

# Desarrollo de un sistema de 4 barras para seguimiento de trayectoria

Unidad Profesional de Ingeniería Interdisciplinaria Campus Tlaxcala

Herrera Rodríguez R. A., Aldama Ramos J. A., Betancourt González J., Castelán Rosete M. A..

**Abstract**—En la presente práctica se pretende unificar el desarrollo de un mecanismo de cuatro barras formulado a partir de un algoritmo memético y que está condicionado bajo ciertas restricciones, de manera que se detallará el modelo teórico y físico para posteriormente ser programado y simulado y a través de este proyectar un sistema de aplicación práctico y funcional para cualquier usuario, donde los resultados obtenidos satisfagan las reglas de optimización.

**Index Terms**—Algoritmo, restricción, optimización, simulación, mecanismo.

## I. INTRODUCCIÓN

LA denominación genérica de 'Algoritmos meméticos' (MA) se utiliza para abarcar una amplia clase de metaheurísticas (es decir, métodos de propósito general destinados a guiar una heurística subyacente). El método está basado en una población de agentes y demostró ser de éxito práctico en una variedad de dominios de problemas y en particular para la solución aproximada de problemas de optimización. El desarrollo óptimo de sistemas industriales resulta ampliamente útil en el diseño de componentes mecánicos, de manera que los sistemas mecánicos optimizados, permiten que los parámetros dados por las condiciones de restricción establecidas permitan construir un sistema con el cual se obtengan valores óptimos. Los sistemas de cuatro barras son relevantes en cualquier sistema de accionamiento mecánico ya que pueden generar un movimiento complejo y preciso. Suele estar diseñado por experimentación. Sin embargo, al estar diseñados por este método se puede causar un error. En las últimas décadas, se ha encontrado que muchos investigadores intentan resolver la optimización de la generación de caminos para los sistemas de cuatro barras utilizando métodos metaheurísticos. El objetivo de este enfoque es determinar los valores óptimos de la longitud de los enlaces del mecanismo de cuatro barras (manivela, biela, balancín) con punto acoplador para seguimiento de trayectoria.

## II. DESARROLLO

### Requerimientos de usuario.

#### Requerimientos funcionales:

- Accion: Posicionamiento de robot para soldadura de piezas.
- Descripción: El propósito básico de un posicionado de piezas de trabajo es expandir el espacio de trabajo del robot. Esto los hace similares a otros ejes auxiliares en los que el propio robot está montado sobre un carril lineal, un carril curvo o una grúa pórico.

- Entrada: Las principales entradas, del posicionamiento de puntos para colocar, cada una de las piezas a soldar en un momento exacto sin ningún problema y en tiempo aproximado.
- Salida: Como algunas de las salidas encontramos, los diferentes puntos soldados, así como un mecanismo correcto, con el tiempo ajustado en la potencia para evitar un recorrido superior o menos de cada pieza.
- Restricciones o excepciones: Ley de Grashof: Es una consideración fundamental del diseño, ya que define los criterios para asegurar la movilidad completa de al menos un eslabón en un mecanismo de cuatro barras. Esta ley establece que, para un eslabón plano de cuatro barras, la suma de la barra más corta y la barra más grande no puede ser mayor que la suma de las barras restantes, si se desea una rotación relativa continua entre dos elementos.
- Secuencia de ángulos de entrada: La generación de una trayectoria sin sincronización prescrita requiere un orden ascendente o descendente de los valores de los ángulos del cigüeñal, en correspondencia con cada punto de precisión.

#### Requerimientos de usuario:

- El sistema tendrá una interfaz clara que nos ayude a comprender el funcionamiento de nuestro modelo.
- El sistema tendrá dos posibles variables cambiantes.
- El sistema tendrá la mayor seguridad para evitar el cambio de variables etc. Por cambio de variables.
- El sistema tendrá una representación de modelos clara, representada en dos cajas con una interfaz clara y agradable a al visto.
- No encontraremos ninguna falla en el sistema que indique, la variación de ejes en sentido contrario.
- El sistema indicara los puntos en el modelo.
- El sistema indicará las variables a asignar.
- El sistema tendrá un apartado de información con la explicación de los componentes del sistema, así como el objetivo del mismo.

#### Requerimientos del sistema:

- Cuando el usuario quiera ingresar a la vista del sistema, el sistema debe de tener esa diferencia entre líneas y que la interfaz ayude a comprender el funcionamiento del modelo.
- El usuario podrá modificar un total de dos variables, pero sin la posibilidad de hacer un cambio total del funcionamiento.
- Cuando el usuario quiera realizar un cambio de alguna variable, deberá haber ciertas restricciones de seguridad,

para no entorpecer el modelado, y volverlo un modelo sin fundamentos.

- Cada que el usuario habrá la interfaz encontrará un modelado con todas las variables exactas y con conocimiento de cada una de sus funcionalidades.
- El usuario no tendrá la opción de meter alguna variable negativa dentro del modelo que afecte el funcionamiento.
- El usuario podrá observar de forma correcta los puntos, los cuales tendrán como objetivo la soldadura en un sistema de barras.
- El usuario tendrá la seguridad de asignar variables en el sistema, que indica la parte de las variables como la forma en la que se indican cada una de ellas.
- La interfaz generada para el usuario será clara y tendrá todos los puntos de fuente de letra y colores para tener en claridad cada funcionamiento del modelado.

### *Historias de usuario.*

#### 1) Interfaz amigable.

Como trabajador quiero una interfaz fácil de usar para monitorear la sección en la línea de ensamblaje del robot soldador.

##### a) Validación:

- Comprobar que lo ingresado en la interfaz funciona perfectamente como entrada en nuestro algoritmo.
- Comprobar que las gráficas obtenidas como resultado y mostradas al usuario son correctas.
- Comprobar la funcionalidad de cada sección dentro de la interfaz

b) Valor: 100

c) Prioridad: 1

d) Duración: 8 hrs.

#### 2) Control de trayectoria.

Como gerente quiero poder programar el robot soldador para seguir una trayectoria específica.

##### a) Validación:

- Comprobar distintas trayectorias y buscar que funcionen adecuadamente.

b) Valor: 100

c) Prioridad: 1

d) Duración: 28 hrs.

#### 3) Soldadura precisa.

Como gerente de producción quiero tener la certeza de que se suelden los puntos correctos en la línea de ensamblaje para asegurar una excelente calidad en la soldadura.

##### a) Validación:

- Comprobar que se sueldan los puntos correctos.

b) Valor: 150

c) Prioridad: 1

d) Duración: 30 hrs.

#### 4) Ciclo de soldadura.

Como gerente de producción quiero tener la certeza de que el ciclo de soldadura concuerda con los tiempos

de la línea de ensamblaje para asegurar una línea de ensamblaje sin interrupciones.

