Entre Ciencia e Ingeniería, ISSN 1909-8367
Año 6. No. 11 – Primer semestre de 2012, páginas 183-199

Diseño Óptimo de un Reductor de Velocidad Mediante Enjambre de Partículas[1]

Optimal Design of a Speed Reducer through Particle Swarm

Andrés Felipe Castillo Torres
Ingeniero Electrónico
Grupo de Investigación CEMOS
Universidad Industrial de Santander
andresfcastillo@live.com

Jhon Fernando Ávila Uribe
Ingeniero Electrónico
Grupo de Investigación CEMOS
Universidad Industrial de Santander
jhonavil2003@hotmail.com

Carlos Rodrigo Correa Cely
Ingeniero Químico
Maestría en Ingeniería área control de procesos
PhD. En Ingeniería área control de procesos
Grupo de Investigación CEMOS
Docente Universidad Industrial de Santander
crcorrea@uis.edu.co

Recibido octubre 20 de 2011 – Aceptado Mayo 30 de 2012

RESUMEN

El presente artículo describe la estrategia numérica utilizada para la optimización del conocido reductor de velocidad de Golinski mediante una modificación del tradicional método de enjambre de partículas (PSO), con una función de penalización que involucra las restricciones propias de su aplicación.

Este trabajo se motivó por la necesidad de utilizar componentes mecánicos óptimos en áreas como la robótica. Se encontró una notable precisión de

[1] Documento resultado del proyecto de investigación "Análisis del diseño óptimo de un reductor de velocidad", financiado por la Universidad Industrial de Santander.

sus resultados con relación a los reportados tanto por métodos clásicos de optimización, como por herramientas computacionales comerciales.

No obstante, su tiempo de cálculo fue una desventaja notoria, aunque no crítica, dada su aplicación fuera de línea. Confirman también estos resultados que el PSO, y la variante utilizada en particular, es una herramienta matemática sencilla y robusta para optimización de problemas en ingeniería.

Palabras Clave: reductor de velocidad, PSO con restricciones, función de penalización.

ABSTRACT

This article presents the optimization of the well-known Golinski speed reducer, through a modified particle swarm optimization algorithm (PSO), which includes a penalty system for considering the system restrictions.

A good grade of similarity with results reported for both, traditional optimization methods and commercial software, was found. However and even though it isn't critical, computation times were higher with PSO. The conclusion is that this PSO modification is a robust and mathematically simple solution for engineering optimization problems.

Key words: speed reducer, PSO with constrains, penalty function.

1. INTRODUCCIÓN

Convencionalmente, en áreas como la robótica, se hace necesario realizar movimientos precisos y con una velocidad de posicionamiento definida. Por esta razón, se diseñan reductores de velocidad, que paralelamente cumplan criterios relacionados con su peso y dimensiones y que entreguen el torque y/o velocidad requerido. Es aquí donde un algoritmo robusto de optimización entra a ser pieza clave.

Teniendo en cuenta que los modelos matemáticos desarrollados están sujetos a restricciones, es fundamental seleccionar un método capaz de trabajar en presencia de ellas, determinando respuestas que estén dentro del conjunto de soluciones posibles en la vida real y, que simultáneamente, respeten las restricciones del diseño. Este artículo utiliza un modelo matemático como función objetivo, para el proceso de optimización de un reductor de velocidad sujeto a restricciones, y describe el método numérico utilizado para minimizarlo (Golinski, J., 1970, pp. 287-309).

2. FUNDAMENTOS TEÓRICOS

En robótica, generalmente se necesita de un mecanismo capaz de transmitir el movimiento de un actuador a una articulación, creando una relación entre movilidad y dimensión. Un reductor de velocidad toma gran importancia, debido a que sus características físicas desarrollan efectos sobre el conjunto motor-reductor y en todo el sistema. Este sistema de transmisión debe ser preciso, además de poseer ciertas características dimensionales apropiadas para su ubicación y ensamble dentro del espacio presupuestado por el diseñador.

La relación de engranajes no puede ser demasiado alta, puesto que una parte de la potencia del motor se puede estar empleando para mover los engranajes del reductor introduciendo pérdidas en el torque. La transmisión de velocidad en un reductor se realiza a través de sus engranajes. Por ende, las dimensiones de un reductor de velocidad se relacionan directamente con la geometría del diente, la tensión en los mismos, el material del piñón, la superficie de contacto, el ángulo de presión, la potencia transmitida y la velocidad angular.

Igualmente, su peso depende de la geometría del engranaje y la geometría de cada uno de sus ejes. Se propone un engranaje cilíndrico recto como modelo, ya que se utilizan habitualmente debido a su simplicidad y economía de mantenimiento (Sanchez, Francisco T., Pérez, Antonio, Sancho, Joaquin L., Rodríguez, Pablo J., 2006, pp. 54).

Reductor de Golinski

Se tomó como caso de ejemplo el problema del reductor de velocidad abordado inicialmente por Golinski en los años 70s, con el propósito de optimizar su peso, satisfaciendo simultáneamente restricciones impuestas en el tamaño de los ejes y piñones. El objetivo final es entonces minimizar su geometría para que el reductor rote a su más eficiente velocidad, estando su peso definido por sus dimensiones. Ver figura 1.

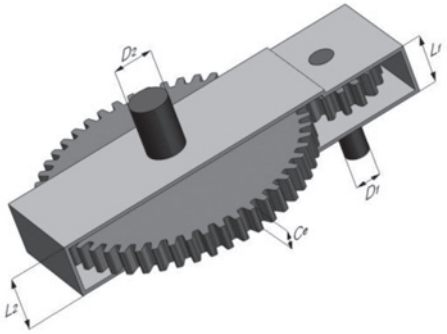


Figura1. Reductor de velocidad
Fuente: Los autores.

