

# Algoritmo de optimización por enjambre de partículas para la sintonización de controladores en sistemas con retardo.

Bryan R. Parra Monterrosas  
*Instituto Politécnico Nacional*  
*ESIME Culhuacán*  
Ciudad de México  
bryanrpm1196@gmail.com

Diego Martínez Guillén  
*Instituto Politécnico Nacional*  
*ESIME Culhuacán*  
Ciudad de México  
diegomtzg@hotmail.com

Francisco D. Hernández Salas  
*Instituto Politécnico Nacional*  
*ESIME Culhuacán*  
Ciudad de México  
fhernandezs094@gmail.com

Basilio del Muro Cuéllar  
*Instituto Politécnico Nacional*  
*ESIME Culhuacán*  
Ciudad de México  
bdelmuro@yahoo.com

**Resumen**—Se presenta el algoritmo de enjambre de partículas implementado en lenguaje lisp, aplicado a la búsqueda de parámetros de un controlador usando un criterio de minimización de la señal del error determinado por la función de costo norma-2 para estabilizar sistemas que presentan retardo. Se propone que a través de un algoritmo iterativo de búsqueda como lo es el método de enjambre de partículas, se encuentren de manera automática los parámetros de un controlador con el cual sea posible estabilizar el sistema. El funcionamiento del algoritmo se muestra mediante algunos ejemplos para un controlador PI y PD para un sistema inestable de primer orden con retardo, así también para un controlador PID para un sistema de segundo orden con un polo inestable y otro estable con retardo.

**Index Terms**—particle swarm optimization, dynamical systems, control system, swarm intelligence, artificial intelligence, delay systems

## I. INTRODUCCIÓN

El análisis de los sistemas con retardo resulta complicado debido a que el término del retardo aparece en la ecuación característica del sistema en lazo cerrado como se muestra en la Figura (1), lo que implica un número infinito de raíces.

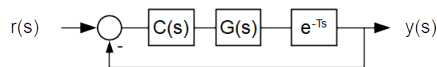


Figura 1: Sistema con retardo en lazo cerrado.

Otra técnica implica realizar el diseño en tiempo discreto donde el término del retardo se convierte en  $n$ -polos en el origen del círculo unitario descrito por la ecuación (1) que relaciona el tamaño del retardo y el tiempo de muestreo.

$$n = \frac{\tau}{T} \quad (1)$$

Otra opción corresponde al análisis en el dominio de la frecuencia, a partir de diagramas de bode y nyquist, donde se puede determinar la cota del retardo y el rango de valores en los que se puede estabilizar un sistema para un controlador P,PI o PID. [4] Algunas técnicas más elaboradas incluyen aproximaciones de Padé, series de Maclaurin o propuestas de un esquema observador. [2]

Sin embargo, en cualquiera de los casos, se presenta únicamente el rango de valores donde se puede estabilizar un sistema pero no se aplica ningún criterio que determine los valores óptimos del controlador. El algoritmo de enjambre de partículas presenta una alternativa práctica y automática de encontrar los parámetros donde la respuesta sea aceptable, ya que el objetivo del método es la optimización de los parámetros de una estructura definida y tiene la particularidad de converger a una solución. [1]

## II. DESCRIPCIÓN DEL ALGORITMO

El algoritmo de enjambre de partículas (PSO) por sus siglas en inglés, surge de una combinación de dos campos de estudio, la psicología y los métodos computacionales evolutivos. [3]

EL PSO se basa en el comportamiento colectivo por la búsqueda de alimento inspirado en parvadas y cardúmenes. Si algún individuo del grupo obtiene información de alguna fuente con mayor cantidad de alimento, se transmite el conocimiento a cada individuo y entonces todo el grupo se mueve en esa dirección, existiendo la probabilidad de encontrar una nueva fuente de alimento.

Abstrayendo el concepto, cada individuo se denomina partícula y representa una solución candidata al problema al que se aplica el algoritmo. El conjunto de estas partículas se denomina enjambre y el espacio donde se mueve se conoce como espacio de búsqueda.