##### a) Validación:

- Comprobar los tiempos de nuestra línea de soldadura.
- Comprobar la velocidad de avance de nuestra línea de ensamblaje.

b) Valor: 150

c) Prioridad: 1

d) Duración: 40 hrs.

#### 5) Alertas de falla.

Como gerente quiero recibir alertas cuando el robot soldador detecte una falla en el proceso de soldadura para actuar lo más pronto posible.

##### a) Validación:

- Comprobar el funcionamiento de mensajes de alerta.
- Buscar parámetros para mantener monitoreados.

b) Valor: 150

c) Prioridad: 1

d) Duración: 40 hrs.

#### 6) Configuración de tamaño.

Como gerente quiero configurar el robot soldador para que se ajuste al tamaño de los componentes de la línea de ensamblaje para ser preciso al momento de soldar.

##### a) Validación:

- Comprobar el funcionamiento al variar los puntos de soldadura.
- Buscar parámetros para mantener monitoreados en caso de objetos demasiado grandes o demasiado pequeños.

b) Valor: 150

c) Prioridad: 1

d) Duración: 50 hrs.

### *Diagrama de secuencia.*

Para poder hablar de nuestro mecanismo aplicado para un robot de soldadura implementado en una línea de ensamblaje es necesario conocer algunos componentes, es decir, el transportador para los paneles laterales, la plantilla para fijarlos, la cortina de oscurecimiento para proteger al trabajador del rayo láser, el cabezal óptico y la selladora para sellar los paneles son necesarios para la estación de soldadura láser.

Con estos componentes podemos modelar la interacción, consistente en un conjunto de componentes y sus relaciones mediante el uso del diagrama de secuencia. Como muestra la Figura 1, las relaciones entre los componentes pueden construirse de dos formas: haciendo hincapié en las actividades de los componentes y haciendo hincapié en los mensajes ordenados en tiempo creciente desde la parte superior del diagrama hasta la parte inferior que pueden enviarse entre ellos.

Gráficamente, la flecha que vuelve sobre sí misma teniendo función "Movimiento" en el componente 'Banda transportadora' significa que se desplaza de la coordinación inicial a la nueva durante el tiempo dado. La otra flecha conectada con

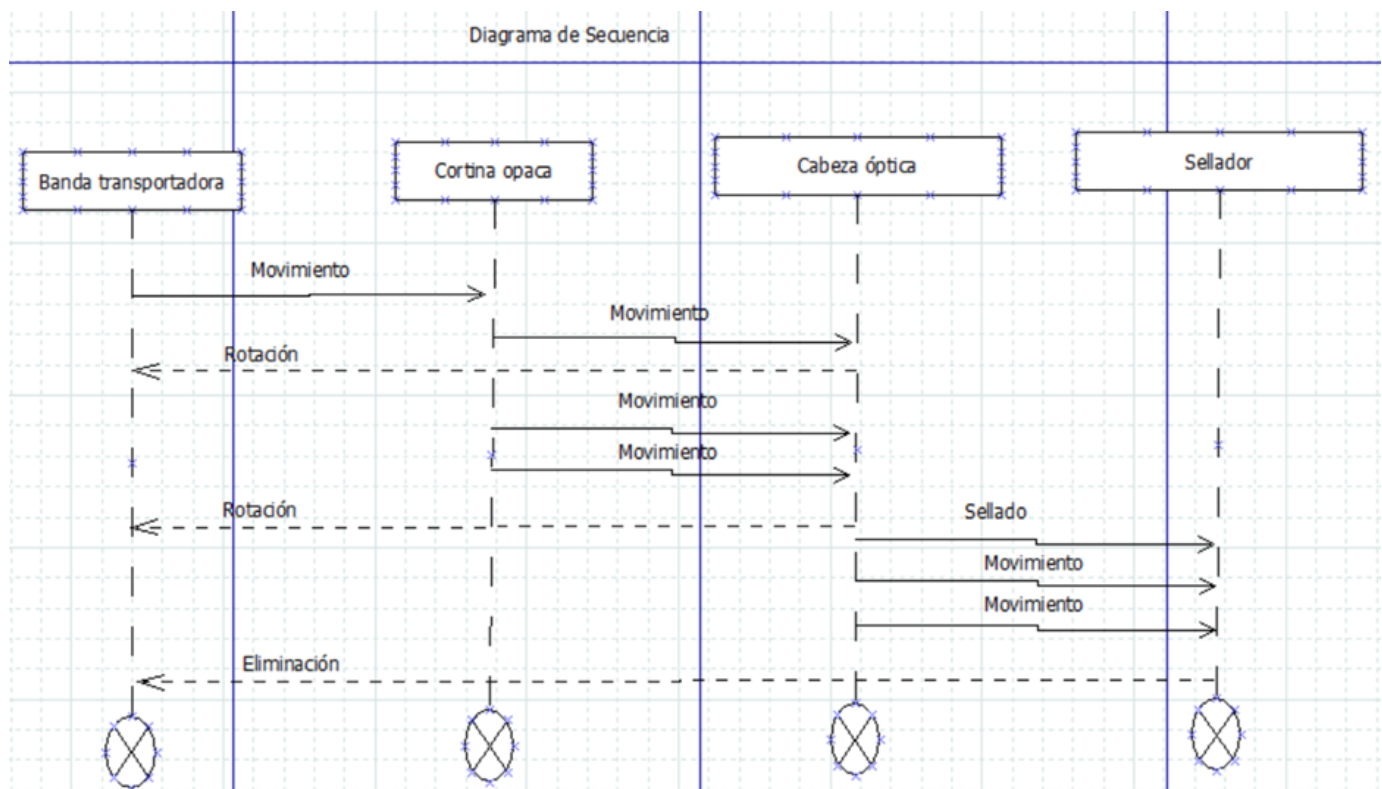


Fig. 1. Diagrama de secuencia.

otro componente indica el flujo del mensaje de control. Dice que un componente envía un mensaje para la finalización de su tarea al controlador del sistema y otro componente recibe un mensaje para iniciar la tarea asignada desde el controlador del sistema. Este diagrama de secuencia se aplica directamente para simular los procesos del sistema de montaje [1].

