

Visual Control Law Tuning Using the JADE Algorithm Applied to Leaf Detection and Cutting

J. Cebada-Reyes, P. Wiederhold, P. Sánchez-Sánchez, and I. López-Cruz

Abstract— In the propagation of plant tissue *in vitro* it is necessary to obtain from the leaf a circular section cut. In order to achieve this type of cuts the visual feedback is used, allowing us to analyze the RGB planes to detect the green color and find the image centroid. In order to fulfill the cut objective a PD+ control is used with self-tuning based on the adaptive differential evolutionary algorithm JADE.

Index Terms— Plant tissue analysis, Leaves cutting, Circular control trajectory, Centroid, visual-servoing.

I. INTRODUCCIÓN

LA propagación *in-vitro* se basa en la habilidad de re-generar plantas completas a partir de la diferenciación y re-diferenciación de células vegetales, generalmente provenientes de plantas maduras. La regeneración es potenciada por dos vías morfogénicas: la organogénesis y la embriogénesis somática [1], [2]. Estas respuestas se pueden obtener a través del cultivo de secciones de hoja, regularmente circulares de 0.02 m², para obtener material totipotente e inducirlos en un medio de cultivo, y así generar las condiciones artificiales para producir un individuo nuevo [2]. Este proceso, al ser repetitivo, permite desarrollar tecnología cuyo objetivo es mejorar este tipo de tareas al hacer más eficiente el corte del tejido de las hojas. En este contexto, la robótica ha recibido considerable atención dentro del campo de la ingeniería agrícola [3], ya que los robots son capaces de hacer múltiples tareas de forma autónoma y repetitiva, bajo ciertas restricciones en su espacio de trabajo, puesto que se encuentran dotados de sensores que les permiten interactuar con el ambiente, y sus algoritmos de control pueden ser reprogramados [3]–[5]. Sin embargo, en la práctica, la sintonización de las ganancias es un procedimiento engorroso ya que en algunos casos hay que linealizar el sistema y en otros ajustar las ganancias de forma empírica (prueba y error) [5]. Esta forma de proceder genera incertidumbre en la obtención de las ganancias adecuadas. En este sentido la sintonización de ganancias puede formularse como un problema de optimización a través de métodos globales y búsqueda heurística como algoritmos estocásticos, algoritmos evolutivos y bio-inspirados [6], [7]. Los algoritmos evolutivos se caracterizan por basarse en operadores evolutivos los cuales son: el operador de selección, el operador de cruzamiento y el operador de mutación. Cada algoritmo evolutivo

implementa estos operadores de manera diferente, pero todos comparten esta característica. Algunos operadores evolutivos son estocásticos. Por ejemplo, en evolución diferencial son estocásticos tanto el operador de mutación como el operador de cruzamiento. El operador de selección es determinístico [6]–[8]. Actualmente destaca el algoritmo JADE (*the adaptive differential evolution with optional external archive*), cuya característica más importante es que el **factor de mutación** y el **factor de cruzamiento** se auto-ajustan en cada iteración a diferencia del algoritmo de evolución diferencial se ajusta los parámetros **factor de mutación** y el **factor de cruzamiento** de manera manual [10]. Estos algoritmos se han utilizado ampliamente para la sintonización de controladores tipo PID en diversas aplicaciones industriales [10]–[12]. En estos trabajos se da la pauta para extender la propuesta de sintonización a controladores cuya estructura incluye una parte proporcional y otra derivativa tal como en el controlador PD+ propuesto en [13]. Esta estructura de control incluye el control proporcional del error de posición, el control proporcional del error de velocidad más la dinámica completa del robot manipulador, brindando robustez [5], [13]. En este sentido es importante determinar las ganancias adecuadas para que el sistema cumpla con la tarea asignada. El artículo se ha organizado como sigue: en la Sección II se define el modelo dinámico del robot, así como algunas propiedades y características del sistema de visión, el control y la sintonización. En la Sección III se describe la planta y en la Sección IV se muestran los resultados experimentales. Finalmente, las conclusiones se presentan en la Sección V.

II. MODELADO

Considérese un robot cartesiano de tres grados de libertad con una cámara puesta en el área de trabajo en la configuración cámara-fija, Figura 1.

A. Dinámica del Robot

El modelo dinámico de un robot manipulador juega un papel importante en la simulación de movimiento y el análisis de estructuras de control. La ecuación dinámica de un robot manipulador de n grados de libertad está basada en la metodología de Euler-Lagrange [5], [14], [15], y está definida como

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \quad (1)$$

donde $q \in \mathbb{R}^n$ es el vector de coordenadas generalizadas, $H(q) \in \mathbb{R}^{n \times n}$ es la matriz de masas e inercias, la cual es simétrica y definida positiva, $C(q, \dot{q})\dot{q} \in \mathbb{R}^n$ es el vector de

J. G. Cebada-Reyes, Tecnológicos Nacionales de México/ ITES de Teziutlán, México (e-mail: joseguillermo.cebada@live.itsteziutlan.edu).

P. Wiederhold, Depto. Control Automático, CINVESTAV Unidad Zacatenco, CDMX, México (e-mail: bien@ctrl.cinvestav.mx).

P. Sánchez-Sánchez, Facultad de Ciencias de la Electrónica, BUAP, Puebla, México (e-mail: pablo.sanchez@correo.buap.mx). (**Corresponding autor**).

I. López-Cruz, Depto. Ingeniería Agrícola, UACh, Texcoco, México (e-mail: ilopez@correo.chapingo.mx).

Coriolis y fuerza centrípeta, $g(q) \in \mathbb{R}^n$ es el par gravitacional, y $\tau \in \mathbb{R}^n$ es el par aplicado.