Cada partícula, al estar en movimiento tiene asociada una posición definida por un vector de  $n$  elementos mostrada en la ecuación (2) al igual que una velocidad como se muestra en la ecuación (3).

$$\vec{x} = \{x_1, x_2, x_3, \dots, x_n\} \quad (2)$$

$$\vec{v} = \{v_1, v_2, v_3, \dots, v_n\} \quad (3)$$

Por otro lado, el movimiento de cada partícula estará influenciado por: la mejor posición que ha tenido hasta el momento y la mejor posición conocida por todo el enjambre. De manera analítica, la velocidad esta determinada por la ecuación (4) y la posición esta definida en función de la velocidad como se muestra en la ecuación (5):

$$v_i(k+1) = v_{i_{inr}}(k) + v_{i_{cgn}}(k) + v_{i_{scl}}(k) \quad (4)$$

$$x_i(k+1) = x_i(k) + v_i(k+1) \quad (5)$$

La velocidad mostrada en la ecuación (4) equivale a la suma vectorial de tres componentes: inercial, cognitiva y social.

$$v_{i_{inr}}(k) = w[x_i(k-1) - x_i(k)] \quad (6)$$

$$v_{i_{cgn}}(k) = c_{cgn}[x_{i_b}(k) - x_i(k)] \quad (7)$$

$$v_{i_{scl}}(k) = c_{scl}[x_{i_{sb}}(k) - x_i(k)] \quad (8)$$

La componente inercial mostrada en la ecuación (6) relaciona la última dirección en la que se movió la partícula. Corresponde a la distancia entre la posición anterior de la partícula  $x_i(k-1)$  y la posición actual  $x_i(k)$ .

La componente cognitiva observada en la ecuación (7) corresponde a la distancia entre la mejor posición obtenida por partícula  $x_{i_b}(k)$  y su posición actual. Esta componente se relaciona con la memoria de la partícula.

La componente social (8) equivale a la distancia entre la mejor posición de todo el enjambre  $x_{i_{sb}}(k)$  y la posición actual de la partícula  $x_i(k)$ .

Cada componente de la velocidad está multiplicada por un escalar que determina el peso en cada dirección. Estos factores reciben el nombre de factor inercial  $w$ , factor cognitivo  $c_{cgn}$  y factor social  $c_{scl}$ .

A partir de calcular la velocidad, es posible determinar la nueva posición de la partícula, descrito por la ecuación (5). Estas ecuaciones son las que determinan el movimiento a través del espacio de búsqueda cuya dimensión  $R^n$  se determina a partir del numero de elementos que componen la posición de la partícula.

La relación entre los coeficientes cognitivo y social se traduce en una convergencia-exploración. Si el valor del coeficiente cognitivo es mayor que el social, cada partícula tiene una mayor libertad de movimiento por lo que el

espacio de búsqueda a explorar es mayor. Por el contrario, cuando el coeficiente social es mayor que el cognitivo, las partículas se dirigen hacia la mejor posición que ha tenido el enjambre causando que el algoritmo converja rápidamente a esa solución. La constante inercial  $w$  influye en la aceleración de la partícula, de modo que ninguna partícula se estacionará directamente en la mejor posición concida por el enjambre, a excepción de un coeficiente inercial bajo, lo que se traduce en un desplazamiento lento y que a su vez implica una convergencia lenta.

La manera de definir cual ha sido la mejor posición es a través de una función  $f : R^n \rightarrow R$  con la cual se pueda determinar el desempeño de la partícula. A esta función se le denomina función de costo y varía dependiendo del problema a optimizar.

### III. APLICACIÓN DEL ALGORITMO

La aplicación del algoritmo requiere de un simulador que ya ha sido implementado [6] en lenguaje lisp donde la versatilidad del mismo permite definir la estructura que se muestra en la Figura (1) y variar únicamente los parámetros del controlador para un sistema dinámico definido.

