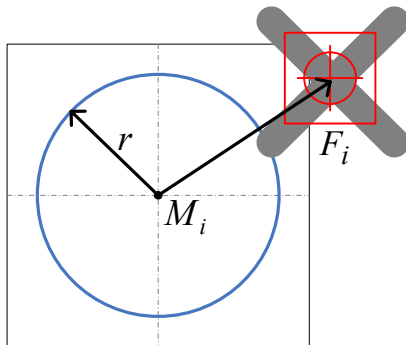


Figura 9. Relación de las distancias entre Puntos M-Puntos F y el radio r del Área de validación

Una vez construida la matriz numérica a partir del Módulo de Adquisición y Procesamiento de Imágenes, se envía al siguiente módulo que se dedica a tomar las decisiones de los turnos en lo que respecta al juego, este módulo es el de Estrategias del Juego.

4. MÓDULO DE ESTRATEGIA DEL JUEGO

La finalidad del presente módulo es de planificar la siguiente jugada del sistema, en base al estado actual de la partida del juego.

El juego Tic-Tac-Toe, es un juego de estrategia basándose en una matriz de tres filas por tres columnas (matriz 3x3), donde el objetivo es que el primer jugador en completar: una fila, columna o diagonal, con sus respectivas fichas sea el ganador de la partida. En la siguiente sección, se mostrará el algoritmo que sigue el sistema robot.

4.1 Algoritmo

Este Algoritmo esta pensado para lograr la mayor eficiencia posible, es por ello que este mismo algoritmo es capaz de detectar si el sistema robot tiene la posibilidad de ganar o de bloquear las acciones que realice su oponente.

Ya que este juego se basa en una matriz de 3 x 3, para un mejor desempeño del algoritmo, la hemos transformado en un vector de 9 elementos. En la Figura 10, se muestra las equivalencias de las

posiciones, al realizar la conversión de matriz a vector.

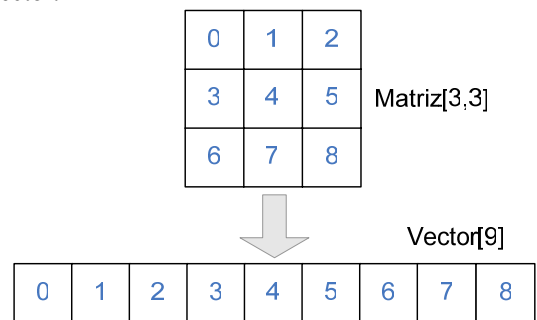


Figura 10. Equivalencias de las posiciones de las piezas a la conversión de matriz a vector.

Al tener un arreglo lineal, pudimos dejar como constantes cada una de las posiciones antes mencionadas. Este proceso hace que se haga más rápida la comprobación, formando combinaciones que nos dan como resultado las filas, columnas y diagonales, para su procesamiento.

Al ingresarlas al algoritmo se toma los tres elementos que corresponden, ya sea a una fila, columna o diagonal, las cuales van entrando en su respectivo orden, formándose de esta manera un subvector que consta de los elementos $[U_1, U_2, U_3]$ y es el que va a ser comparado.

Según como vayan entrando a ser analizadas las filas, columnas y diagonales, van recibiendo un nombre tipo constante, filas y columnas (entre 1 y 3) y diagonales entre (1 y 2), la Figura 11 muestra las etiquetas que se añaden a filas, columnas y diagonales.

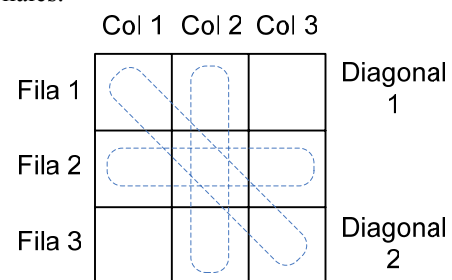


Figura 11. Etiquetas que se añaden a filas, columnas y diagonales.

Al realizar las sucesivas comparaciones, se obtiene un nuevo vector de tipo booleano, llamado vector $V = [X_1, X_2, X_3]$, a este vector V se le realiza una suma binaria de todos sus elementos tomando como referencia que si es verdadero es igual a 1, y si es falso es igual a 0. La ecuación (5) muestra la fórmula utilizada para la suma binaria del vector V.

$$\sum_{i=1}^N ? \quad i] \quad V \quad R \quad (5)$$

Cuando se obtiene el resultado R , si es diferente de 0, entonces se vuelve a crear un nuevo vector (T), pero ahora con las etiquetas de cada una de las filas, columnas o diagonales $T = [etiqueta_1, etiqueta_2, etiqueta_3]$.