La función objetivo del reductor se define como, (Agrawal, G. et al, 2004, pp. 21):

$$f(x) = \sum_{i=0}^{2} [8M_d{}^2(N_d - 4{,}4)\frac{\pi C_e}{4} + ((M_d N_d - 6{,}4M_d)^2 - 5{,}76D_i{}^2)\frac{\pi C_e}{12} + (4{,}76D_i{}^3)\frac{\pi}{2} + \frac{\pi}{4}D_i{}^2 L_i] \quad (1)$$

Cabe resaltar que M_d se define como la relación entre la circunferencia primitiva y el número de dientes para cada piñón N_d. La definición formal del modelo matemático de las dimensiones de un reductor de velocidad se puede consultar en libros de diseño de engranajes (Budynas, N., 2006, pp. 348-383), (Grote, K-H. & Antonsson, E. K., 2008, pp. 334-398).

Método de Optimización con Enjambre de Partículas (PSO)

El PSO fue desarrollado en 1995 por Eberhart y Kennedy, influenciados por el comportamiento colectivo de algunas colonias de animales (Golinski, J., 1970, pp. 287-309). Su esquema computacional se describe como:

$$V_{i+1} = w * V_i + c_1 * r_1 * (P_l - X_i) + c_2 * r_2 * (P_g - X_i) \quad (2)$$

$$X_{i+1} = X_i + V_{i+1} \quad (3)$$

Como se puede observar en las ecuaciones (2) y (3), si se opera en un espacio vectorial de N dimensiones se tiene que el vector X_i representa la posición de cada partícula expresado como $Xi = (X_{i1}, Xi_2, \ldots, X_{in})$mientras que el vector V_i representa su velocidad expresada como $Vi = (V_{i1}, Vi_2, \ldots$

$,V_{in})$. Como se puede analizar en la ecuación (2), en cada posición de la partícula se analizan dos valores relacionados con el óptimo de la función, la mejor posición de cada partícula (P_{best}) representado por $Pi = (P_{il}, Pi_2, \ldots ,P_{in})$ y la mejor posición de toda la población (P_{global}) expresado mediante $Pi = (P_{g1}, Pi_{g2}, \ldots, P_{gn})$.

Finalmente se tienen c_1, c_2 las cuales son constantes positivas llamadas "coeficientes de aceleración", w es la ponderación de inercia; r_1 y r_2, valores aleatorios que están en un rango de 0 a 1.

PSO unificado (U-PSO)

Las propiedades de búsqueda del mejor local, P_l , y global P_g, son aplicadas en una mejora al algoritmo, donde las velocidades del global G_i y el local L_i se definen como sigue (Parsopoulos, K. E., Vrahatis, M. N., 2005, pp. 590-599), (Parsopoulos, K. E., Vrahatis, M. N., 2007, pp. 198-213):

$$G_i = \chi\left(V_i + c_1 * r_1 * (P_l - X_i) + c_2 * r_2\left(P_g - X_i\right)\right) \quad (4)$$

$$L_i = \chi\left(V_i + c_1 * r_3 * (P_l - X_i) + c_2 * r_4\left(P_g - X_i\right)\right) \quad (5)$$

Estas dos direcciones son unificadas en una sola ecuación.

$$U_i = u * G_i + (1 - u) * r_n * L_i \quad (6)$$

El factor de restricción en las ecuaciones (4) y (5) se introduce para limitar la velocidad de las partículas evitando así la explosión del enjambre. El factor de unificación en la ecuación (6) determina el dominio de las componentes global y local enriqueciendo las capacidades de búsqueda del algoritmo. La ecuación (3) que modela la posición de las partículas se remplaza por:

$$X_{i+1} = X_i + U_i \quad (7)$$

Para manejar las restricciones se trabajó con una función de penalización, que modifica la función objetivo mediante un factor que busca castigar aquellas partículas que no se encuentren dentro de la región factible.

La Función de Penalización

Los problemas en ingeniería generalmente están sujetos a restricciones de la forma:

$$g_i(x) \leq 0 \quad i=1,\ldots, m$$

Para el manejo de estas restricciones se usó una función penalizada *fp (x)* definida como (Konstantinos, E. et al, 2010, pp. 142-258):

$$fp(x) = f(x) + k^{\frac{3}{2}} * H(x) \qquad (8)$$

fp (x), corresponde a la función objetivo y *H (x)* es el factor de penalización. El valor de penalización varía con el número de iteraciones k. La función *H(x)* se definió en la ecuación (9) como,

$$H(x) = \sum_{i=0}^{m}\left(\theta\left(g_i(x)\right) * g_i(x)^{r(g_i(x))}\right) \quad \text{siendo} \quad g_i(x) = \begin{cases} g_i(x), & g_i(x) > 0 \\ 0, & g_i(x) < 0 \end{cases}$$

(9)

Los parámetros de penalización se establecieron en las ecuaciones (10) y (11) como:

$$\theta\left(g_i(x)\right) = \begin{cases} 10, & si \;\; g_i(x) < 0{,}001 \\ 20, & si \;\; 0{,}001 \le g_i(x) \le 0{,}1 \\ 100, & si \;\; 0{,}1 \le g_i(x) \le 1{,}0 \\ 300, & de\ otro\ modo \end{cases} \qquad (10)$$

$$r(g_i(x)) = \begin{cases} 1 & si \;\; g_i(x) < 1 \\ 2 & de\ otro\ modo \end{cases} \qquad (11)$$

$r(g_i(x))$, está relacionada con el peso de la función de penalización, es decir, aplica una sanción según sea la intensidad de la violación de las restricciones. Los valores de los parámetros de penalización que se muestran en la ecuación (11), se proponen como resultado de cálculos experimentales (Jinn-Moon Yang, Ying-Ping Chen, Jorng-Tzong Horng, and Cheng-Yan Kao, 2002, pp. 207-208).