La posición de la partícula se construye a partir de los parametros a optimizar. Por ejemplo, para un controlador PID mostrado en la ecuación (9), los parámetros corresponden a una ganancia proporcional  $k_p$ , una ganancia integral  $k_i$  y una ganancia derivativa  $k_d$  mostrado en (10).

$$C(s)_{pid} = k_p + k_d s + \frac{k_i}{s} \quad (9)$$

$$\vec{x}_i = \{k_p, k_i, k_d\} \quad (10)$$

En el algoritmo se generan y optimizan los parámetros del controlador que corresponden a la posición de la partícula, por lo que se simula y evalúa a través de la función de costo cada partícula. Para este trabajo se utiliza el criterio de la norma-2 del error definida en la ecuación (11).

$$f_c = \sqrt{\int_a^b |e(t)|^2 dt} \quad (11)$$

La norma-2 es una noción de distancia, para este caso representa que tan cerca o lejos se encuentra la señal de salida con respecto a la señal de referencia. De ese modo se puede determinar el desempeño de cada una de las partículas.

#### ■ Representación de sistemas

El simulador trabaja a partir de la definición y la interconexión de bloques lo que permite la independencia entre ellos.

Particularmente, el bloque donde se define el sistema trabaja con una representación en variables de estado. Para aumentar la versatilidad del simulador se implementa una librería donde se pueda hacer la representación de un

sistema por los coeficientes del numerador y el denominador, la ubicación de los polos y ceros o mediante matrices y en cualquiera de esas representaciones internamente se hace la transformación a variables de estado.

La metodología que se implementó para la transformación es la representación de la forma conónica controlable [?] mostrada en (??) considerando al sistema definido por una función de transferencia descrita en (12)

$$G(s) = \frac{b_0 s^m + b_1 s^{m-1} + \dots + b_{m-1} s + b_m}{s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n} \quad (12)$$

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \dots & -a_1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ \dots \\ 0 \\ 1 \end{bmatrix} u \quad (13)$$

$$y = [b_n - a_n b_0 \quad b_{n-1} - a_{n-1} b_0 \quad \dots \quad b_1 - a_1 b_0] x + b_0 u \quad (14)$$

- Bloque de retardo

Otro bloque particular es el del retardo el cual se implementó como un registro que es una estructura tipo cola de longitud equivalente a la división del tiempo de retardo entre el tamaño de paso en la simulación descrito por la ecuación (15).

$$longitud\_del\_registro = \frac{time\_delay}{dt} \quad (15)$$

Como se observa en la Figura (2) cada elemento en la cola se inicializa con un valor cero y en cada intervalo de tiempo ( $dt$ ) se almacena la información de la señal de salida y se devuelve el último elemento en la cola.

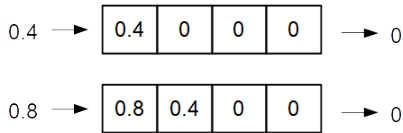


Figura 2: Estructura tipo cola implementada para el retardo.

#### IV. METODOLOGÍA DEL ALGORITMO

El proceso para la implementación del algoritmo se describe a continuación:

- Crear enjambre

Un enjambre se constituye de un número determinado de partículas iniciales, cada una es una lista de tres elementos: posición actual, posición anterior y mejor posición.

Una partícula, es un vector de dimensión equivalente al número de parámetros en el espacio de búsqueda los cuales son inicializados con valores aleatorios comprendidos en un rango. La posición actual, la posición anterior y la mejor

posición de una partícula inicial corresponden al mismo vector.

- Evaluar partícula

Cada vez que se crea o actualiza la posición de una partícula, esta debe ser evaluada mediante la función de costo mostrada en la ecuación (11). La posición es un dato compuesto de las coordenadas (parámetros) y el desempeño de la partícula (fitness).