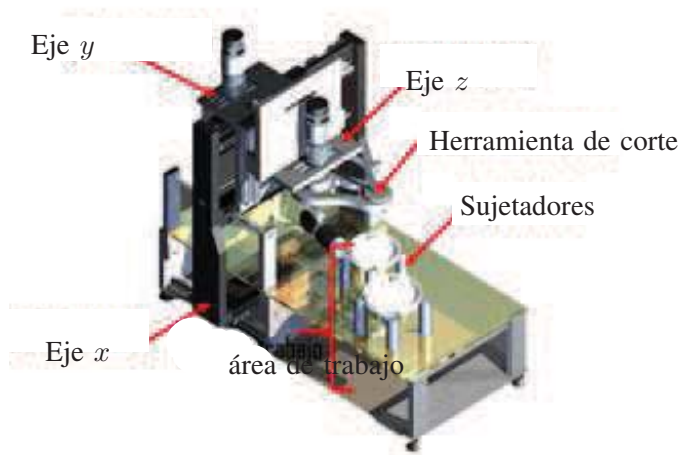


Fig. 1. Robot cartesiano 3 g.d.l (Plataforma experimental).

Por simplicidad se considera que todos los robots tienen articulaciones rotacionales, lo que permite enunciar las siguientes propiedades:

Propiedad 1. [14], [15] $H_i(q_i)$ *satisface*

$$\lambda_{h_i} \|x_i\|^2 \leq x_i^T H_i(q_i) x_i \leq \lambda_{H_i} \|x_i\|^2 \quad \forall q_i, x_i \in \mathbb{R}^n \quad (2)$$

donde $\lambda_{h_i} \triangleq \min_{\forall q_i \in \mathbb{R}^n} \lambda_{\min_i}(H_i(q_i))$, $\lambda_{H_i} \triangleq \max_{\forall q_i \in \mathbb{R}^n} \lambda_{\max_i}(H_i(q_i))$, y $0 < \lambda_{h_i} \leq \lambda_{H_i} < \infty$. ■

Propiedad 2. [14], [15] *Con la apropiada definición de $C_i(q_i, \dot{q}_i)$, la matriz*

$$\left[\dot{H}_i(q_i) - 2C_i(q_i, \dot{q}_i) \right] \quad (3)$$

es antisimétrica. ■

El efector final del robot manipulador interactúa con un entorno rígido, por lo que su movimiento está restringido a un espacio $(n - m)$ dimensional φ definido por $\varphi(q) = \mathbf{0}$ donde la función $\varphi: \mathbb{R}^n \rightarrow \mathbb{R}^m$ es continuamente diferenciable tal que $\varphi \in \mathcal{C}^2$. Debido a la existencia de la función φ se supone que existe una región operativa $\Omega \subset \mathbb{R}^n$ definida como $\Omega = \Omega_1 \times \Omega_2$, donde Ω_1 es un subconjunto convexo de \mathbb{R}^{n-m} , Ω_2 es un subconjunto abierto de \mathbb{R}^m . Debido a la región operativa $\Omega \subset \mathbb{R}^n$ se asume la existencia de una función $k_1: \Omega_1 \rightarrow \mathbb{R}^m$ que es continuamente diferenciable tal que $k_1 \in \mathcal{C}^2$, donde $\varphi(q^1, k(q^1)) = \mathbf{0} \quad \forall q^1 \in \Omega_1$. Bajo estas suposiciones, q^2 puede definirse de manera única por q^1 de modo que $q^2 = k(q^1) \quad \forall q^1 \in \Omega_1$. Bajo este supuesto, la matriz Jacobiana $J(q)$ existe y es no singular $\forall q \in \Omega_1$.

En este artículo se usa un robot cartesiano de 3 grados de libertad, Figura 1, cuyo modelo dinámico se define como

$$\underbrace{\begin{bmatrix} m_1 + m_2 + m_3 \\ m_1 + m_2 \\ m_1 \end{bmatrix}}_{H(q)} \underbrace{\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix}}_{\ddot{q}} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ m_3 \end{bmatrix}}_{g(q)} = \underbrace{\begin{bmatrix} \tau_{m1} \\ \tau_{m2} \\ \tau_{m3} \end{bmatrix}}_{\tau_m} \quad (4)$$

donde m_1, m_2, m_3 son las masas de los eslabones del robot cartesiano. Es importante resaltar que los robots cartesianos no poseen matriz de Coriolis y fuerza centrípeta [5], [16].

B. Modelo de Visión

El objetivo de un sistema de visión es crear un modelo del mundo real a partir de imágenes al recuperar la información útil de una escena usando sus proyecciones bidimensionales. Esta recuperación requiere la inversión de muchos datos en un plano. Considérese un sistema coordinado en espacio cartesiano $\Sigma_R = \{R_1, R_2, R_3\}$ ubicado en la base del robot donde los ejes R_1 y R_2 representan el área de trabajo del robot. Además, se cuenta con un sistema de referencia $\Sigma_E = \{E_1, E_2, E_3\}$ ubicado en el extremo final del robot cuyo origen es determinado por la cinemática directa x_R la cual es una función vectorial que describe la relación entre la posición articular de q con las coordenadas cartesianas de $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ donde n es el número de grados de libertad y m representa la dimensión del sistema de referencia de las coordenadas cartesianas. La posición $x_R \in \mathbb{R}^3$ del efector final del robot con respecto al sistema de coordenadas se da en términos de las posiciones articulares es decir $x_R = f(q)$. La cámara-CCD tiene asociado un sistema de referencia el cual se denota por $\Sigma_C = \{C_1, C_2, C_3\}$ cuyo origen se localiza en la intersección formada por el eje óptico y el centro de la lente Σ_C . Las coordenadas de un punto con respecto a este sistema de referencia se expresa como x_C . La localización relativa entre los sistemas de referencia del robot Σ_R y el sistema de la cámara Σ_C está representada por el vector $o_C = [o_{C1}, o_{C2}, o_{C3}]^T$. El objetivo (*target*) tiene un punto en el sistema cartesiano $\Sigma_T = \{T_1, T_2, T_3\}$, cuyo origen se referencia con respecto a su centro geométrico. La posición del marco del objeto con respecto a Σ_R se denota por el $o_T = [o_{T1}, o_{T2}, o_{T3}]^T$ [17], [18].