El Enjambre

Para que las partículas permanezcan dentro de los límites de la región de búsqueda se definió (Gao F, Yibo Qi, Qiang Yin, Jiaqing Xiao, 2002, pp. 1-4):

$$x_i = \begin{cases} x_{max} - 0{,}1 * rand_1 * (x_{max} - x_{min}), & x_i > x_{max} \\ x_{min} + 0{,}1 * rand_2 * (x_{max} - x_{min}), & x_i < x_{min} \\ x_i, & de\ otro\ modo \end{cases} \qquad (12)$$

De igual manera para inicializar las partículas se tuvo en cuenta que se encontraran distribuidas aleatoriamente dentro de la región comprendida entre los límites superior e inferior. Para ello se emplearon las ecuaciones (13) y (14) (Ruben E. Perez and Kamiran Behdinan, 2007, pp. 376-377):

$$x_0 = x_{min} + rand_3(x_{max} - x_{min}) \quad (13)$$

$$v_0 = x_{min} + rand_4(x_{max} - x_{min}) \quad (14)$$

3. RESULTADOS Y ANÁLISIS

Las características del equipo en el cual se realizaron todas las pruebas son: Sistema Operativo Windows 7, procesador AMD Athlon II Dual-Core M300 2GHz y memoria RAM de 2 Gb. Para comprobar la efectividad del algoritmo se resolvieron varios ejercicios de los cuales sólo se presentan los resultados de uno de ellos (función helical valley); con dichos ejercicios se buscó observar la precisión, el efecto del factor de unificación, el tiempo de cómputo, el efecto que tiene la precisión sobre la calidad de las soluciones y el costo computacional que esto implica.

El error para cada variable se define en el algoritmo como el valor absoluto de la diferencia entre la partícula siguiente y la partícula anterior como se aprecia en la ecuación (15):

$$error = |x_{i+1} - x_i| \quad (15)$$

Para observar el comportamiento de la función objetivo y el efecto que tiene la precisión sobre ésta, se varió esta última, definida como parámetro de entrada y criterio de parada del algoritmo (una vez $p < error$). A continuación se muestran los resultados de uno de los ejercicios realizados y encaminados a verificar la fiabilidad de los métodos numéricos.

Ejercicio (función helical valley)

$$f(x) = 10\left(x_3 - 10\,\theta(x_1\,;\,x_2)\right) + \left(10\left(\sqrt{x_1^2 - x_2^2} - 1\right)\right)^2 + x_3^2\,, \qquad (16)$$

donde:

$$\theta(x_1\,;x_2) = \begin{cases} \dfrac{1}{2\pi}\tan^{-1}\dfrac{x_2}{x_1} & si\ x_1 > 0 \\ \dfrac{1}{2\pi}\tan^{-1}\dfrac{x_2}{x_1} + 0{,}5 & si\ x_1 < 0 \end{cases}$$

En la Tabla 1 se registra solamente el efecto de la precisión sobre la función objetivo con un factor de unificación μ = 0,5 y una población inicial de 100 partículas.

Tabla 1. Efecto de la precisión, μ = 0,5 ,100 partículas

Precisión	X_1	X_2	X_3	$f(x)$	Iter.	T (s)
$1\ x\ 10^{-1}$	0,9981	0,1549	0,2500	0,0750	5	3
$1\ x\ 10^{-3}$	1,0000	0,0000	0,0000	$2{,}6188\ x\ 10^{-10}$	117	3
$1\ x\ 10^{-4}$	**1,0000**	**0,0000**	**0,0000**	**$1{,}9729\ x\ 10^{-10}$**	**138**	**5**
$1\ x\ 10^{-5}$	1,0000	0,0000	0,0000	$5{,}6784\ x\ 10^{-12}$	147	4
$1\ x\ 10^{-6}$	1,0000	0,0000	0,0000	$6{,}3241\ x\ 10^{-10}$	82	4

Fuente: los autores

Como se aprecia en los resultados obtenidos en esta tabla para $p=1\ x\ 10^{-4}$ la solución tiende a un valor que satisface la función objetivo. Se advirtió un efecto no deseado al tomar valores del factor de unificación en diferentes rangos. La mejor opción se encontró entre $u = 0{,}3$ y $u = 0{,}5$. Sin embargo, $u = 0{,}5$ se seleccionó dado sus buenos resultados numéricos y bajo tiempo de cómputo. Los resultados confirman la respuesta reportada por las referencias (Rao, S., 2009, pp. 446-483).

Resultados para el Reductor de Golinski

La definición de la precisión estuvo influenciada por el sentido físico de la respuesta y estrechamente relacionado con la fabricación del reductor. Para la obtención de los resultados, algunos de los cuales se presentan a título de ejemplo, se utilizó una precisión $p=1\ x\ 10^{-4}$. La función objetivo quedó como:

$$f(\vec{x}) = 0{,}7854 C_e M_d{}^2 \left(3{,}3333 N_d{}^3 + 14{,}9334 N_d - 43{,}0934\right) - 1{,}508 C_e \left(D_1{}^2 + D_2{}^2\right) + 7{,}4777\left(D_1{}^3 + D_2{}^3\right) + 0{,}7854\left(L_1 D_1{}^2 + L_2 D_2{}^2\right) \quad (17)$$