- Actualizar la mejor posición del enjambre

Al tener la información de cada partícula, es posible determinar cual tiene la mejor posición en el enjambre, se compara el fitness de cada una y la que tenga el menor valor es elegida como la mejor posición del enjambre.

- Mover enjambre

El movimiento del enjambre, es el resultado de mover cada partícula, lo cual es un proceso iterativo para cada partícula. Mover un enjambre implica:

Mover partícula: Corresponde al cálculo de la velocidad descrito en la ecuación (4) y actualizar la posición mostrada en la ecuación (5). Al tener una nueva posición, se debe evaluar nuevamente mediante la función de costo.

Actualizar la mejor posición de la partícula: Es un proceso donde se compara el fitness de la nueva posición con el fitness de la mejor posición que ha tenido la partícula desde que se creó, si el fitness de la nueva partícula es menor, entonces se actualiza la mejor posición con esos datos.

El proceso es iterativo, una vez calculada la nueva posición de las partículas se retorna al paso donde se actualiza la posición del enjambre y el algoritmo continúa hasta el número de iteraciones que se hayan definido.

#### V. RESULTADOS

Para todos los ejemplos, los parámetros de simulación que se consideraron en el algoritmo se muestran en la Tabla (I)

Tabla I: Parámetros del algoritmo.

Parametro	Valor
$w$	0.8
$c_{cgn}$	0.5
$c_{scl}$	0.3
Partículas	5
Iteraciones	10

##### V-A. Ejemplo 1: Sistema de primer orden inestable

En este ejemplo se considera un sistema de primer orden con un polo inestable y retardo  $\tau = 0.2$  el cual tiene la función de transferencia descrita por la ecuación (16).

$$G(s) = \frac{1}{s-2} e^{-0.2s} \quad (16)$$

A continuación se muestran los resultados obtenidos para un controlador PI y PD con sus respectivas gráficas de simulación para el sistema descrito anteriormente.

### V-B. Control PI

La estructura de un controlador PI se constituye de dos parámetros, uno proporcional  $k_p$  y otro integral  $k_i$  por lo que la posición de cada partícula es un vector de dos elementos que representa cada una de las ganancias por lo tanto el espacio de búsqueda es de dos dimensiones.

En la Figura (3) se muestra la posición inicial de las partículas y en color rojo se destaca la mejor posición del enjambre. En la Figura (4) se presenta la posición de las partículas después de que el algoritmo se itero 10 veces, se puede observar como las partículas se mueven en dirección a la mejor posición del enjambre y así mismo ésta se actualiza.

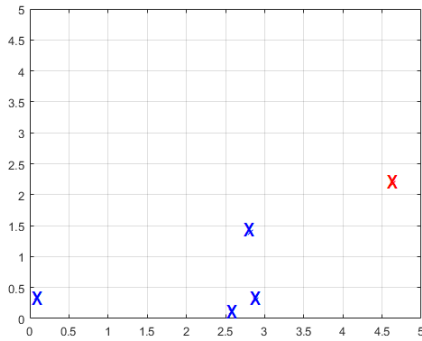


Figura 3: Posición de las partículas del enjambre inicial.

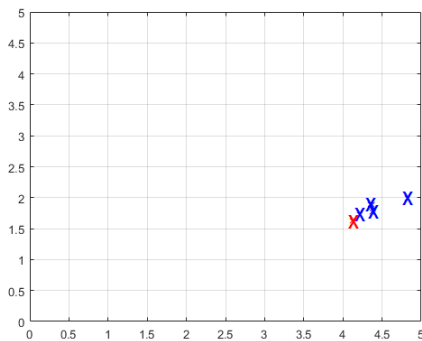


Figura 4: Posición de las partículas después de 10 iteraciones.

En la Tabla (II) se muestran los valores de los parámetros del controlador para la mejor posición del enjambre inicial y así también para la mejor posición después de 10 iteraciones.

En la Figura (5) se observa la señal de salida simulando el controlador con los parámetros de la mejor posición del enjambre inicial. Por otro lado, en la Figura (6) se muestra la señal de salida que se obtiene al simular el controlador con la mejor posición del enjambre después de las 10 iteraciones.