La escena adquirida por la cámara se proyecta en el CCD, la cual tiene asociado un sistema de referencia descrito por $\Sigma_I = \{I_1, I_2\}$, cuyo origen se encuentra en el centro geométrico del CCD. Los ejes I_1 e I_2 son paralelos y apuntan en la misma dirección de los ejes C_1 y C_2 respectivamente. Para obtener las coordenadas de la imagen en el plano del CCD es necesario obtener una transformación de perspectiva. La imagen de la escena sobre el CCD es digitalizada y transferida hacia la pantalla de la computadora, y se define un nuevo sistema coordinado bidimensional $\Sigma_D = \{u, v\}$, cuyo origen se ubica en la esquina superior izquierda del monitor [17], [18]. Por consiguiente, el sistema de visión completo, para la configuración cámara fija, expresa las coordenadas de la

imagen en píxeles. De esta forma, el modelo del sistema de visión es descrito por:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{\lambda}{\lambda + x_{C3}} \begin{bmatrix} \alpha_u & 0 \\ 0 & -\alpha_v \end{bmatrix} \begin{bmatrix} x_{C1} \\ x_{C2} \end{bmatrix} \quad (5)$$

$$\begin{bmatrix} x_{C1} \\ x_{C2} \\ x_{C3} \end{bmatrix} = \mathbf{R}^T(\theta) [x_R - \mathbf{o}_R^T] \quad (6)$$

donde $\alpha_u > 0, \alpha_v > 0$ son los factores de escalamiento en pixels/m, $\lambda > 0$ es la distancia focal de la cámara y $\frac{\lambda}{\lambda + x_{C3}}$ [19]. La Figura 2 muestra el diagrama a bloques del sistema visual usado: el espacio de trabajo no cambia, el bloque de control depende del bloque de extracción [18], [19]. En esta configuración, la adquisición de imágenes posee la prioridad como evento, para proporcionarles las coordenadas de arriba para el efector final del robot de la ecuación:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_c + r \sin(\omega t) \\ y_c + r \cos(\omega t) \end{bmatrix} \quad (7)$$

donde x_c y $y_c \in \mathbb{R}$ representan las coordenadas del centro del círculo, $r \in \mathbb{R}$ es el radio del círculo, $\omega \in \mathbb{R}$ es el período en el que se realizará el trazado del círculo.

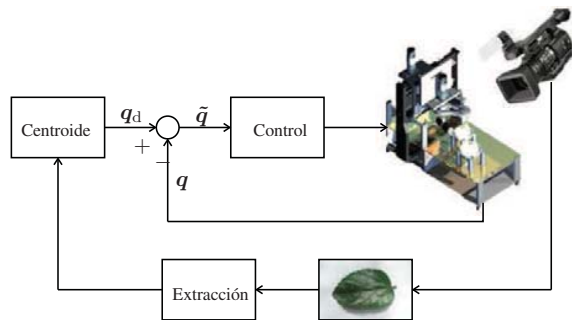


Fig. 2. Lazo de control con retroalimentación visual. q_d es la posición deseada, q es la posición real, $\tilde{q} = q - q_d$ es el error de posición.

Se usó información obtenida de un sensor de visión para controlar el movimiento del robot [19]. Para la detección de hojas se montó una cámara Logitech C270 a 0.4 m de distancia y de manera perpendicular de la escena, vea Figura 3. Es importante tener en cuenta que una cámara digital forma una representación discreta del plano del objeto que incide sobre un arreglo de fotosensores. El número de fotosensores en el arreglo CCD no coincide con el número de elementos en la imagen digital debido al procesamiento analógico y la digitalización. Las coordenadas del centro del plano imagen tampoco concuerdan con las coordenadas del centro de la imagen digital [18], [19]. En el sistema, Figura 2, la cámara se encuentra perpendicular a la escena que se está analizando, por lo tanto, la siguiente ecuación relaciona los píxeles (u, v) con las coordenadas reales de escena (x_m, y_m) :

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{f\alpha}{z} \begin{bmatrix} x_m \\ y_m \end{bmatrix} \quad (8)$$

donde f es la distancia focal, z es la distancia de la cámara a la escena y α es el factor de escala.

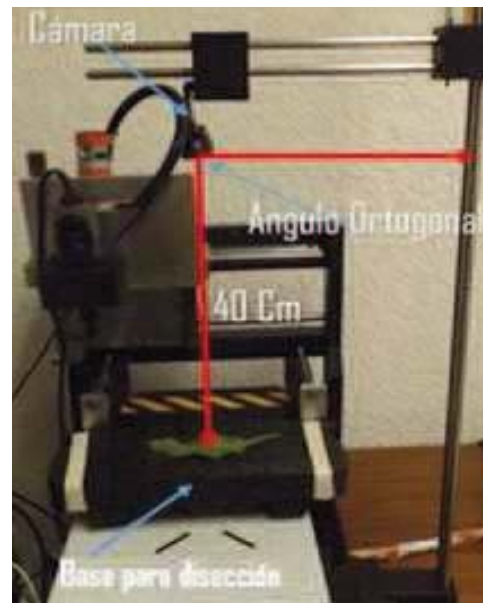


Fig. 3. Sistema cámara-robot-objetivo.

Se usó para la detección de las hojas una cámara Logitech C270, cuyas características son descritas en la Tabla I.

TABLA I
CARACTERÍSTICAS DE LA CÁMARA LOGITECH C270

Característica	Valor
Distancia Focal	4 mm
Tamaño de píxel	2.8 x 2.8 μm
Resolución de imagen	640 x 480 píxeles

Tomando en cuenta que nuestra cámara está a 0.4 m de la escena para que no colisione el robot con la estructura de cámara, se tiene lo siguiente:

$$\begin{bmatrix} u \\ v \end{bmatrix} = 3663 \begin{bmatrix} x_m \\ y_m \end{bmatrix} \quad (9)$$

Para detectar el objeto de interés, es decir, la hoja presente en la imagen, caracterizada por su color verde, y discriminar los objetos que no son de interés, se utilizó el algoritmo de Bhargav Anand [20], el cual consiste en la manipulación de los planos RGB de tal manera que:

$$W = R_{adj} + G_{adj} + B_{adj} \quad (10)$$

donde R_{adj}, G_{adj} y B_{adj} son los factores de atenuación para los planos rojo, verde y azul; estos valores son asignados por el usuario y varían en $[0.1, 1]$. Esto es importante ya que las hojas tienen la característica de estar cargados a rojos que representan verdes cálidos cuando las hojas son jóvenes o bien verdes fríos que están cargados al plano del azul cuando las hojas son maduras [4], [19].

El set-point requerido para el lazo de control, es hallado como el píxel más cercano al centroide de la hoja, la cual antes fue detectada y localizada en la imagen como la región de interés de color verde.