La cortina opaca se usa como barrera para evitar que la luz se propague hacia un área no deseada. Esta cortina puede estar hecha de materiales como aluminio, acero inoxidable, plástico o incluso una combinación de materiales. Esta cortina se coloca entre el láser y el área que se está iluminando, y sirve para bloquear la luz no deseada. Esto ayuda a evitar el almacenamiento de energía en áreas donde no se desea, lo que puede provocar daños. La banda transportadora es un sistema de transporte mecánico que mueve objetos a lo largo de una línea de producción y permite que el láser trabaje más rápido y con mayor precisión. Además, reduce la posibilidad de daños a los materiales mientras se mueven a través de la línea de producción. La cabeza óptica se usa para controlar la dirección y la potencia del haz láser. Esto significa que puede utilizarse para enfocar la luz láser en un objetivo específico o para distribuir la luz láser en un patrón específico. Finalmente, el sellador utiliza un haz de luz para calentar el material y sellar.

#### Diagrama de actividades.

Un diagrama de actividades es una herramienta importante para el control de las tareas y tomar en cuenta el flujo

del mismo, siendo tan vital, por la carga de actividades de nuestro modelado. Es esencial para el diseño de un sistema de soldadura en una línea de ensamble porque proporciona una imagen clara y detallada para comprender y documentar la arquitectura del sistema. El diagrama de actividades también facilita la detección de problemas y la implementación del sistema, ya que ayuda a los ingenieros a identificar los problemas, abreviados a cada actor que aplica dicha actividad. El diagrama también es útil para identificar cualquier componente problemático. Este diagrama de actividades que se muestra en la figura 3 se observan una unidad de control del láser, y una unidad de control CNN que realmente son muy dependientes entre unas y otras, de ahí partimos al flujo de cada actividad a través de nuestro nuevo actor que vendría a hacer nuestra unidad de cámara que sería nuestro reconocimiento de cada una de nuestras piezas y nuestro componente que funciona como sensor, pasaremos a la parte del registrador y microcontrolador a partir de aquí pasamos a la última instancia que vendría a hacer el sellador total. Cada uno de nuestros componentes tendrá ciertas actividades que converjan en la obtención de las piezas soldadas sin margen de error, cada una con la necesidad, de checar la actividad de la otra, para que no exista un fallo de soldadura. [4]

#### Diagrama de componentes.

Un diagrama de componentes es una herramienta vital para la ingeniería de sistemas. Es esencial para el diseño de un sistema de soldadura en una línea de ensamble porque

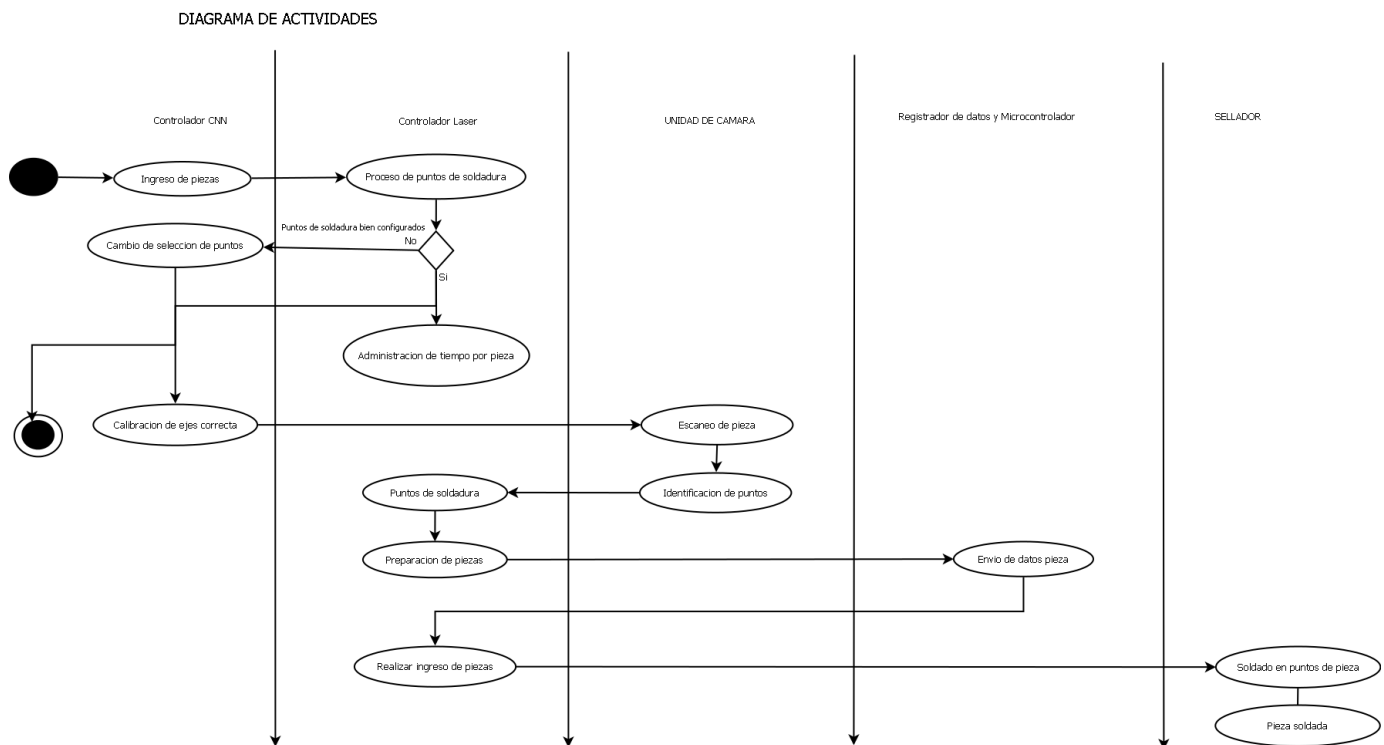


Fig. 2. Diagrama de actividades.

proporciona una imagen clara y detallada para comprender y documentar la arquitectura del sistema.

El diagrama de componentes también facilita el diseño y la implementación del sistema, ya que ayuda a los ingenieros a identificar, organizar y conectar los componentes del sistema. El diagrama también es útil para identificar cualquier componente problemático y definir la manera en que todos los componentes interactúan entre sí. Esto ayuda a garantizar que el sistema sea completamente funcional antes de su implementación.

Este diagrama de componentes que se muestra en la figura 2 muestra una unidad de control del láser, y una unidad de control CNN. El software de control permite la automatización y el control de la línea de ensamblaje, mientras que la unidad de control del láser le permite al software controlar el láser para realizar tareas específicas. La unidad de control CNN se conecta a los dispositivos de control para proporcionar información y controlar los procesos en la línea de ensamblaje. Esto permite que el sistema sea más eficiente y seguro, y garantiza una producción de alta calidad [2].