Con este nuevo vector T hacemos el recorrido de las posiciones buscando cual es la posición que dio verdadero al inicio, lo cual afirma que hay dos fichas del mismo jugador en una misma fila, columna o diagonal, esto nos sirve para identificar si el siguiente turno se intenta ganar o bloquear al oponente, bajo este principio podemos identificar con exactitud en que fila está la posición ideal, este resultado es almacenado en la variable S (etiqueta de la fila, columna o diagonal a la que pertenece la posición ideal).

Nuestro algoritmo es capaz de analizar simultáneamente las tres filas, al no encontrar un resultado satisfactorio pasa a las dos diagonales y a las tres columnas logrando así mayor eficacia en el procesamiento, este proceso nos ha permitido obtener los tres resultados al mismo tiempo, y luego sometidos a un grafo nos permite ponderar los tres resultados obtenidos en las filas, columnas y diagonales, para luego desechar los menos útiles y seleccionar el mejor, el cual va hacer que gane la partida o que impida que gane el oponente.

5. MÓDULO DE SOLUCIÓN DE TRAYECTORIA DEL BRAZO ROBOT

Luego que se determina la mejor jugada, el siguiente paso es realizar la equivalencia entre aquella jugada y su posición en el espacio de cobertura del brazo robot, con el objetivo de mover las piezas del lugar en el que se encuentran al lugar correspondiente en la matriz de juego.

Con el objetivo de que el brazo robot responda a las órdenes con exactitud y precisión se debe tener un modelo matemático. Este modelo matemático se lo conoce como “cinemática inversa”.

El brazo robot manipulador que se utilizó para la presente tarea es un brazo con 5 grados de libertad incluyendo el movimiento de apertura y cierre de la pinza. El movimiento relativo en las articulaciones resulta en el movimiento de los elementos que posicionan el brazo en una orientación deseada. En la mayoría de las aplicaciones de robótica, se está interesado en la descripción espacial del efector final (pinza) del manipulador con respecto a un sistema de coordenadas de referencia fija, para lo cual necesariamente se debe resolver el problema de la cinemática inversa. Denavit y Hartenberg propusieron un método matricial para resolver en forma sistemática y generalizada este problema.

La cinemática inversa consiste en hallar el vector de ángulos de las articulaciones $q=[q1,q2,q3,q4]$ a partir de la orientación y posición del efector final, el cual es un problema de difícil solución debido a que incluye ecuaciones no lineales y múltiples soluciones.

Para la resolución del control del brazo robot tenemos como variables de entrada la posición de cada una de las fichas reconocidas y además la jugada que se va a realizar en función de la matriz de salida procesada en el bloque del algoritmo del juego. Con lo que se procede al cálculo de las trayectorias para que el efector final del brazo robot tome una ficha de la mesa y luego la ubique en la posición de la jugada deseada. Esta secuencia de tareas se esquematiza en la Figura 12.

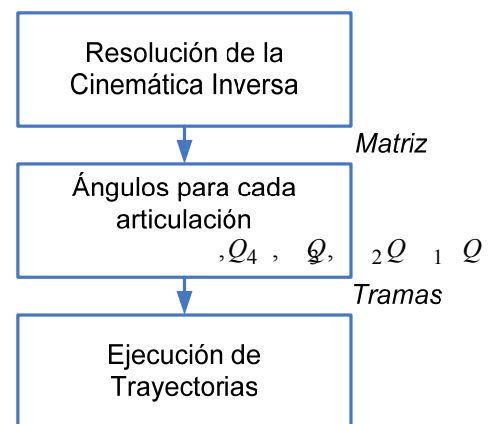


Figura 12. Secuencia de tareas que involucra el Módulo de Solución de Trayectoria del brazo robot.