Si $u_{x,i}$, $v_{x,i}$ con $i = 1, \dots, m$ denotan las coordenadas de los píxeles de la hoja, entonces su centroide (u, v) , también llamado centro de masas o punto de centro gravitacional, es dado por

$$u = \frac{1}{m} \sum_{i=1}^m u_{x,i} \quad (11)$$

$$v = \frac{1}{m} \sum_{i=1}^m v_{x,i} \quad (12)$$

C. Control

La estructura de control usada para realizar la trayectoria definida en (7), está dada por

$$\tau = \mathbf{K}_p \tanh(\tilde{\mathbf{q}}) + \mathbf{K}_v \tanh(\dot{\tilde{\mathbf{q}}}) + \tau_m \quad (13)$$

donde $(\tilde{\cdot}) \in \mathbb{R}^n$ representa el error de (\cdot) , $\tau_m \in \mathbb{R}^n$ es el modelo dinámico del sistema, $\mathbf{K}_p \in \mathbb{R}^{n \times n}$ es la matriz de ganancia proporcional y $\mathbf{K}_v \in \mathbb{R}^{n \times n}$ es la matriz de ganancias derivativa, ambas son simétricas y diagonales [5], [13].

D. Sintonización

La sintonización de ganancias y parámetros en las estructuras de control para robots manipuladores, es un procedimiento empírico que halla los valores numéricos a prueba y error. Sin embargo, usando algoritmos evolutivos, se puede generar un vector de entrada V_{iG} que engloba los parámetros del modelo dinámico del robot, así como los límites máximos y mínimos del torque del motor, para entregar un vector de salida U_{iG} que representa los parámetros ajustados para ser implementados a nivel programación en el robot. Los parámetros que se evolucionan son \mathbf{K}_p y \mathbf{K}_v son parámetros que dependen de las características de torque del motor ya que sus unidades son [Nm] y [Nm/s] [5]. Los algoritmos de evolución diferencial son una clase de algoritmos evolutivos [21], [22] planteados como métodos globales de optimización [7]–[9], [20]. Los parámetros fijados por el usuario son: el parámetro de mutación F , el tamaño de población NP , el factor de cruzamiento CR , y el número de las generaciones asignadas G .

1) *Sintonización clásica*: Para sintonizar las ganancias \mathbf{K}_p y \mathbf{K}_v descritas en (2), se consideró cierto porcentaje del torque máximo del motor [5], [13], como sigue:

$$\mathbf{K}_p \leq 80\% \tau_{\max} \quad (14)$$

$$\mathbf{K}_v \leq 25\% \mathbf{K}_p \quad (15)$$

Esta forma de sintonización permite fijar los límites “superior” e “inferior” para el algoritmo de evolución diferencial.

2) *Descripción del Algoritmo JADE*: El algoritmo de Evolución Diferencial adaptable (*JADE*) tiene como característica que los parámetros: factor de mutación (F) y probabilidad de cruzamiento (CR) no son constantes, sino que su valor cambia en cada generación [8], [9].

El algoritmo evolución diferencial adaptativo *JADE* puede usar $DE|rand - to - best|1|bin$ con el operador de cruzamiento

$$\vec{v}_i = \vec{u}_{r1} + F_i(\vec{u}_{best} - \vec{u}_{r1}) + F_i(\vec{u}_{r2} - \vec{u}_{r3}) \quad (16)$$

o bien $DE|current - to - best|1|bin$ el cual tiene el operador de cruzamiento

$$\vec{v}_i = \vec{u}_i + F_i(\vec{u}_{best} - \vec{u}_i) + F_i(\vec{u}_{r1} - \vec{u}_{r2}) \quad (17)$$

donde \vec{u}_i es el vector objetivo y F_i es el factor de mutación asociado con \vec{u}_i , el cual cambia en cada generación.

Una modificación mayor [8] ocurre cuando se usa un archivo externo (A) para guardar soluciones inferiores a las de la población actual (P). Entonces el operador de mutación para el algoritmo $DE|rand - to - best|1|bin$ es:

$$\vec{v}_i = \vec{u}_{r1} + F_i(\vec{u}_{best} - \vec{u}_{r1}) + F_i(\vec{u}_{r2} - \vec{u}_{r3}) \quad (18)$$

donde \vec{u}_{r3} es un vector diferente de \vec{u}_{r1} y de \vec{u}_{r2} , elegido aleatoriamente de la unión de A y P . Y el operador de mutación para el algoritmo $DE|current - to - best|1|bin$ es:

$$\vec{v}_i = \vec{u}_i + F_i(\vec{u}_{best} - \vec{u}_i) + F_i(\vec{u}_{r1} - \vec{u}_{r2}) \quad (19)$$

donde \vec{u}_{r2} es un vector diferente de \vec{u}_i y de \vec{u}_{r1} , seleccionado aleatoriamente de la unión de A y P . La adaptación del factor de mutación (F_i) se lleva a cabo mediante la siguiente ecuación

$$F_i = randc_i(\mu_F, 0.1) \quad (20)$$

donde S_F es el conjunto de todos los factores de mutación exitosos de F_i ; $randc_i(\mu_F, 0.1)$ significa una distribución Cauchy con el parámetro de localización μ_F y el parámetro de escala 0.1. Si $F_i \geq 1$ este es truncado en 1 y si sucede que $F_i \leq 0$, este es regenerado. El parámetro μ_F es inicializado con 0.5 y al final de cada generación es actualizado mediante

$$\mu_F = (1 - c)\mu_F + c \times mean_L(S_F) \quad (21)$$

donde $c \in [0, 1]$ y $mean_L(S_F)$ es la media de Lehmer definida como:

$$mean_L(S_F) = \frac{\sum_{F \in S_F} F^2}{\sum_{F \in S_F} F} \quad (22)$$

La adaptación de la probabilidad de cruzamiento (CR_i) se lleva a cabo para cada individuo mediante una distribución normal con media μ_{CR} y desviación estándar 0.1.

$$CR_i = randn_i(\mu_{CR}, 0.1) \quad (23)$$

A continuación se presenta el pseudo código del algoritmo de evaluación diferencial.