El software de control es responsable de la programación de los movimientos del robot, el control de los sensores, el procesamiento de los datos y la toma de decisiones. Esto permite al robot realizar tareas de forma autónoma, sin la necesidad de la supervisión humana. El software de control también se utiliza para procesar los datos recibidos de los sensores, para asegurar que el robot no se desvíe de su curso, para detectar cambios en el entorno y para realizar cálculos a partir de los datos recopilados. En este software implementaremos nuestra optimización para el seguimiento de

trayectorias. La unidad de control de láser es el dispositivo electrónico que controla el funcionamiento de un láser.

Esta unidad es responsable de la emisión, dirección y control de la energía. La unidad de control CNN utiliza una red neuronal convolucional para procesar la información de entrada, lo que le permite realizar predicciones sobre el entorno. Estas predicciones, a su vez, se utilizan para tomar decisiones sobre cómo debe actuar el robot. Esto les permite a los robots realizar tareas complejas con mayor precisión y eficiencia que los controladores tradicionales. Esta unidad se utiliza para el control de fallas o de entornos no considerados inicialmente. El registrador de datos puede almacenar información sobre la posición, los movimientos previstos, la ubicación y los datos de seguimiento. El microcontrolador se usa para controlar los movimientos, los sensores y otros dispositivos. Estos relés permiten aprovechar la potencia y la precisión de los láseres para realizar funciones más eficazmente que con otros métodos. La unidad óptica se usa para detectar y medir la luz, y, permite que la luz se concentre en un área específica para generar un patrón específico de luz. Finalmente, la unidad de cámara se usa para ver el entorno y detectar objetos [3].

#### Diagrama de casos de uso.

Podemos encontrar que un diagrama de actividades aplicado para un robot de soldadura implementado en una línea de ensamblaje es necesario conocer los diferentes componentes de control que existen, esto implementado a través de un diagrama de casos de uso, cada uno de estos diagramas, se verán implicados, para ciertas tareas, como podemos observar tomemos en cuenta que sabemos que existen 5 actores en

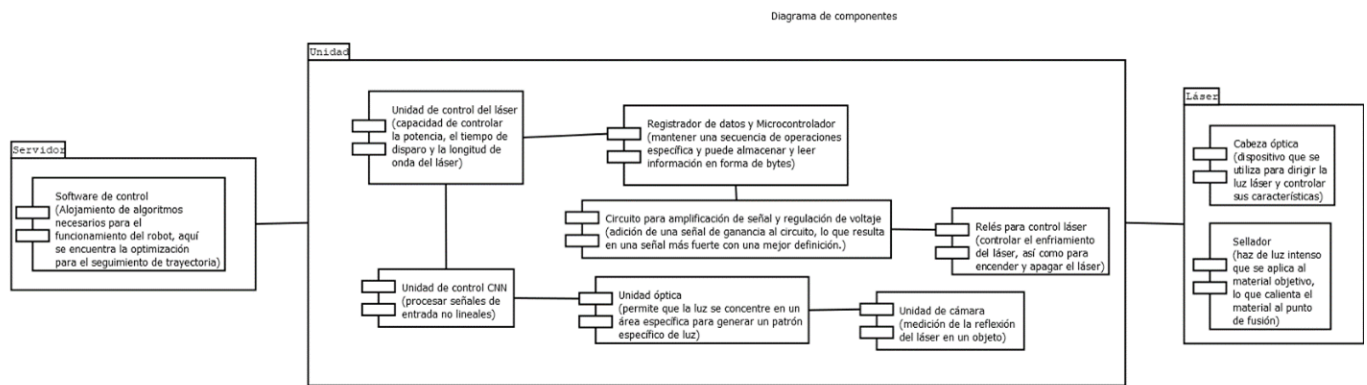


Fig. 3. Diagrama de componentes.

## Diagrama de caso de usos (Modelo de 4 barras)

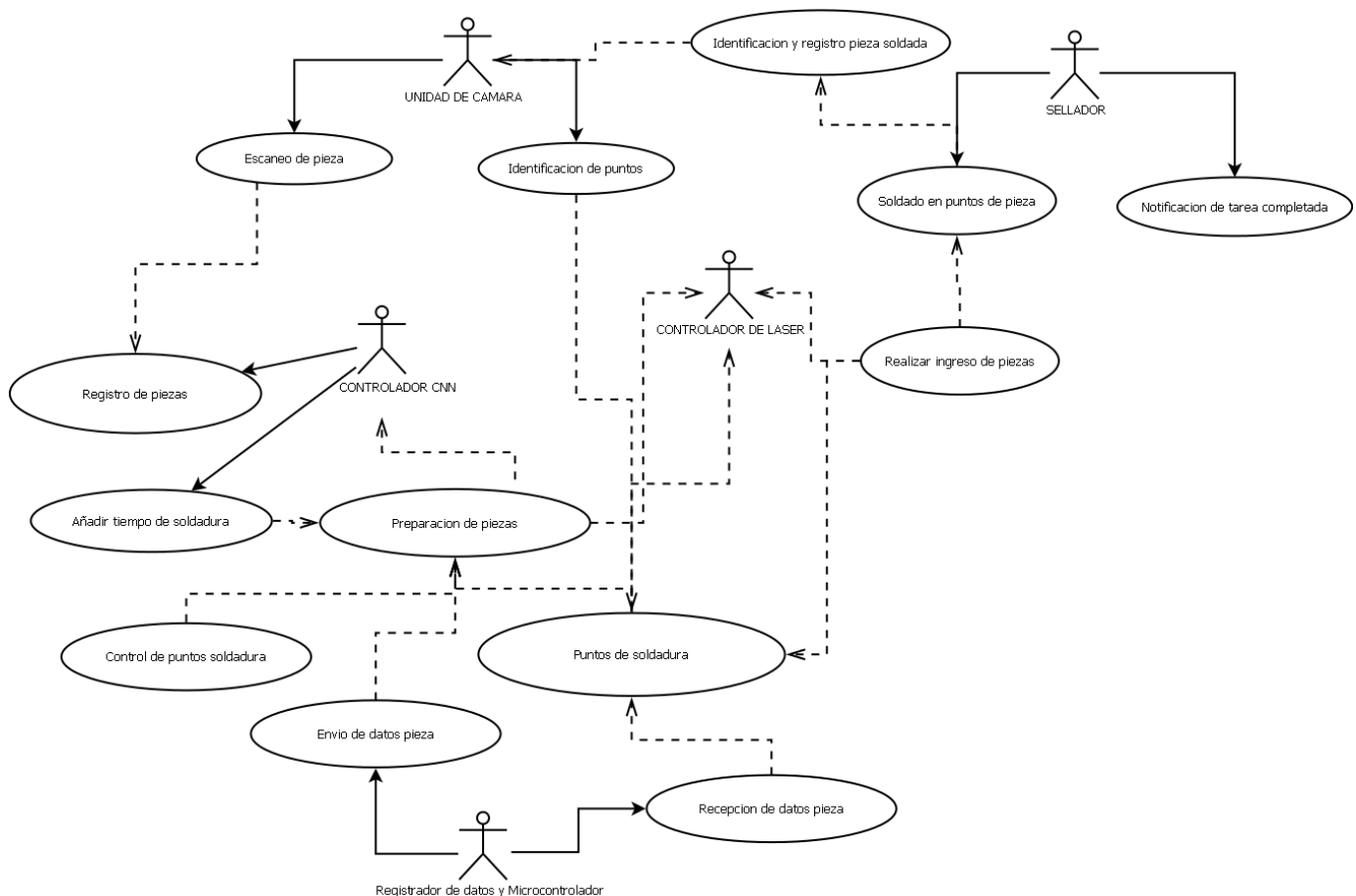


Fig. 4. Diagrama de casos de uso.