Tabla II: Parámetros del controlador PI.

Parámetros	Mejor posición del enjambre inicial	Mejor posición del enjambre después de 10 iteraciones
$k_p$	4.63	4.11
$k_i$	2.18	1.62
$fitness$	1.72	1.57

Se puede observar mediante las gráficas y los datos de la Tabla (II) como es la evolución del sistema a lo largo de 10 iteraciones, el valor de fitness disminuye y el sistema tiene una mejor respuesta.

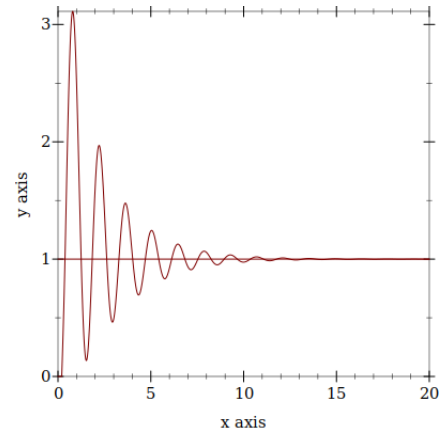


Figura 5: Gráfica de controlador PI con la mejor posición del enjambre inicial.

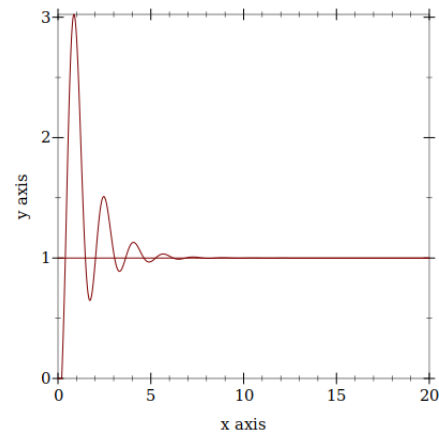


Figura 6: Gráfica de controlador PI con la mejor posición del enjambre después de 10 iteraciones.

### V-C. Controlador PD

Para el caso de un controlador PD, también se tienen dos ganancias, una proporcional  $k_p$  y otra integral  $k_d$ . La partícula como en el caso del controlador PI se compone de dos elementos, sin embargo, el resultado no es el mismo ya que la estructura del controlador es diferente.

En la Tabla (III) se muestran los parámetros de la mejor posición inicial y la mejor posición después de 10 iteraciones.

Tabla III: Parámetros del controlador PD.

Parámetros	Mejor posición del enjambre inicial	Mejor posición del enjambre después de 10 iteraciones
$k_p$	3.66	5.41
$k_d$	0.99	1.02
$fitness$	5.36	2.87

En la Figura (7) se muestra la señal de salida al simular el controlador PD con los parámetros de la mejor posición del enjambre inicial. Así mismo, se presenta en la Figura (8) la señal de salida del sistema simulando el controlador con los parámetros de la mejor posición del enjambre después de 10 iteraciones.

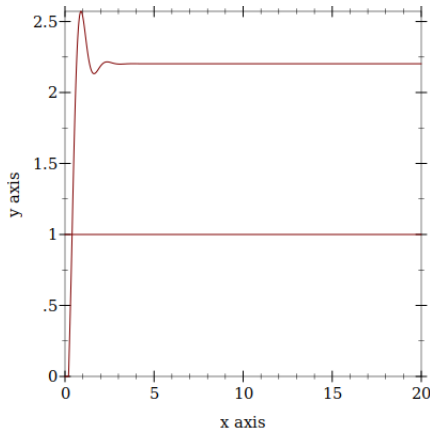


Figura 7: Gráfica de controlador PD con la mejor posición del enjambre inicial.