- 1 *Generar una población ($P(0)$) de soluciones*
- 2 *Evaluar cada solución $g = 1$*
- 3 *while (convergencia no se alcance)*

4 **for** $i = 1$ to NP

 Aplicar el operador de mutación diferencial
 Aplicar el operador de cruzamiento diferencial
 Evaluar la nueva solución usando la función objetivo
 Aplicar el operador de selección diferencial

5 **end**

$g = g + 1$

6 **end**

Los elementos mas importantes del programa se encuentra en el apéndice A, y la descripción de cada operador en los apéndices B, C y D.

E. Procedimiento de Sintonización con JADE

Los robots manipuladores son sistemas mecánicos muy complejos, debido a la naturaleza no lineal de su dinámica. Por esta razón, en la comunidad de robótica no existen criterios bien establecidos para una evaluación adecuada de las estructuras de control. Sin embargo, en la práctica se acepta el comparar el rendimiento de los controladores mediante el uso de la norma \mathcal{L}_2 como medida numérica objetiva para un comparar la curva de error. La norma \mathcal{L}_2 se define como

$$\mathcal{L}_2 = \sqrt{\frac{1}{T} \int_{t_0}^{t_1} \|\tilde{\mathbf{q}}(t)\|^2 dt}. \quad (24)$$

donde T es el período de muestreo, $\tilde{\mathbf{q}} \in \mathbb{R}^n$ es el error de posición en coordenadas articulares, t_0 y t_1 son los tiempos inicial y final, respectivamente.

La norma \mathcal{L}_2 promedia los errores de posición bajo la acción de diferentes sintonizaciones para ello **JADE** toma la a norma \mathcal{L}_2 como **función objetivo** para calcular las ganancias K_p y K_v del controlador descrito en (13) el cual es evaluado en el modelo dinámico del robot manipulador (4), tal que la función objetivo tienda a cero a un punto fijo. El tiempo estimado que realiza este proceso es 3604 segundos. Estas soluciones de ganancias que satisfagan la función objetivo genera un vector resultante de ganancias, sin embargo, como este proceso es iterativo dado la naturaleza del algoritmo **JADE**, si la aptitud del vector resultante o población mutada es mejor que el miembro de la población elegida entonces el nuevo vector reemplaza al vector con el cual fue comparado y se guarda en el archivo, de lo contrario mantiene la población elegida [10], [21], [22]. Para el ajuste de parámetros con este algoritmo, se considera un set–point en parámetro de mutación a $F = 0.7$ y el parámetro de cruzamiento $CR = 0.7$ que se auto–ajustaran en cada iteración [8], [9]. El número de población es $NP = 18$, el número de las generaciones asignadas es $G = 150$. Este algoritmo se usó para encontrar las ganancias K_p y K_v cuyo límite superior de ganancias se determinó a partir del criterio clásico de sintonización, mientras que, el límite inferior se definió al 1% de lo fijado en (14) y (15).

III. PLATAFORMA EXPERIMENTAL

En la Figura 1 se muestra un robot cartesiano de tres grados de libertad construido de aluminio y naylamid con eslabones de corte prismático. El robot usa tres motores de

corriente directa de 24 V a 2.1 A, tiene una velocidad de 0–122 rpm, una precisión de 0.367° y un torque máximo de 1.568 Nm. Además, el sistema posee encoders incrementales de 980 pulsos por revolución [23]. Como herramienta de corte, se implementó un cortador vertical (End Mills) de 1/32 de pulgada, el cual permite obtener secciones de hoja de 0.02 m^2 , unida a una superficie adherente que sujeta la hoja mientras el robot ejecuta la trayectoria de corte. Las características mecánicas del robot de 3 grados de libertad se presentan en la Tabla II

TABLA II
CARACTERÍSTICAS MECÁNICAS

Eslabón	Peso N	Tamaño m	Masa kg
1	3.80	0.40×0.07	0.387
2	2.70	0.41×0.39	0.275
3	2.35	0.21×0.21	0.239

El sistema de visión, consiste en una cámara fija **Logitech C270**[®] colocada en un soporte a 90° de la escena. Este dispositivo tiene la misión de detectar hojas y asignar el punto inicial (punto de partida) deseado de tal manera que el algoritmo de control descrito en (2), tenga un set–point para ejecutar una trayectoria circular, es decir, estimar un torque tal que las posiciones y velocidades asociadas a las articulaciones del robot sigan con exactitud a las posiciones deseadas para que el error de posición tienda a cero conforme el tiempo tiende a infinito [5].

IV. RESULTADOS EXPERIMENTALES

Para efectos de este artículo se ha desarrollado el robot cartesiano de tres grados de libertad descrito en [23] con retroalimentación visual (*visual servoing*). El objetivo del presente trabajo consiste en la evaluación de dos métodos de sintonización de ganancias aplicados a un controlador de trayectorias PD+ con acciones acotadas [5], [13], [23] para la detección y disección de hojas de plantas, con las metas de minimizar el error de arribo de la herramienta de corte y de realizar trayectorias circulares de corte cerradas. El ajuste se realiza fijando un punto de prueba como lo reporta [23]. Usando una computadora Toshiba AMD8 de 2GHz, con 12 Gb de RAM, se programó y ejecutó el algoritmo JADE para minimizar \bar{v}_i repitiendo la búsqueda 10 veces hasta obtener el promedio de los parámetros ajustados. Las ganancias obtenidas a través de los métodos JADE y Clásico descritos en el apartado IV de sintonía son presentadas en la Tabla III.