5.1 Ejecución de la trayectoria

Para realizar el proceso de movimiento de cada uno de los servos en función del cálculo de los ángulos de las articulaciones obtenido a través de la cinemática inversa, procedemos a crear las tramas basándonos en el software firmware version SSC32-1 que es el que posee la tarjeta de control del brazo robótico para el movimiento en conjunto de todos los servos la cual presenta la siguiente estructura:

```
# <ch>_P<pw>_S<spd>_#<ch>_P<pw>_
S<spd>_T<time>_<cr>
```

<ch> = numero de canal del 0-31.

<pw>= ancho de pulso en microsegundos desde 500-2500.

<spd>= velocidad de movimiento en us por cada canal (opcional).

<time>=velocidad de movimiento en ms, afecta a todos los canales , 65535 max.

<cr>= carácter enter, código ascii 13 (requerido para inicial la acción).

Para calcular el ancho de pulso de cada uno de los servos, nos basamos en la ecuación (6).

$$X = \frac{t}{90} * 0.5 * 1000 \quad (6)$$

Donde X (ancho de pulso) es la salida de ancho de pulso en función de t (tiempo ms) que va en un rango de 500us-2500us. Cada una de las tramas de los servos las unimos y creamos una trama general que la escribimos a través del puerto serial para realizar el movimiento.

6. RESULTADOS

En el desarrollo e implementación del presente proyecto, se obtuvo los siguientes resultados:

- Se ha diseñado e implementado un sistema de control de un brazo robot, donde el brazo robot desempeñe sus movimientos en base a algoritmos de visión artificial e inteligencia artificial.
- Se ha investigado todo el fundamento matemático, de algoritmia y modelado, acerca de la integración de tres áreas de investigación en las áreas de Electrónica y Computación, como son: robótica, procesamiento de imágenes, e inteligencia artificial, con el objetivo de enfocarlos hacia aplicaciones de Control y Automatización y logrando añadirles cierto grado de inteligencia en la toma de decisiones de dichos sistemas.
- Se ha desarrollado (implementado) un prototipo económico, completamente modular, lo que lo hace flexible para aplicaciones de laboratorio en las diferentes universidades donde se impartan la ingeniería orientada a la integración de soluciones.
- Se ha completado una investigación que sirve como aporte a la implementación orientado al ámbito industrial. Se ha desarrollado un modelo de integración de estas líneas de investigación, con el objetivo de implementar un sistema industrial igual de flexible y adaptable a los requerimientos de cada necesidad.

7. CONCLUSIONES

- El diseño es potencialmente aplicable tanto a la Industria como a la Academia, para ayudar en el Control de Procesos en Industrias Locales y Regionales.
- Los resultados obtenidos permitieron comprobar la facilidad con la que se integra tecnologías en el lenguaje de Programación LabVIEW. A la vez hemos comprobado que LabVIEW posee funciones y capacidades de comunicación con diversos dispositivos y a la vez con diversos protocolos de comunicación.

- Las tecnologías relacionadas a la inteligencia artificial hacen de cualquier sistema, un sistema capaz de absorber y aprender información, dotando así de una buena capacidad para tomar decisiones y por ende proporcionar una buena respuesta.

8. REFERENCIAS

- [1] Relf Christopher, *Image Acquisition and Processing with LabVIEW*, Editorial CRC Press, 2004.
- [2] National Instruments, *IMAQ Vision User Manual*, National Instruments, Austin, Texas, August 2005.
- [3] Robert H. Bishop, *Learning with LabVIEW 8*. Editorial Prentice Hall, 2006.
- [4] P. J. McKerrow, *Introduction to Robotics*, Sidney: Addison-Wesley Publishing Company.
- [5] E Keogh, P Smyth, *A probabilistic approach to fast pattern matching in time series databases*, Proceedings of the 3rd International Conference of Knowledge, 1997.
- [6] R. O. Duda, P. E. Hart, *Use of the Hough transformation to detect lines and curves in pictures*, Communications of the ACM, 1972.
- [7] M Yamamoto, K Koshikawa, *Human motion analysis based on a robot arm model*, Computer Vision and Pattern Recognition Proceedings, 1991.
- [8] Lee, C., *Robot arm kinematics, dynamics, and control*, Computer. Vol. 15, 1982.
- [9] Efraim Turban, Paul R. Watkins, *Integrating Expert Systems and Decision Support Systems*, *MIS Quarterly*, Vol. 10. 1986.

e-mail: cacalderon@utpl.edu.ec