En la Tabla 2 se muestran las restricciones que limitan la función objetivo

Tabla 2. Restricciones de la función objetivo

$g_1(\vec{x}) = \frac{27}{C_e {M_d}^2 N_d} - 1 \leq 0$ (18)	$g_2(\vec{x}) = \frac{397{,}5}{C_e {M_d}^2 N_d} - 1 \leq 0$ (19)	$g_3(\vec{x}) = \frac{1{,}93 {L_1}^3}{M_d N_d {D_1}^4} - 1 \leq 0$ (20)
$g_4(\vec{x}) = \frac{1{,}93 {L_2}^3}{M_d N_d {D_2}^4} - 1 \leq 0$ (21)	$g_5(\vec{x}) = \frac{1{,}0}{110 {D_1}^3} \sqrt{\left(\left(\frac{745{,}0 L_1}{M_d N_d}\right)^2 + 16{,}9 \times 10^6\right)} - 1 \leq 0$ (22)	
$g_6(\vec{x}) = \frac{1{,}0}{85 {D_2}^3} \sqrt{\left(\left(\frac{745{,}0 L_2}{M_d N_d}\right)^2 + 157{,}5 \times 10^6\right)} - 1 \leq 0$ (23)		$g_7(\vec{x}) = \frac{M_d N_d}{40} - 1 \leq 0$ (24)
$g_8(\vec{x}) = \frac{5 M_d}{C_e} - 1 \leq 0$ (25)	$g_9(\vec{x}) = \frac{C_e}{12 M_d} - 1 \leq 0$ (26)	$g_{10}(\vec{x}) = \frac{1{,}5 D_1 + 1{,}9}{L_1} - 1 \leq 0$ (27)
$g_{11}(\vec{x}) = \frac{1{,}1 D_2 + 1{,}9}{L_2} - 1 \leq 0$ (28)		

Fuente: Los autores

Cada restricción tiene un sentido físico relacionado con la tensión de flexión de los engranajes, la tensión de compresión máxima, límites de longitud, el número de dientes de piñón y sus valores numéricos son ajustados con los parámetros propuestos en (Agrawal, G. et al, 2004, pp. 21).

En total son once restricciones, todas expresadas como desigualdades y no existen parámetros fijos. El algoritmo debe ser capaz de encontrar una solución que satisfaga las restricciones. Se manejan siete variables continuas en el diseño del reductor, cada una con su respectivo límite inferior y superior, que representan las condiciones físicas impuestas por el modelo general del peso de un reductor de velocidad:

$$2{,}6 \leq C_e \leq 3{,}6\,;\, 0{,}7 \leq M_d \leq 0{,}8\,;\, 17 \leq N_d \leq 28\,;\, 7{,}3 \leq L_1 \leq 8{,}3\,;$$
$$7{,}8 \leq L_2 \leq 8{,}3\,;\, 2{,}9 \leq D_1 \leq 3{,}9\,;\, 5{,}0 \leq D_2 \leq 5{,}5.$$

Efecto de la Precisión

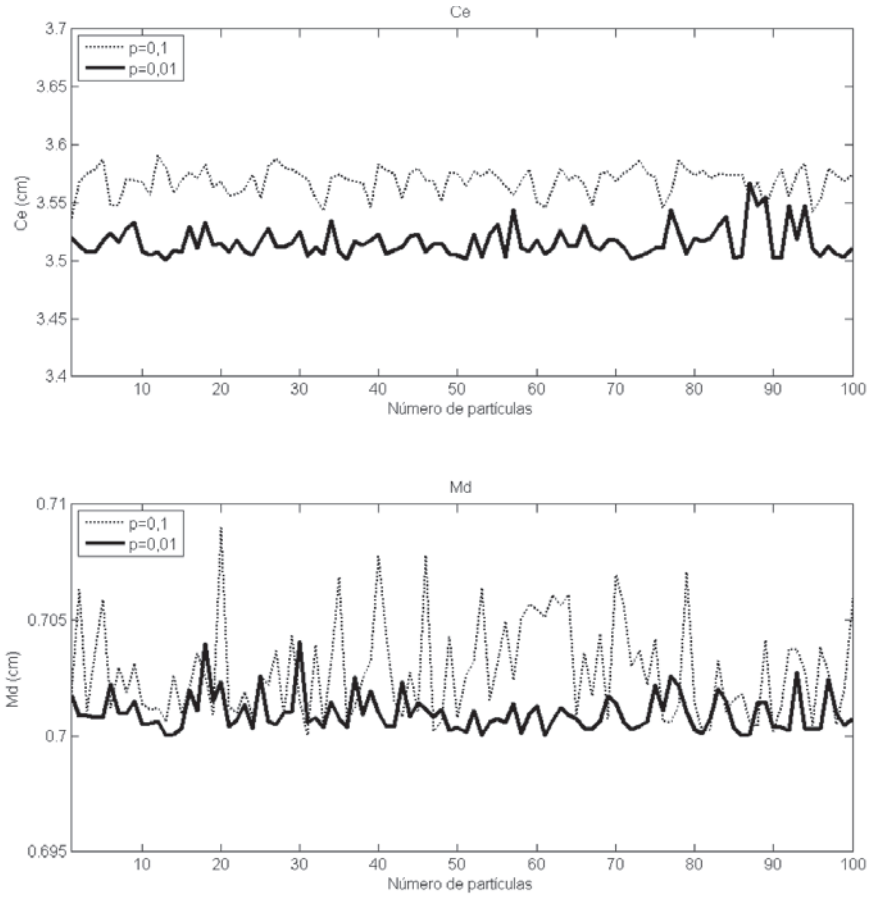
En la Tabla 3 se observa el efecto que produce aumentar la precisión sobre la función objetivo con un factor de unificación $u = 0{,}5$ y una población de 100 partículas.