TABLA III
GANANCIAS OBTENIDAS PARA EL CONTROLADOR PD+

Ganancia	Clásica	JADE
K_{p1}	1.254 [N]	0.788 ± 0.0607 [N]
K_{p2}	1.254 [N]	0.946 ± 0.2116 [N]
K_{p3}	1.254 [N]	0.825 ± 0.2459 [N]
K_{v1}	0.313 [N]	0.308 ± 0.0016 [N/s]
K_{v2}	0.313 [N]	0.281 ± 0.0385 [N/s]
K_{v3}	0.313 [N]	0.192 ± 0.0749 [N/s]
Tiempo estimado	—	3604 [s]

La Figura 4a muestra el proceso del lazo de visión descrito en la Figura 2, donde toma el valor $u = 384.72$ y $v = 341.95$, siendo las coordenadas del set-point en píxeles en la imagen. Por el modelo de visión (5), las correspondientes coordenadas reales de ese punto son $x_m = 0.08m$ y $y_m = 0.1m$, definiendo al punto de partida para el trazo de la trayectoria cerrada descrita en (1) para la herramienta de corte, tal y como se observa en la Figura 4b). En la Figura 5, se aprecia el comportamiento del robot al ejecutar la trayectoria cerrada, en azul se representa la acción de control con la sintonía JADE; se observa que los sobre-impulsos son menores a lo largo de la trayectoria, que la acción de control sintonizada por el método clásico descrita en color negro. El error de posición ponderado de los tres eslabones fue $\tilde{q} = 0.001$ m mientras que en la sintonización clásica el error ponderado de los tres eslabones fue $e = 0.005$ m. Finalmente, la sintonización permitió obtener secciones circulares de 0.02 m² y 0.01 m² de área, de las hojas, sin que se desgarraran con el paso del cortador vertical, como se muestra en la Figura 6.

V. CONCLUSION

Los resultados de los experimentos demuestran que la sintonización a través del método JADE mejora el comportamiento del controlador disminuyendo las perturbaciones que se presentan a lo largo de la acción de control.

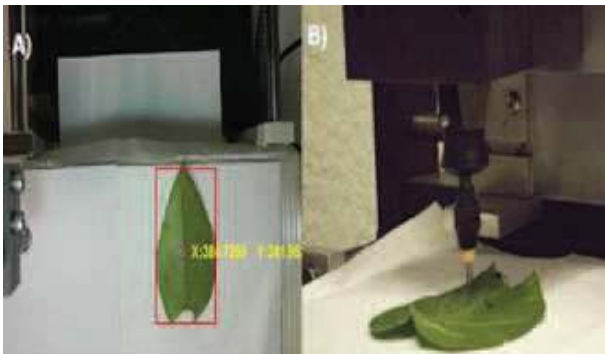


Fig. 4. A) Punto de arriba a través de visual servoing, B) Cortador vertical arribando al set-point de la trayectoria circular.

Otro punto importante es la verificación del funcionamiento del lazo de control por visión donde los píxeles de la imagen son transformados a coordenadas reales para que el controlador ejecute primero el punto de arriba y un instante de tiempo después ejecute la trayectoria cerrada. Se apreció que el error ponderado de los tres eslabones es de $\tilde{q} = 0.001$ m lo que garantiza ejecutar de manera suave la trayectoria circular evitando desgarrar en la hoja y así obtener secciones de hojas enteras.

APÉNDICE A PROGRAMA JADE

```
1 % Inicializa poblacion
2 Pop = zeros(NP, n+2);
3 PopNew = zeros(NP, n+2);
```

```
4 ObjV = zeros(NP, 1);
5 for i = 1:NP
6 Pop(i, 1:n) = xmin + rand(1, n).*(xmax-xmin
7 );
8 Pop(i, n+1) = F;
9 Pop(i, n+2) = CR;
10 end
11 % Evalua poblacion inicial
12 for i = 1:NP
13 ObjV(i) = feval(fname, Pop(i, 1:n));
14 end
15 [BestValue, BestIndex] = min(ObjV);
16 [WorstValue, WorstIndex] = max(ObjV);
17 BestIndividual = Pop(BestIndex, :);
18 WorstIndividual = Pop(WorstIndex, :);
19
20 % Genera el vector de pruebas de acuerdo
    a la estrategia seleccionada
```

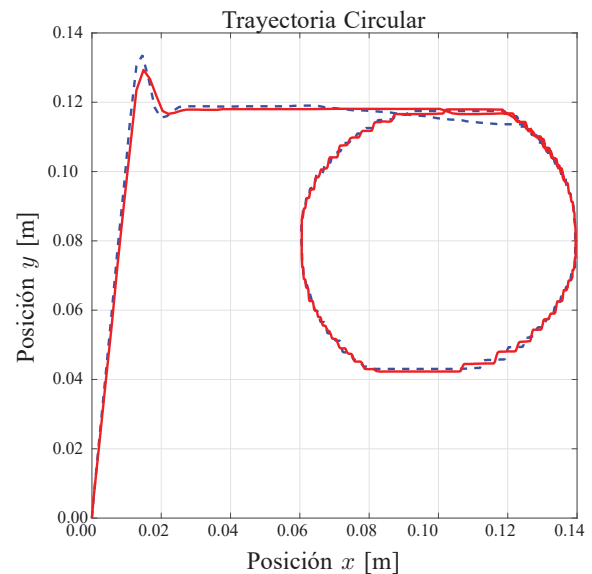


Fig. 5. Trayectoria debido al Método Clásico (---) y Trayectoria debido al Método JDE (—).