nuestro diagrama, tenemos nuestro actor controlador CNN que tiene como actividades las siguientes podemos ver algunas como, la de preparación de piezas, registro de piezas y añadir tiempo de soldadura también existe un nuevo actor que converge con el controlador CNN así podemos ver el controlador de láser con algunas actividades como las de

preparación de piezas, puntos de soldadura y realizar el ingreso de las piezas, como parte importante de nuestro sistema basado en un modelado físico observamos el actor de unidad de cámara las actividades son las siguientes, escaneo de pieza e identificación de puntos, como último actor encontramos el sellador de las piezas algunas de las actividades de este

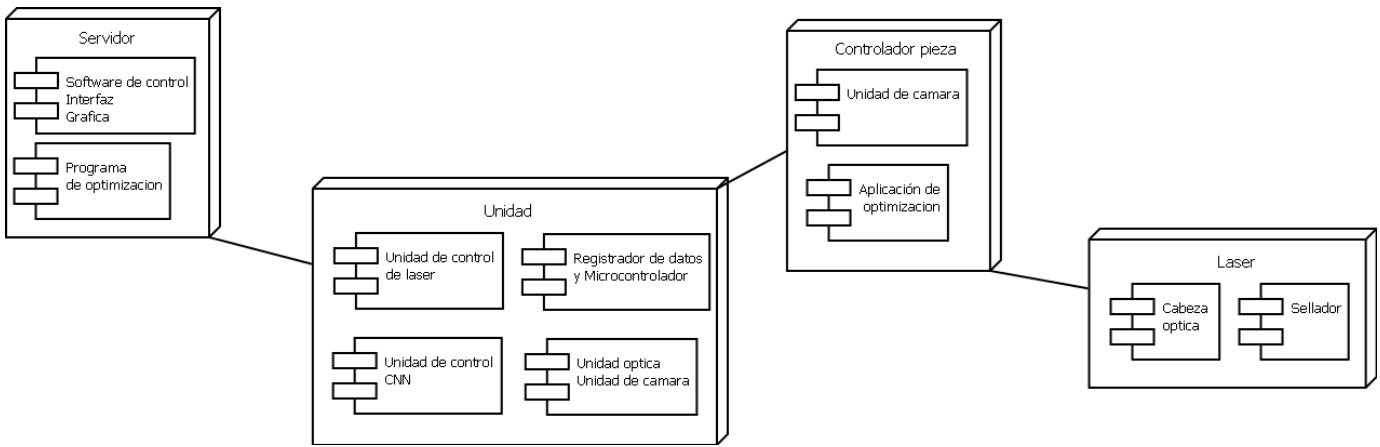


Fig. 5. Diagrama de despliegue.

sellador serían las siguientes, identificación y registro de piezas soldadas así como notificación de tarea completada.

Dice que un componente envía un mensaje para la finalización de su tarea al controlador del sistema y otro componente recibe un mensaje para iniciar la tarea asignada desde el controlador del sistema. Si bien existe cierta inclusión dentro de algunas actividades, los dos actores son totalmente dependientes, ya que, si alguno falla cierta actividad, el otro tendrá problemas al realizar nuestro objetivo, podemos observar cada actividad en nuestra figura [3].

#### Diagrama de despliegue.

Partimos con este diagrama a través del servidor empezando con nuestro servidor que contiene la parte gráfica, más que nada para el entendimiento del modelado poco a poco, a partir de ahí pasamos a la interfaz, convergente con el modelado y su uso dentro de la aplicación y un software de control que nos permite que el sistema sea más eficiente y seguro, y garantiza una producción de alta calidad [5].

El software de control es responsable de la programación de los movimientos del robot, el control de los sensores, el procesamiento de los datos y la toma de decisiones. Esto permite al robot realizar tareas de forma autónoma, sin la necesidad de la supervisión humana completamente pasando a el despliegue de la unidad que contiene una unidad de control de láser, así como el registro de datos y Microcontrolador, Unidad de control Con que contiene un software de control también se utiliza para procesar los datos recibidos de los sensores, para asegurar que el robot no se desvíe de su curso, para detectar cambios en el entorno y para realizar cálculos a partir de los datos recopilados. Parte de este software contiene un tipo de sensor dentro de la unidad óptica e unidad cámara que funciona como nuestro sensor dentro de nuestro modelado. En este software implementaremos nuestra optimización para el seguimiento de trayectorias que se encuentra dentro de nuestro otro componente de despliegue que es nuestro controlador de pieza.

La unidad de control de láser es el dispositivo electrónico que controla el funcionamiento de un láser. Esta unidad es

responsable de la emisión, dirección y control de la energía. Controlado a través de una cabeza óptica y un sellador que termina la tarea principal de nuestro modelado y acabando con una pieza totalmente soldada, dependiendo cada una de los componentes [5].

### III. RESULTADOS

Para solucionar el problema de las 4 barras debemos primero modelar matemáticamente como podemos encontrar el movimiento de el punto acoplador sobre la biela de un sistema manivela, biela, balancín; para esto comenzaremos por definir como será el sistema; para empezar definiremos el sistema en un marco de referencia rotado por cierto ángulo  $\theta_0$  y trasladado por ciertos  $x_0$  y  $y_0$ , teniendo de esta forma una forma fácil de cambiar de ubicación al sistema, junto con esto tendremos a 4 vectores los cuales representarán las 4 barras cumpliendo la ecuación 1 tal como se puede ver en la figura 1:

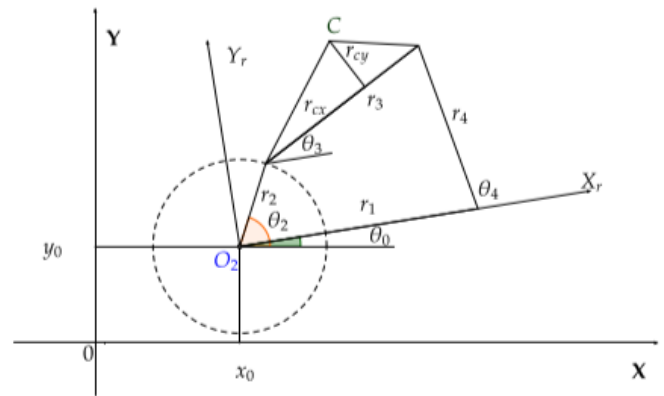


Fig. 6. Representación gráfica.

$$\vec{r}_1 + \vec{r}_4 = \vec{r}_2 + \vec{r}_3 \quad (1)$$

Con estas ecuaciones definidas podemos pasar a representarlas de manera polar y a través de la ecuación de Euler separarlas en su parte real y su parte imaginaria como se puede ver en la ecuación 2.

$$\begin{aligned} r_1 \cos \theta_1 + r_4 \cos \theta_4 &= r_2 \cos \theta_2 + r_3 \cos \theta_3 \\ r_1 \sin \theta_1 + r_4 \sin \theta_4 &= r_2 \sin \theta_2 + r_3 \sin \theta_3 \end{aligned} \quad (2)$$

Finalmente despejamos las parte correspondiente al cuarto vector, las elevamos al cuadrado y las sumamos para obtener una sola ecuación, consecuentemente despejando  $\theta_3$  y eliminando  $\theta_4$ , como se puede ver en la ecuación 3.

$$\begin{aligned} A &= 2r_3(r_2 \cos \theta_2 - r_1 \cos \theta_1) \\ B &= 2r_3(r_2 \sin \theta_2 - r_1 \sin \theta_1) \\ C &= r_1^2 + r_2^2 + r_3^2 - r_4^2 - 2r_1 r_2 \cos(-\theta_2) \\ \theta_3 &= 2 \arctan \left[ \frac{-B + \sqrt{B^2 + A^2 - C^2}}{C - A} \right] \end{aligned} \quad (3)$$

Finalmente con esta función podremos obtener el punto acoplador en función de  $\theta_2$ , que asignaremos nosotros,  $\theta_3$  que también es una función que solo depende de  $\theta_2$  y 2 constantes  $r_{cx}$  y  $r_{cy}$  que son vectores que definen la posición del punto acoplador referente al inicio de la biela, por lo que la función final estará en función de una sola variable, y al conocer un poco el sistema sabemos que  $\theta_2$  gira libremente hasta completar 360 grados, siendo este el ángulo de la manivela.

$$\begin{aligned} C_{xr} &= r_2 \cos \theta_2 + r_{cx} \cos \theta_3 - r_{cy} \sin \theta_3 \\ C_{yr} &= r_2 \sin \theta_2 + r_{cx} \sin \theta_3 + r_{cy} \cos \theta_3 \end{aligned} \quad (4)$$

Con esta información podemos regresar al marco de referencia general con una matriz de rotación y la suma del vector conformado por las coordenadas del origen movido, obteniendo lo siguiente:

$$\begin{aligned} C_x &= C_{xr} \cos \theta_0 - C_{yr} \sin \theta_0 + x_0 \\ C_y &= C_{xr} \sin \theta_0 + C_{yr} \cos \theta_0 + y_0 \end{aligned} \quad (5)$$

Obteniendo así la posición del punto acoplador en función de  $\theta_2$ , mientras este gira de 0 a 360 grados, (ya que es la manivela), a través de esta ecuación seremos capaces de evaluar ciertas variables para obtener las optimas para obtener una trayectoria deseada.

Para esto primero debemos de obtener las variables que manipulemos a través de nuestro algoritmo, las cuales serán el tamaño de todas las barras, el desplazamiento del punto acoplador de la biela, el ángulo de rotación del sistema, el desplazamiento del sistema y algunos puntos muestra de  $\theta_2$ , los cuales nos servirán para evaluar la función y compararla con lo que llamaremos los puntos de precisión los cuales nos ayudan a la hora de saber si el sistema esta siguiendo la trayectoria deseada, como se muestra a continuación:

$$\begin{aligned} \vec{p} &= [r_1, r_2, r_3, r_4, \\ &\quad r_{cx}, r_{cy}, \theta_0, x_0, y_0, \\ &\quad \theta_2^1, \theta_2^2, \theta_2^3, \theta_2^4, \theta_2^5, \theta_2^6] \end{aligned}$$

(6)

En este caso deseamos realizar el seguimiento de una recta, debido a la necesidad de la soldadora de convertir movimiento circular a vertical, por lo que usaremos 6 puntos para esto los cuales son:

$$\Omega = (20, 20), (20, 25), (20, 30), (20, 35), (20, 40), (20, 45) \quad (7)$$

Y con esto seremos capaces de generar una función objetivo que realice el análisis del error de nuestro mecanismo contra estos puntos de precisión:

$$\Sigma_{i=1}^N = [(C_{xd}^i - C_x^i)^2 + (C_{yd}^i - C_y^i)^2] \quad (8)$$

La cual nos encargaremos de minimizar con nuestro algoritmo, junto con esto debemos de someter esta función a ciertas restricciones, en las que se incluyen las que resultan al considerar la ley de Grashof y algunos límites para las variables:

$$g_1(\vec{p}) = p_1 + p_2 - p_3 - p_4 \leq 0 \quad (9)$$

$$g_2(\vec{p}) = p_2 - p_3 \leq 0 \quad (10)$$

$$g_3(\vec{p}) = p_3 - p_3 \leq 0 \quad (11)$$

$$g_4(\vec{p}) = p_4 - p_3 \leq 0 \quad (12)$$

$$g_5(\vec{p}) = p_{10} - p_{11} \leq 0 \quad (13)$$

$$g_6(\vec{p}) = p_{11} - p_{12} \leq 0 \quad (14)$$

$$g_7(\vec{p}) = p_{12} - p_{13} \leq 0 \quad (15)$$

$$g_8(\vec{p}) = p_{13} - p_{14} \leq 0 \quad (16)$$

$$g_9(\vec{p}) = p_{14} - p_{15} \leq 0 \quad (17)$$

$$g_{10}(\vec{p}) = -p_{10} \leq 0 \quad (18)$$

$$g_{11}(\vec{p}) = -p_{11} \leq 0 \quad (19)$$

$$g_{12}(\vec{p}) = -p_{12} \leq 0 \quad (20)$$

$$g_{13}(\vec{p}) = -p_{13} \leq 0 \quad (21)$$

$$g_{14}(\vec{p}) = -p_{14} \leq 0 \quad (22)$$

$$g_{15}(\vec{p}) = -p_{15} \leq 0 \quad (23)$$

$$g_{16}(\vec{p}) = p_{10} - 2\pi \leq 0 \quad (24)$$

$$g_{17}(\vec{p}) = p_{11} - 2\pi \leq 0 \quad (25)$$

$$g_{18}(\vec{p}) = p_{12} - 2\pi \leq 0 \quad (26)$$

$$g_{19}(\vec{p}) = p_{13} - 2\pi \leq 0 \quad (27)$$

$$g_{20}(\vec{p}) = p_{14} - 2\pi \leq 0 \quad (28)$$

$$g_{21}(\vec{p}) = p_{15} - 2\pi \leq 0 \quad (29)$$

[3]

Con todas restricciones seremos capaces de dejar las variables dentro de rangos deseados.