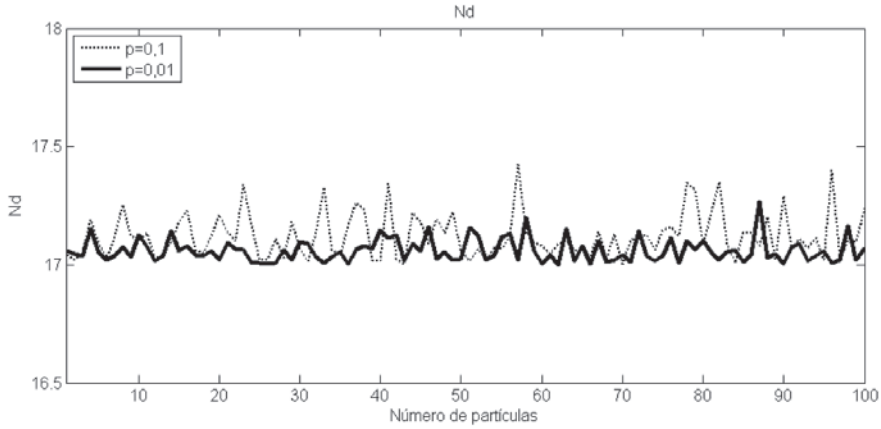
Tabla 3. Efecto de la precisión, ,100 partículas

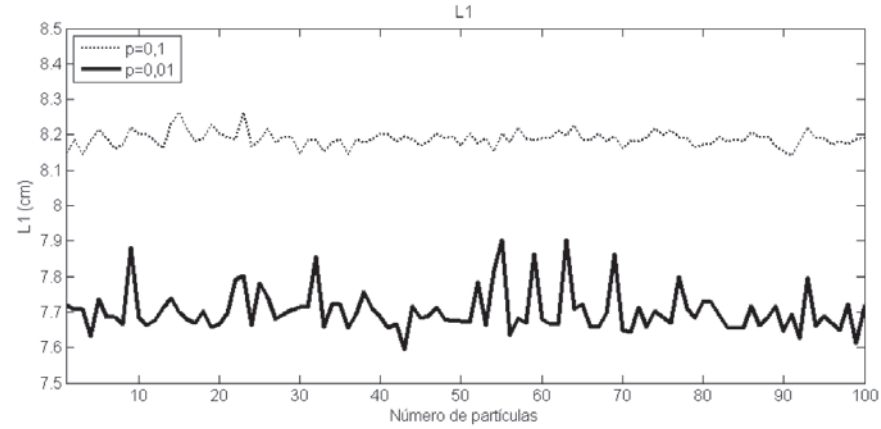
Precisión	x_1	x_2	x_3	x_4	x_5	x_6	x_7	$f(x)$	Iter.	t (s)
$1\,x\,10^{-1}$	3,5413	0,7016	17,0195	7,8147	7,8126	3,4736	5,3007	3070,1	13	9
$1\,x\,10^{-2}$	3,5020	0,7004	17,0051	7,6557	8,1416	3,3553	5,2893	3013,3	57	9
$1\,x\,10^{-3}$	3,5009	0,7000	17,0010	7,3579	8,2057	3,3546	5,2874	3008,0	4556	19
$1\,x\,10^{-4}$	**3,5001**	**0,7000**	**17,0001**	**7,3051**	**7,8017**	**3,3503**	**5,2868**	**2996,6**	**782765**	**22307**

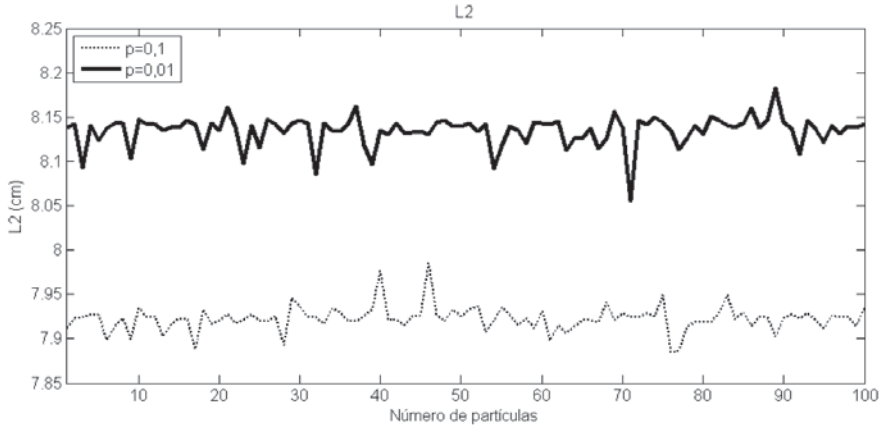
Fuente: los autores

En la Figura 2 se observa el efecto de aumentar la precisión sobre la función objetivo con una población inicial de 100 partículas y un factor de unificación $u = 0{,}5$.