Fig. 6. Sección de hoja obtenida al sintonizar el controlador PD+.

```

21 Trial = feval(StrategyName, Pop, CR, F, n      cruzamiento
    , a, b, c, d, e, i, BestIndex, Kp,
    xmin, xmax, NP, Fl, Fu, Tau1, Tau2);
22
23 % Calcula la funcion de costos
24 [BestValue, BestIndex] = min(ObjV);
25 if BestValue <= VTR
26 return
27 end
28 [~, WorstIndex] = max(ObjV);
29 BestIndividual = Pop(BestIndex, :);
30 WorstIndividual = Pop(WorstIndex, :);
31
32
33 % Estrategia DE/rand/2/bin
34 function Trial = Rand2bin(Pop, varargin)
35 j = floor(rand*NVAR);
36 jj = j+1;
37 for k = 1:NVAR
38 if ((rand < CR) || (k==NVAR))
39 Trial(jj) = Pop(e, jj)+F*(Pop(a, jj)+Pop(b,
    jj)-Pop(c, jj)-Pop(d, jj));
40 else
41 Trial(jj) = Pop(i, jj);
42 end
43 end
44
45 % DE/best/2/bin
46 function Trial = Best2bin(Pop, varargin)
47 j = floor(rand*NVAR);
48 jj = j + 1;
49 for k = 1:NVAR
50 if ((rand < CR) || (k==NVAR))
51 Trial(jj) = Pop(BestIndividual, jj)+F*(Pop(
    a, jj)+Pop(b, jj)-Pop(c, jj)-Pop(d, jj));
52 else
53 Trial(jj) = Pop(i, jj);
54 end
55 end
56
57 % Funcion objetivo a traves -best/2/bin
58 function Trial=RandtoBestbin(Pop,
    varargin)
59 j = floor(rand*NVAR);
60 jj = j+1;
61 for k = 1:NVAR
62 if ((rand < CR) || (k==NVAR))
63 Trial(jj)=Pop(i, jj)+ F*(Pop(
    BestIndividual, jj) + Pop(a, jj) - Pop(
    i, jj) - Pop(b, jj));
64 else
65 Trial(jj) = Pop(i, jj);
66 end
67
68 % Estrategia DE/best/1/bin
69 function Trial = Best1bin(Pop, varargin)
70
71 % Aleatorizacion del factor de
    
```

```

72 j = floor(rand*NVAR);
73 jj = j+1;
74 for k = 1:NVAR
75 if ((rand < CR) || (k==NVAR))
76 Trial(jj)=Pop(BestIndividual, jj) + F*(
    Pop(a, jj) - Pop(b, jj));
77 else
78 Trial(jj) = Pop(i, jj);
79 end
80 end
81
82 % Funcion objetivo a traves de DE/rand
    //bin
83 function Trial = Rand1bin(Pop, varargin)
84 Trial = zeros(1, NVAR + 2);
85
86
87 % Funcion de adaptacion factor de
    mutacion
88 if (rand<Tau1)
89 F = Fl + rand*Fu;
90 else
91 F = Pop(i, NVAR + 1);
92 end
93 if (rand<Tau2)
94 CR = rand;
95 else
96 CR = Pop(i, NVAR + 2);
97 end
98
99 % Aplica el factor de mutacion
    Aleatorizado
100 j = floor(rand*NVAR);
101 jj = j + 1;
102 for k = 1:NVAR
103 if ((rand < CR) || (k==NVAR))
104 Trial(jj)=Pop(i, jj) + Kp*(Pop(c, jj) -
    Pop(i, jj)) + F*(Pop(a, jj) - Pop(b,
    jj));
105 else
106 Trial(jj)=Pop(i, jj);
107 end
    
```

APÉNDICE B OPERADOR DE MUTACIÓN

El operador de mutación genera NP nuevos vectores (\vec{v}) de acuerdo a la ecuación:

$$\vec{v}_i = \vec{u}_{r_1} + F(\vec{u}_{r_2} - \vec{u}_{r_3}) \quad (25)$$

donde $r_1, r_2, r_3 \in [1, 2, \dots, NP]$ son los índices mutuamente diferentes para que exista mutación y también diferentes del índice i para que (25) no se transforme en un operador de cruzamiento.

Esto se garantiza haciendo un muestreo de tres vectores en forma aleatoria, a partir del número de individuos (NP) en la población y verificando que se satisfagan las condiciones

$r_1 \neq i, r_2 \neq r_1, r_2 \neq i, r_2 \neq i, r_3 \neq r_1, r_3 \neq r_2$. Además, F es una constante escalar que afecta la variación diferencial entre dos vectores.

APÉNDICE C

OPERADOR DE CRUZAMIENTO

El cruzamiento consiste en combinar el vector $\vec{v}_i = [v_{i1}, v_{i2}, \dots, v_{id}]$, mutado previamente, con otro vector llamado el vector blanco $\vec{u}_i = [u_{i1}, u_{i2}, \dots, u_{id}]$, el cual puede ser considerado como un individuo padre de la generación anterior. Se genera así un nuevo vector descendiente $\vec{u}'_i = [u'_{i1}, u'_{i2}, \dots, u'_{id}]$, también llamado vector ensayo. El cruzamiento binomial se implementa mediante la siguiente ecuación:

$$u'_{ij} = \begin{cases} v_{ij} & \text{si } (\text{randb}(j) \leq CR) \text{ o } j = \text{rnbr}(i) \\ u_{ij} & \text{si } (\text{randb}(j) > CR) \text{ o } j \neq \text{rnbr}(i) \end{cases} \quad (26)$$

para $i = 1, 2, \dots, NP$ y $j = 1, 2, \dots, d$; donde $\text{randb}(j) \in [0, 1]$ es la j -ésima evaluación de un generador de números aleatorios uniforme; $\text{rnbr}(i) \in 1, 2, \dots, d$ es un índice generado en forma aleatoria. $CR \in [0, 1]$ representa la probabilidad de cruzamiento, la cual incrementa la diversidad de la población. Valores grandes de CR generan un vector hijo \vec{u}'_i más similar al vector mutado \vec{v}'_i y viceversa, por lo que, con un valor más cercano a la unidad, se incrementa la velocidad de convergencia del algoritmo. Aún con una probabilidad de cruzamiento igual a cero, el operador de cruzamiento asegura que el vector descendiente sea diferente al vector padre, al menos en un valor.

APÉNDICE D

OPERADOR DE SELECCIÓN

Es un operador muy simple ya que sólo consiste en comparar el valor de la función de costo $f(\vec{u}_i)$ del vector blanco \vec{u}_i contra el correspondiente valor de la función objetivo $f(\vec{u}'_i)$ del vector ensayo \vec{u}'_i . El vector que tenga el menor valor de la función de costo pasa a ser miembro de la población en la siguiente generación.

AGRADECIMIENTOS

Este trabajo fue financiado por la dirección de Investigación y Posgrado de la Universidad Autónoma Chapingo y el Consejo Nacional de Ciencia Y Tecnología (CONACyT) a través del Programa de Beca Mixta para movilidad Académica, para la realización de pruebas y validación en el Departamento de Control Automático del CINVESTAV Unidad Zacatenco.

REFERENCIAS

- [1] C. Santos-Briones, and T.S.M. Hernández-Sotomayor, "Coffee biotechnology", *Brazilian Journal of Plant Physiology*, vol. 18, no. 1, pp. 217–227, March 2006.
- [2] P. López-Gómez, L. Iracheta-Donjuan, M. Castellanos-Juárez, I. Méndez-López, A. Sandoval-Esquivel, J. F. Aguirre-Medina, M. C. Ojeda-Zacarias and Gutiérrez-Díez A., "Influence of explant and culture medium on somatic embryogenesis of coffee leaves", *Rev. Fitotecnia Mexicana*, vol. 33, no. 3, pp. 205–213, Febr. 2010.