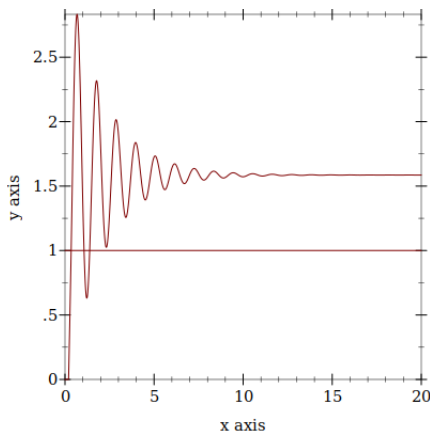


Figura 8: Gráfica de controlador PD con la mejor posición del enjambre después de 10 iteraciones.

## VI. SISTEMA DE SEGUNDO ORDEN

El siguiente ejemplo se realiza para un sistema de segundo orden con un polo estable y uno inestable con retardo  $\tau = 0.2$ , descrito por la ecuación (17).

$$G(s) = \frac{1}{(s-1)(s+4)} e^{-0.2s} \quad (17)$$

Los parámetros del controlador al inicio, en la quinta iteración y al término del algoritmo se muestran en la Tabla (IV).

Tabla IV: Parámetros del controlador PID.

Parámetros	Mejor posición del enjambre inicial	Mejor posición del enjambre después de 5 iteraciones	Mejor posición del enjambre después de 10 iteraciones
$k_p$	3.66	5.41	6.59
$k_i$	0.1	-0.02	0.68
$k_d$	4.31	5.60	5.68
$fitness$	23.69	9.24	2.69

Para este ejemplo, se aplica el algoritmo únicamente a un controlador PID. A continuación se presenta el espacio de búsqueda en el que se ubican las partículas al inicio y al término de 10 iteraciones. En la Figura (9) se observa la posición de las partículas al inicio del algoritmo, en color rojo se muestra la mejor posición del enjambre. Al paso de 10 iteraciones las partículas se dirigen a la mejor posición del enjambre como se muestra en la Figura (10).

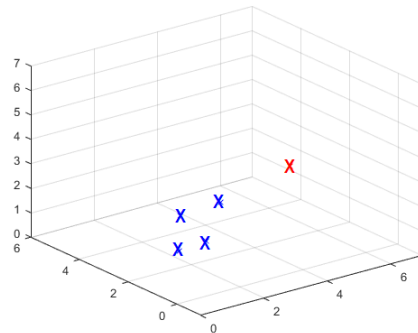


Figura 9: Posición de las partículas del enjambre inicial.

Particularmente, este ejemplo es muy ilustrativo de la convergencia del algoritmo. En la Figura (11) se muestra la simulación del controlador con los parámetros iniciales de la mejor posición del enjambre. En la Figura (12) se puede observar la simulación del controlador con los parámetros de la mejor posición del enjambre en la quinta generación donde se puede notar que los resultados tanto al inicio como en esta iteración no tienen una respuesta aceptable. Sin embargo, en la décima iteración se muestra una gráfica con una mejor respuesta como se muestra en la Figura (13).

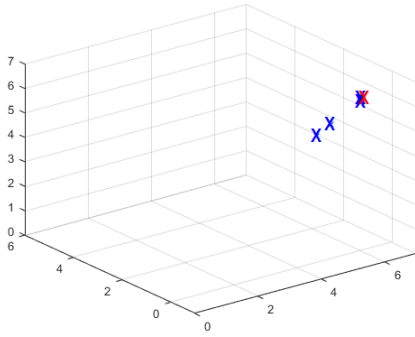


Figura 10: Posición de las partículas después de 10 iteraciones.

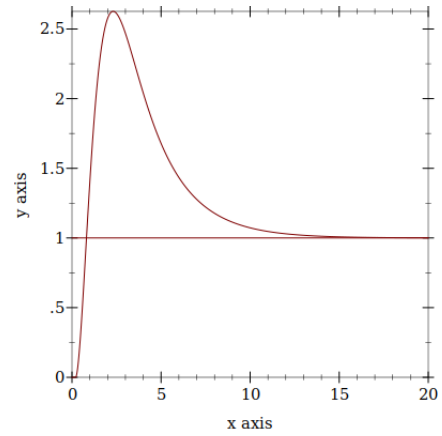


Figura 13: Gráfica de controlador PID con la mejor posición del enjambre después de 10 iteraciones.