Nd
p=0,1
p=0,01
Nd
Número de partículas


L1
p=0,1
p=0,01
L1 (cm)
Número de partículas


L2
p=0,1
p=0,01
L2 (cm)
Número de partículas

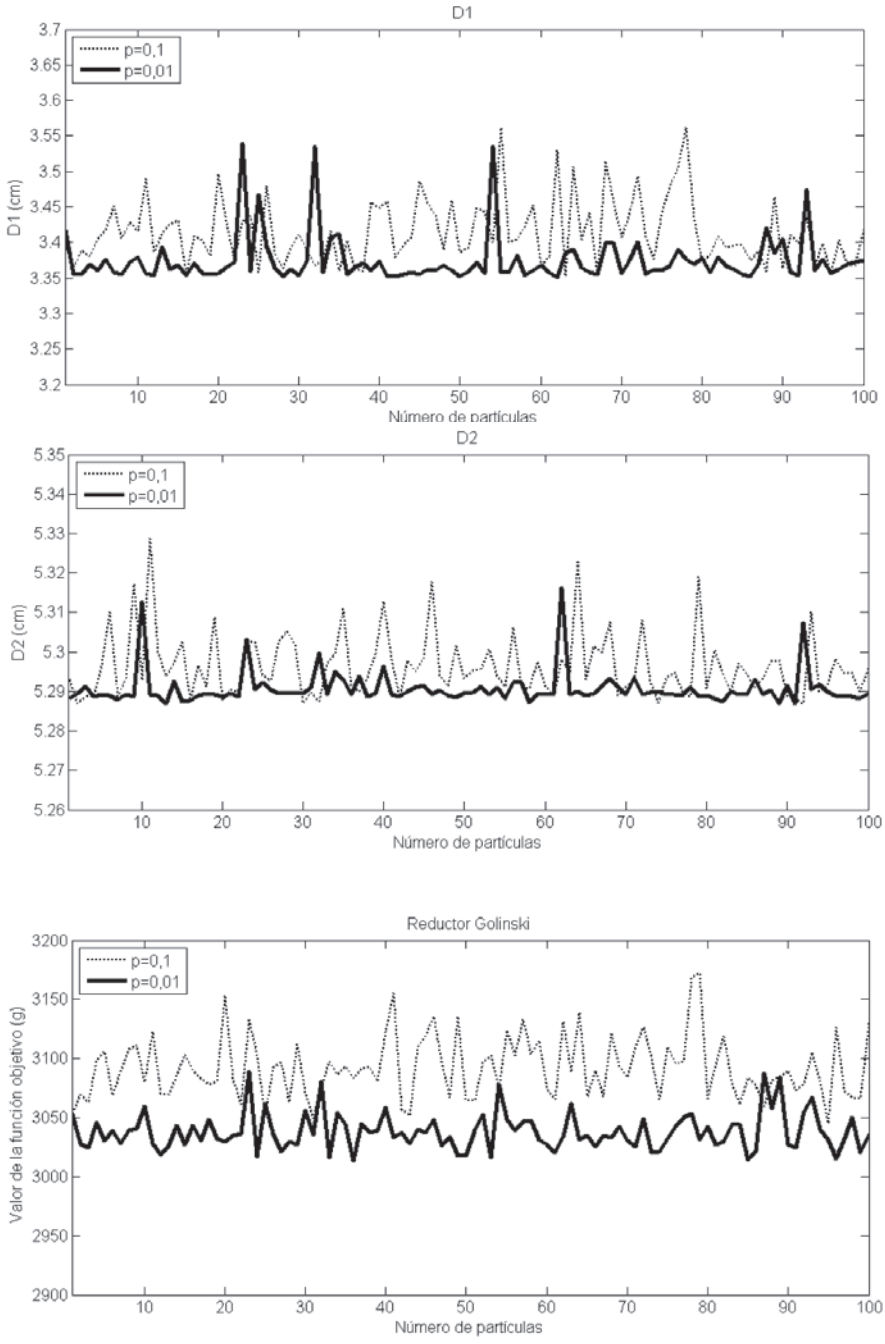


Figura 2. Efecto de la precisión, $p = 1 \times 10^{-1}$ y $p = 1 \times 10^{-2}$ en la función objetivo y sus variables

Fuente: Los autores

Los resultados obtenidos muestran que los valores adquiridos por la función objetivo y las siete variables mejoran al incrementarse la precisión. Como consecuencia de ello, aumentar la precisión requiere un mayor costo computacional, tal como se observa en los resultados. La Figura 3 corresponde a un aumento de la precisión hasta $p = 1 \times 10^{-4}$, para observar el efecto de ésta sobre el valor numérico de la función objetivo.

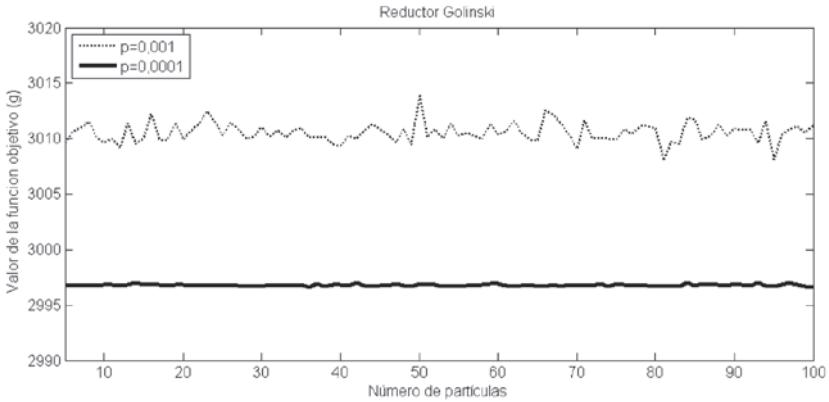


Figura 3. Efecto de la precisión. Se incluyen solamente la función objetivo
Fuente: Los autores

Al observar las gráficas de la Figura 3 se aprecia un cambio positivo en los valores numéricos que adquiere la función objetivo y las variables de diseño. El aumento de precisión se refleja en la oscilación de las partículas solución. Por tal motivo, los valores con $p = 1 \times 10^{-3}$ muestran un comportamiento más aleatorio en comparación con las soluciones encontradas para $p = 1 \times 10^{-4}$ a razón del valor que toma el *error* como criterio de convergencia, influyendo sobre el número de iteraciones y el tiempo de cómputo.

Efecto del Factor de Unificación

Se encontró que el rango para el factor de unificación ideal estaba entre $u = 0{,}5$ y $u = 1$, aunque el mejor valor para este factor fue de $u = 0{,}5$. Incluso ofrece un mejor resultado y un menor tiempo de cómputo. Otros valores como se acercaron también a la respuesta, sin embargo, requieren un mayor gasto de recursos.

Comparación entre PSO y U-PSO

En la Tabla 4 se muestra la diferencia entre PSO convencional (w = 0,5) y U-PSO $u = 0{,}5$ A=PSO, B=U-PSO. Los dos algoritmos trabajan con una precisión fija $p = 1 \times 10^{-2}$.