- [3] C. Wunter-Bac and E. J. van-Enten, "Harvesting robots for high-value crops: state of the art review and challenges ahead", *Journal Field Robotics*, vol. 31, no. 6, pp. 888–914, April 2014.
- [4] N. Kondo and K. C. Ting, "Robotics for plant production", *Artificial Intelligence Review*, vol. 12, pp. 227–243, June 1998.
- [5] F. Reyes, *Robótica: Control de Robots Manipuladores*, Edit. Alfaomega, 2011, pp. 251–470.
- [6] K. V. Price, *An introduction to differential evolution.*, in Edit. New Ideas in Optimization, D. Corne, M. Dorigo, F. Glover (Eds), McGraw-Hill, England, pp:79–108.
- [7] K. V. Price, R. M. Storn, J. A. Lampinen, *Differential Evolution. A Practical Approach to Global Optimization*, Berlin-Heidelberg, German, Springer, (2005).
- [8] J. Zhang and A. C. Sanderson (2009). *JADE: adaptive differential evolution with optional external archive*. *IEEE Transactions on Evolutionary Computation* 13 (5): 945-958.
- [9] A. Marín, J. A. Hernández and J. A. Jiménez, "Tuning multivariable optimal PID controller for a continuous stirred tank reactor using an evolutionary algorithm", *IEEE Latin America Transactions*, vol. 16, no. 2, pp. 422–427, Feb. 2018.
- [10] I. M. Willjuice and S. Baskar, "Evolutionary algorithms based design of multivariable PID controller", *Expert Systems with Applications*, vol. 36, no. 5, pp. 9159–9167, July 2009.
- [11] M. Saad, J. Hishamuddin and Z. M. D. Intan, "PID controller tuning using evolutionary algorithms", *WSEAS Trans. on Systems and Control*, vol. 7, no. 4, pp. 139–149, Oct. 2012.
- [12] M. J. Neath, A. K. Swain, U. K. Madawala and D. J. Thrimawithana, "An optimal PID controller for a bidirectional inductive power transfer system using multi-objective genetic algorithm", *IEEE Trans. on Power Electronics*, vol. 29, no. 3, pp. 1523–1531, May 2013.
- [13] B. Paden and R. Panja, "Globally asymptotically stable PD+ controller for robots manipulators", *International Journal of Control*, vol. 47, no. 6, pp. 1697–1712, June 1987.
- [14] J. Gudiño-Lau and M. A. Arteaga-Pérez, "Force control with a velocity observer", in *Proc. European Control Conference*, (Cambridge, UK), September 2003.
- [15] J. Gudiño-Lau, M. A. Arteaga-Pérez, L. A. Muñoz and V. Parra-Vega, "On the control of cooperative robots without velocity measurements", *IEEE Transactions on Control Systems Technology*, vol. 12, no. 4, pp. 600–608, 2004.
- [16] T. M. Khan, M. Arshad and M. A. Choudhry, "Modeling and control of cartesian robot manipulator", *IEEE 9th International Multitopic Conference INMIC*, Karachi, Pakistan, Dec. 24–25, 2005.
- [17] R. González, L. Hernández, E. Izaguirre and E. Rubio, "Estrategia de control para robots manipuladores con realimentación visual y plataforma electro-neumática de 3gdl", *Revista de Ingeniería Mecánica*, vol. 14, no. 3, pp. 245–257, Dec. 2011.
- [18] P. Corke, *Robotics, Vision, and Control – Fundamental Algorithms in MATLAB*, Edit. Springer, Heidelberg Berlin, pp. 59–67, 2011.
- [19] J. H. Sossa, *Visión Artificial*, Edit. Ra-Ma, Madrid, Spain, pp. 136–188, 2013.
- [20] A. Bhargav-Anand, "Tracking red color objects using Matlab", *Amrita School of Eng.*, Bangalore, 2010 available at <https://www.mathworks.com/matlabcentral/profile/authors/2310475-abhargav-anand> (accessed Jan. 2019).
- [21] J. Zhang and A. Sanderson, *Adaptive Differential Evolution*, Edit. Springer, Berlin Heidelberg, Germany, 2009.
- [22] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, Edit. Springer, Berlin Heidelberg, Germany, 2003.
- [23] J. G. Cebada, F. F. Hahn, A. Ruiz, E. Romantchik and A. Michua, "Design of a position control based on Cuckoo search tuning for a cutter leaves robot", *IEEE Latin America Transactions*, vol. 14, no. 5, pp. 2085–2092, May 2016.



Jose Guillermo Cebada-Reyes Nació en la Ciudad de Puebla, México, en 1983. Es Licenciado en ciencias de la Electrónica por parte de la Benemérita Universidad Autónoma de Puebla. Recibió su grado de Doctor en Ingeniería Agrícola por parte de la Universidad Autónoma Chapingo. Sus áreas de interés: robótica, automatización de procesos agrícolas, aplicación de la optimización metaheurística, electrónica aplicada en biotecnología vegetal, visual servoing.



Petra Wiederhold Profesor investigador de tiempo completo de Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, Unidad Zacatenco, Departamento de Control Automático. Recibió su grado como Doctor en Matemáticas por parte de la Universidad Autónoma Metropolitana, Unidad Iztapalapa. Sus áreas de interés son: modelado y procesamiento digital de imágenes, matemáticas discretas.



Pablo Sánchez-Sánchez Profesor investigador de tiempo completo en la Facultad de Ciencias de la Electrónica de la Benemérita Universidad Autónoma de Puebla. Recibió su grado como Doctor en Ingeniería Electrónica por parte de la Universidad Nacional Autónoma de México. Sus áreas de interés son: modelado de sistemas mecánicos, visión, control de fuerzas, robótica y diseño de simuladores.



Irineo López-Cruz Profesor investigador de tiempo completo en el Posgrado de Ingeniería Agrícola y Uso Integral del Agua, de la Universidad Autónoma Chapingo. Recibió su grado como Doctor en Ingeniería Agrícola por parte de la Universidad de Wageningen, Países Bajos. Sus áreas de interés son: modelado de sistemas biológicos, control óptimo y algoritmos metaheurísticos aplicados.