Con la función objetivo y las funciones de restricción

definidas podemos pasar al diseño del algoritmo, en este caso se hizo uso de una búsqueda guiada por las reglas de DEB para conocer la mas viable, las cuales son las siguientes:

- 1) Entre 2 respuestas viables, elegir la mejor.
- 2) Entre 1 respuesta viable o una no viable, elegir la viable.
- 3) Entre 2 respuestas no viables, elegir la que viole menos.

Lo único que debemos que debemos de hacer es implementar esta búsqueda con estas reglas y tendremos nuestro algoritmo:

---

**Algorithm 1** Optimizacion
 

---

```

actual ← Aproximacion
while true do
  nodos ← Generarhijos
  evaluacion ← Evaluarfunción
  costos ← Evaluarcostos
  nuevo ← SeleccionarporDEB
  if nuevo = actual or evaluacion ≤ 0.001 then
    Salir
  end if
  actual ← new
end while

```

---

Con esto ya podemos hacer nuestro algoritmo, solo debemos de definir como función algunas partes de este, primero la función generar hijos se encarga de generar en un arreglo todos los posibles movimientos que podemos hacer de un grupo de variables a otro, y esto se realiza sumando o restando un delta, donde cada operación en un variable genera un hijo, regresando al final estos hijos junto con el padre.

Justo después de esto podemos pasar a realizar la suma restricciones, junto con la evaluación de la función, para así poder aplicar las reglas de DEB, para realizar la suma de restricciones iteraremos por las restricciones, y en caso en que se violen procederemos a sumar los valores para generar un solo valor que nos determine que tanto viola todas las funciones cierto conjunto de variables.

Finalmente la elección por reglas de DEB se debe de realizar generalizándolas, para esto debemos de separar las variables en un grupo que no viole y uno que si, en el caso en que tengamos elementos en la lista que no viole las ordenaremos y seleccionaremos la óptima, y en el caso de no tomaremos la lista de las que violen y seleccionaremos la que menos lo haga, así implementando las reglas.

Por último pasamos por la condición de paro, la cual checa si la variable que tomamos del grupo de nodos es el padre, lo que querrá decir que ningún hijo es mejor y nos hemos atorado, o hemos logrado nuestro objetivo y nuestra solución es lo suficientemente buena.

Con todo esto, seremos capaces de encontrar una solución utilizando una buena aproximación inicial como se muestra en la siguiente imagen:

```

ans =
32.4877
7.8136
26.1799
32.4877
37.1617
16.1866
4.0430
-10.2006
57.6292
1.4840
2.3844
2.9234
3.4386
4.0009
5.0711

f =
0.0010

g =
-18.3663
-18.3663
-6.3078
-0.0000
-0.9004
-0.5390
-0.5152
-0.5623
-1.0702
-1.4840
-2.3844
-2.9234
-3.4386
-4.0009
-5.0711
-4.7992
-3.8988
-3.3598
-2.8446
-2.2823
-1.2121

```

Fig. 7. Resultados obtenidos

Finalmente podemos pasar a probar nuestro resultado a través de una simulación en GeoGebra, para esto primero debemos de crear el modelo de todo lo que hicimos y representar así nuestro sistema de 4 barras, obteniendo lo siguiente:

Para el cual se usaron los valores de:



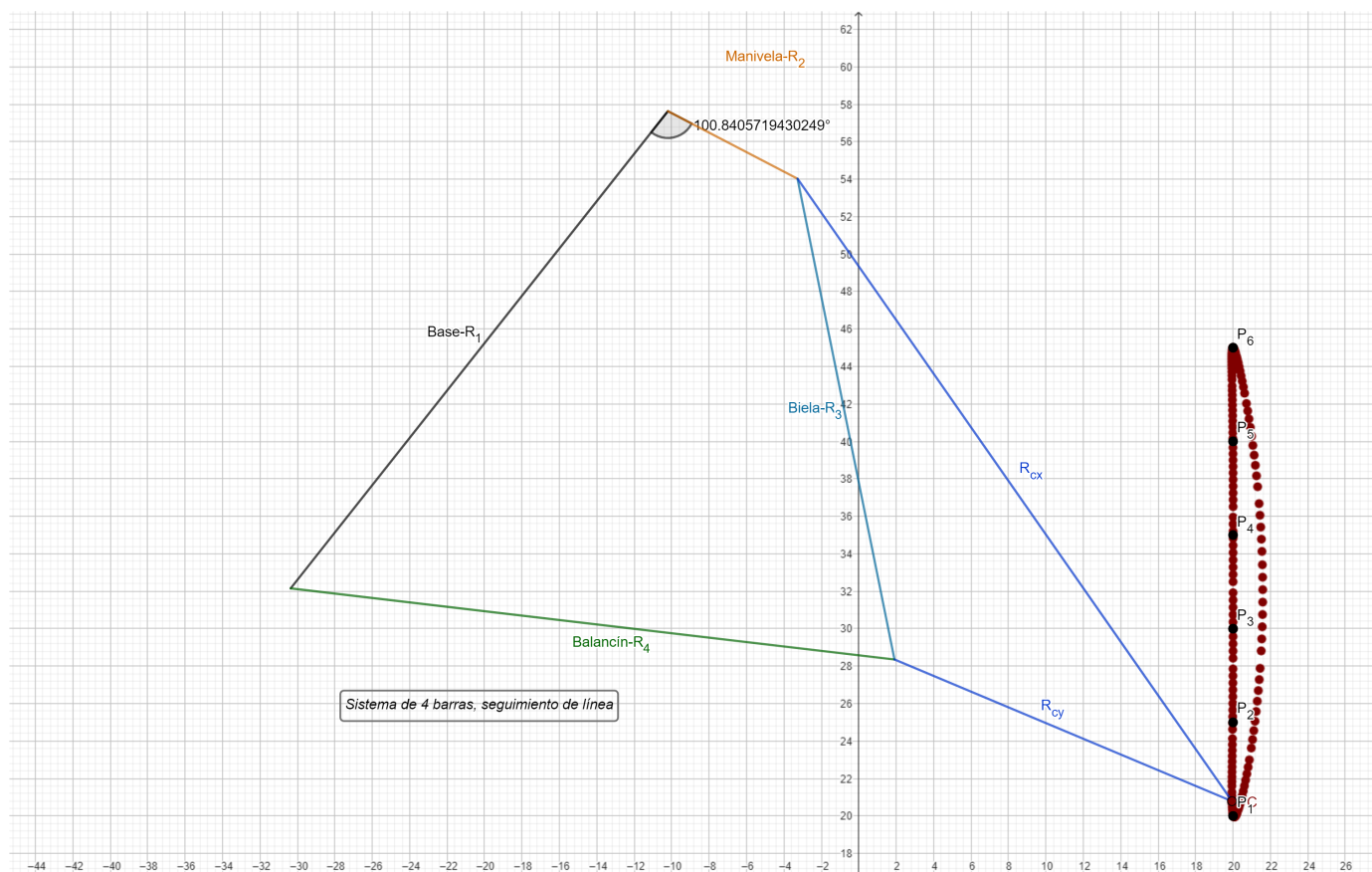


Fig. 8. Simulación

- $r_1 = 32.487720427927720$
- $r_2 = 7.813561926616888$
- $r_3 = 26.179902151170182$
- $r_4 = 32.487716359463600$
- $r_{cx} = 37.161684543377056$
- $r_{cy} = 16.186601683058120$
- $\theta_0 = 4.042971323933426$
- $x_0 = -10.200596830578100$
- $y_0 = 57.629242538297580$
- $FO = 0.001000011244422$

#### IV. OBSERVACIONES

El algoritmo utilizado en general no es tan eficiente comparado con otros métodos de optimización de funciones con restricciones, esto se da principalmente porque buscamos una solución a ciegas además de que le damos la misma importancia a cambiar los valores de las constantes como lo son el tamaño de la barra que a los valores de las  $\theta_2$ , provocando que aunque se tanga un sistema con constantes malas pero con ángulos que le ayuden a acercarse se mantenga sobre este camino, para solucionar este problema se podría generar otra forma de evaluar los puntos de precisión, como lo es evaluar la función en un intervalo y ver si algún valor de  $\theta_2$  coincide con el camino.

Además de lo anterior el algoritmo no tiene un método que le ayude a seguir el camino óptimo, como lo puede ser un

gradiente, si no que simplemente busca la solución a ciegas, a pesar de esto, si se sincroniza con un buen punto inicial, el algoritmo es capaz de encontrar soluciones con un error decentemente pequeño; otro problema que se presenta es la falta de exactitud en los cálculos y en el seguimiento de las variables, esto es provocado por como se manejan los tipos de datos, en MATLAB, lo que se utilizan son datos de tipo Long, los cuales pueden generar un problema de precisión para números con muchos decimales o números muy grandes, perdiendo el valor real de la variable.

Finalmente el algoritmo puede presentar muchos errores si no se sincroniza con variables que no generen valores reales o que violen inicialmente las restricciones del sistema, ya que aunque de cierta manera las reglas de DEB intentan solucionar esto puede que el algoritmo tome una cantidad de tiempo excesiva en encontrar el óptimo o que se desvíe a soluciones que no son óptimas por intentar cumplir las restricciones, por esto se debe tener mucho en cuenta con que datos vamos a comenzar nuestro algoritmo; ya que si no se puede presentar el problema de ciclo infinito, donde los valores ya no pueden encontrar soluciones mejores que la actual, mientras que la actual no es una solución lo suficientemente buena.

## V. CONCLUSIONES

Conocer métodos de optimización para problemas reales es una herramienta muy poderosa a la hora de solucionar problemas que se nos presenten en nuestra línea de investigación o de trabajo, además de que nos puede ayudar a la hora de considerar nuestras opciones a la hora de resolver un problema, este algoritmo es especialmente útil a la hora de tunear pequeños valores de una variable que ya se sabe que funciona, por lo que puede ayudar a modelar los últimos detalles de un sistema.

Aprender el uso de diferentes sistemas mecánicos nos puede llevar a comprender la aplicación de sistemas no tan complejos para la solución de un problema más grande, por ejemplo, el sistema de 4 barras (manivela, biela, balancín) es un sistema muy útil en campos de línea de producción y soldaduras como ya se habló anteriormente, pero también nos puede servir para sistemas más sencillos como una máquina de coser o un extractor de petróleo, cosas que nos rodean y que sin ellas no tendríamos muchas cosas de las que prescindimos hoy en día.

Finalmente se logró utilizar el algoritmo propuesto para terminar de sincronizar variables que se conocían que funcionarían para el problema, por lo que podemos dar como cumplido nuestro objetivo, junto con toda la información que logramos entender y comprender este trabajo fue bastante fructífero.

## REFERENCES

- [1] Park Hong-Seok Lee G.B. "Development of digital laser welding system for automobile side panels." International Journal of Automotive Technology. [https://www.researchgate.net/publication/286324677\\_Development\\_of\\_digital\\_laser\\_welding\\_system\\_for\\_automobile\\_side\\_panels](https://www.researchgate.net/publication/286324677_Development_of_digital_laser_welding_system_for_automobile_side_panels)
- [2] "Closed-loop control of laser power using the full penetration hole image feature in aluminum welding processes - scientific figure on researchgate". [https://www.researchgate.net/figure/Components-of-the-laser-welding-system-The-dashed-line-marks-those-components-added-for\\_fig2\\_232415276](https://www.researchgate.net/figure/Components-of-the-laser-welding-system-The-dashed-line-marks-those-components-added-for_fig2_232415276)
- [3] Vega-Alvarado, E., Santiago-Valentín, E., Sánchez-Márquez, A., Solano-Palma, A., Portilla-Flores, E. A., Flores-Pulido, L. (2014). Síntesis óptima de un mecanismo plano para seguimiento de trayectoria utilizando evolución diferencial. Res. Comput. Sci., 72, 85-98.
- [4] P. Eberhard, T. Gaugele, and K. Sedlaczek. Topology Optimized Synthesis of Planar Kinematic Rigid Body Mechanisms. Springer Vienna, Vienna, 2009.
- [5] "Closed-loop control of laser power using the full penetration hole image feature in aluminum welding processes". [www.researchgate.net. https://www.researchgate.net/figure/Components-of-the-laser-welding-system-The-dashed-line-marks-those-components-added-for\\_fig2\\_232415276/actionsreference](https://www.researchgate.net/figure/Components-of-the-laser-welding-system-The-dashed-line-marks-those-components-added-for_fig2_232415276/actionsreference).