Tabla 4. Resultados de los dos algoritmos: PSO y U-PSO, $p = 1 \times 10^{-2}$

	N	x_1	x_2	x_3	x_4	x_5	x_6	x_7	$f(x)$	Iter.	t (s)
A	20	3.5012	0.7000	17.0036	7.8322	8.1234	3.3514	5.2879	3010,3	337245	215
B	20	3,5020	0,7003	17,0002	8,0005	7,8060	3,3534	5,2958	3011,3	427	22
A	40	3.5003	0.7000	17.0050	7.4991	7.8403	3.3538	5.2871	3001,2	829965	1091
B	40	3,5051	0,7001	17,0010	7,5407	8,1445	3,3777	5,2888	3017,1	281	8
A	80	3.5016	0.7000	17.0047	7.7038	7.8208	3.3571	5.2876	3004,1	13650	42
B	80	3,5013	0,7002	17,0027	7,6802	8,2760	3.3579	5,2890	3015,3	203	7
A	100	3.5004	0.7000	17.0006	7.3643	8.0103	3.3504	5.2871	3002,1	2305416	6586
B	100	3,5274	0,7013	17,0198	8,2634	7,8925	3,3923	5,2900	3040,1	22	7

Fuente: Los autores.

PSO convencional requiere mayor número de iteraciones para acercarse a la solución de U-PSO, reflejado en un mayor costo computacional y un incremento en el número de iteraciones.

Si bien los valores de la función objetivo son muy cercanos cabe resaltar que U-PSO empleó considerablemente menos tiempo en llegar al mismo valor en comparación con el PSO convencional. Se encontró que aumentando el número de partículas mayor es la rapidez en alcanzar el óptimo. No obstante, esta tendencia no es linealmente decreciente, ver Figura 4.

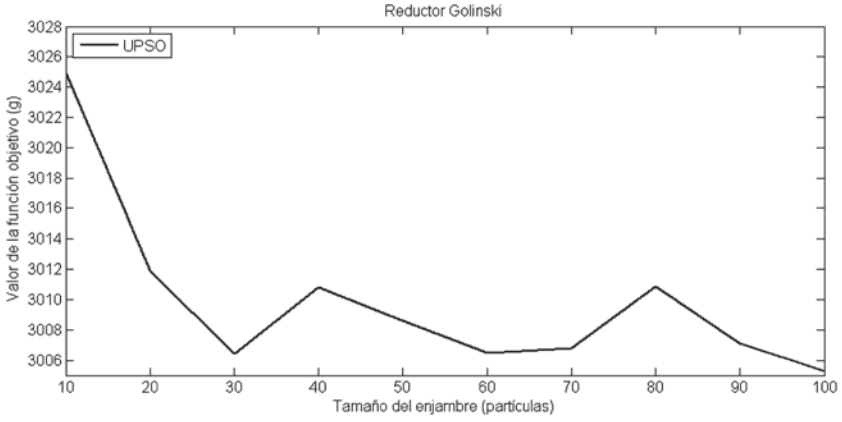


Figura 4. Función objetivo vs tamaño del enjambre,,
Fuente: Los autores

De la Figura 5 se observa que los valores solución están relativamente cerca unos de otros para un mismo tamaño del enjambre (baja dispersión). Esto se debe a los valores que toman las posiciones y velocidades inicialmente. Por esta razón el tiempo de cómputo y número de iteraciones dependen de qué tan cerca del óptimo estén ubicadas las partículas para iniciar su proceso de búsqueda.

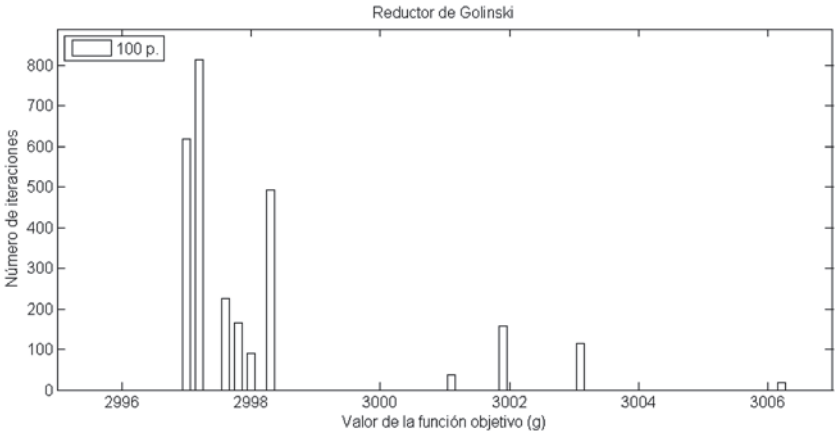


Figura 5. Número de iteraciones vs función objetivo,
$p = 1 \times 10^{-3}$, $u = 0{,}5$, 100 partículas
Fuente: Los autores

Factor de Restricción *X*

A medida que incrementa su valor (cercano a uno) se alcanza el óptimo, con la desventaja de incrementarse el número de iteraciones que a su vez se refleja en mayor costo computacional. Los demás parámetros del algoritmo se dejaron fijos, ver Figura 6.