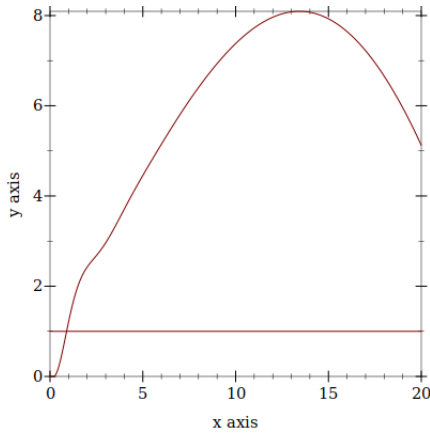


Figura 11: Gráfica de controlador PID con la mejor posición del enjambre inicial.

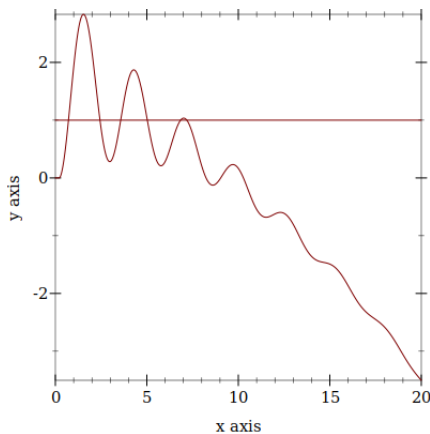


Figura 12: Gráfica de controlador PID con la mejor posición del enjambre en la quinta iteración.

## VII. CONCLUSIONES

En este trabajo se implementó el algoritmo de optimización por enjambre de partículas en lenguaje lisp y se obtuvieron resultados satisfactorios con la aplicación a la búsqueda de los parámetros de controladores de sistemas con retardo cuya respuesta se puede mejorar variando el número de partículas, el número de iteraciones y cada uno de los coeficientes. Al ser un método heurístico no se tienen reglas definidas, el desempeño del algoritmo depende de la exploración en los parámetros. En este trabajo se realizaron simulaciones con un número de partículas y de iteraciones pequeño, sin embargo, como lo muestran los resultados, particularmente el controlador PID para el sistema de segundo orden, se logra la estabilización cumpliendo el objetivo .

## REFERENCIAS

- [1] [Angeline, 1998]ange1998 Angeline, P. J., (1998). Proceedings of the 7th International Conference on Evolutionary Programming, 601-610. *Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences.*
- [2] [Del Muro, 2012]muro2012 Del Muro Cuéllar, B., Márquez Rubio, J. F., Velasco-Villa, M., and de Jesús Álvarez-Ramírez (2012). European Journal of Control, 18(5), 439. *On the control of unstable first order linear systems with large time lag: Observer based approach.*
- [3] [Kennedy and Eberhart, 1995]kennedy1995 Kennedy, J., and Eberhart, R. (1995). International Conference on Neural Networks IEEE. *Particle swarm optimization.*
- [4] [Lee, 2010]lee2010 Lee, S. C., Wang, Q. G., and Xiang, C. (2010). Journal of Process Control, 20(2), 235–239. *Stabilization of all-pole unstable delay processes by simple controllers.*
- [5] [Martínez Guillén, 2020]diego2012 Martínez Guillén, D., Cortés Rodríguez, D., and Hernández Salas F. (2020). Jornada de Ciencia y Tecnología Aplicada, 3(2), 98. *Desarrollo de un simulador de sistemas dinámicos para aplicación en métodos heurísticos de optimización.*
- [6] [Ogata, 2009]ogataControl Ogata, K. (2009). (5th ed.). Pearson. *Modern Control Engineering.*