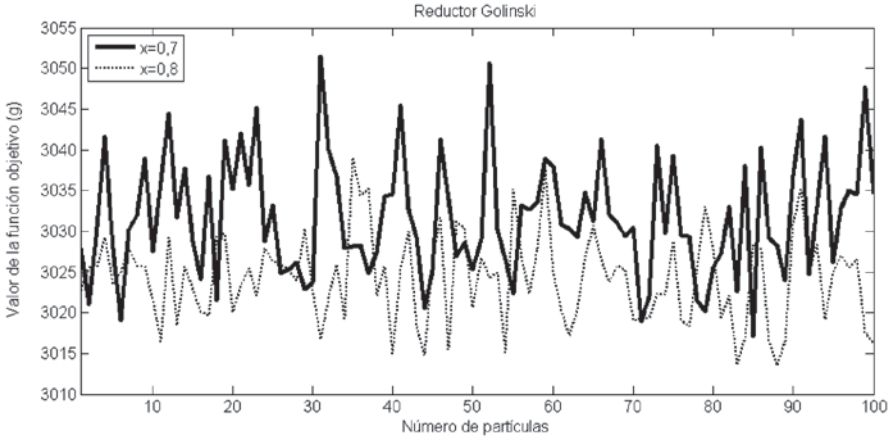


Figura 6. Función objetivo en función del número de partículas
y factor de restricción
Fuente: los autores

Entre menor sea el factor de restricción más dispersas están las partículas. Sin embargo, para valores desde 0,5 a 0,6 se presenta un

mejor comportamiento, entregando mejores resultados. Este problema de optimización se resolvió también mediante el uso de software comercial que utiliza optimización determinística y como caja negra para el usuario. Sus resultados se muestran en la Tabla 5. Se registran Igualmente los resultados obtenidos mediante el PSO modificado para el manejo de restricciones (SiC-PSO), (Cagnina, L. C. et al, 2008, pp. 323-324). Estos valores confirman la exactitud de los resultados obtenidos en este trabajo con una precisión de $p = 1 \times 10^{-4}$ y utilizando U-PSO como estrategia de optimización.

Tabla 5. Resultados del U-PSO contrastados frente a software comercial y al SiC PSO.

Método	x_1	x_2	x_3	x_4	x_5	x_6	x_7	$f(x)$
U-PSO	3,5001	0,7000	17,0001	7,3051	7,8017	3,3503	5,2868	2996,6093
Software comercial	3,5000	0,7000	17,0000	7,3000	7,8000	3,3502	5,2867	2996,3485
SiC PSO	3,5000	0,7000	17,0000	7,3000	7,8000	3,3502	5,2866	2996,3481

Fuente: Los autores

4. OBSERVACIONES Y CONCLUSIONES

El valor ideal para el factor de unificación para este problema en particular fue de 0,5 a razón de sus propiedades balanceadas. El reductor óptimo de Golinski encontrado con U-PSO para coincide bastante bien con los resultados obtenidos por otros métodos de optimización, demostrando su efectividad (ver Tablas 4 y 5). Aumentar la precisión por encima de $1x10^{-4}$ conllevaría a una respuesta más cercana a la teórica. Sin embargo, para el caso del reductor de Golinski no es atractivo dado que se está supeditado al proceso de fabricación, donde incrementar unas décimas de milésimas no representa ningún valor práctico.

BIBLIOGRAFÍA

- Agrawal,S. Parashar,K. W. English, and C. L. Bloebaum. ***"Web-based Visualization Framework for Decision making in Multidisciplinary Design Optimization".*** The State University of New York Buffalo, 2004, pp. 21.

- Budynas, N., (2006). ***"Mechanical Engineering: Shigley's Mechanical Engineering Design"***. Eighth Edition, McGraw-Hill, pp.348-383.
- Cagnina, L. C., Esquivel, S. C. & Coello, C. A., (2008). ***"Solving Engineering Optimization Problems with the Simple Constrained Particle Swarm Optimizer"***. Informática, Vol. XXXII, pp. 323–324.
- Gao F, Yibo Qi, Qiang Yin, Jiaqing Xiao, ***"A- Novel Particle Swarm Optimization With Special Boundaries Restriction Strategy"***. University of Technology, Wuhan, 430070, China, 2010, pp. 1-4.
- Golinski, J. (1970). ***"Optimal Synthesis Problems Solved by Means of Nonlinear Programming and Random Methods".*** Journal of Mechanisms, Vol V, pp. 287-309.
- Grote, K-H. & Antonsson, E. K., (2008). ***"Springer Handbook of Mechanical Engineering"***. Springer, pp. 334-398.
- Jinn-Moon Yang, Ying-Ping Chen, Jorng-Tzong Horng, and Cheng-Yan Kao, ***"Applying Family Competition to Evolution Strategies for Constrained Optimization".*** National Taiwan University, Taipei, Taiwan, 2002, pp 207-208.
- Konstantinos, E., Parsopoulos, K. E. & Vrahatis, M. N., (2010). ***"Particle swarm optimization and intelligence; advances and applications".*** Information science reference, Hershey, New York, IGI global, pp.142-258.
- Parsopoulos, K. E., Vrahatis, M. N., (2005). ***"Unified Particle Swarm Optimization in Dynamic Environments"***. Lecture Notes in Computer Science (LNCS), Vol. MMMCDXLIX, Springer, pp. 590-599.
- Parsopoulos, K. E., Vrahatis, M. N., (2007). ***"Parameter Selection and Adaptation in Unified Particle Swarm Optimization".*** Mathematical and Computer Modelling, 46 (1-2), Elsevier, pp. 198-213.
- Rao, S., (2009). ***"Classical Optimization Techniques",*** in Engineering Optimization: Theory and Practice, J. Wiley, Ed. 4a. New Jersey: J. Wiley & Sons, pp. 446-483.
- Ruben E. Perez and Kamran Behdinan, ***"Particle Swarm Optimization in Structural Design"***. University of Toronto, Institute for Aerospace Studies, Ryerson University, Department of Aerospace Engineering Canada, 2007, pp. 376-377.
- Sanchez, Francisco T., Pérez, Antonio, Sancho, Joaquin L., Rodríguez, Pablo J. (2006). ***"Mantenimiento Mecánico de Máquinas"***. Publicationt de la Universitat Jaume I. pp. 54