

Modelado Automático de Sistemas Mediante Algoritmos Computacionales de Optimización.

Juan J. Berger ^{a*}, Fernando E. Botterón ^a, Roberto E. Carballo ^a

^a *Universidad Nacional de Misiones, Facultad de Ingeniería, GIDE, Oberá, Misiones, Argentina.*

e-mails: juanjoseberger1999@gmail.com, botteron@fio.unam.edu.ar, robertocarballo@fio.unam.edu.ar

Resumen

En este trabajo se implementan algoritmos de optimización para el modelado de sistemas a partir de señales reales de entrada y salida, en este caso, dichas señales son la respuesta al escalón de un motor de corriente continua. Se evaluaron tres algoritmos que se implementan en el lenguaje de programación Python: Mínimos Cuadrados Medios (LMS), Algoritmos Genéticos (AG), y Búsqueda Tabú (TS). Los resultados mostraron que el algoritmo LMS no fue efectivo, ya que replicó la señal de salida sin considerar la señal de entrada. Aunque el algoritmo TS fue más rápido que el algoritmo AG, sus resultados fueron menos precisos y careció de robustez ante la variación de la cantidad de coeficientes que se le permitió utilizar. En contraste, los Algoritmos Genéticos destacaron por su robustez, ofreciendo un rendimiento superior a costa de un mayor tiempo de ejecución. Este estudio concluye que los Algoritmos Genéticos son la opción más adecuada para el modelado del sistema analizado y se implementará dicho algoritmo en el diseño de la herramienta de modelación para convertidores electrónicos de potencia.

Palabras Clave – *Algoritmo Genético, Búsqueda Tabú, Modelado, Mínimos Cuadrados Medios, Python.*

1 Introducción

En el presente trabajo se estudian diferentes formas de modelar sistemas a partir de sus señales de entrada y salida. El problema del modelado se trabaja en el dominio del tiempo discreto y bajo el enfoque de un problema de optimización. Debido a este enfoque, se proponen 3 tipos de algoritmos: El primero de ellos, se basa en una perspectiva teórica basada en disminuir un índice de rendimiento cuadrático, típicamente la suma cuadrática del error medio, lo que da origen a su nombre que es Mínimos Cuadrados Medios o en inglés “*Least Mean Squares*” abreviado como LMS. Los otros dos algoritmos no buscan una solución exacta del problema, en realidad buscan una lo suficientemente precisa y con relativa rapidez. Para lograr esto, buscan en el hiperespacio de soluciones un punto que represente un óptimo local o global mediante mecanismos basados en la teoría de la evolución genética, estos son los algoritmos genéticos abreviados como AG, o mecanismos basados en trayectoria, los algoritmos de Búsqueda Tabú o en inglés “*Tabu Search*” abreviados como TS. En función de la respuesta y bajo parámetros similares, se evalúa cuál de los tres algoritmos es el óptimo para modelar dicho sistema. El algoritmo que mejor se desempeñe modelando el sistema de una entrada y una salida (SISO) de primer orden sobreamortiguado, el del motor de CC, será utilizado para diseñar una herramienta de modelación de convertidores electrónicos de potencia. Estos últimos, suelen ser sistemas SISO aunque tienen una dinámica más compleja en comparación al motor de CC.

1.1 Modelado de Sistemas Discretos

* juanjoseberger1999@gmail.com

Para empezar, se debe comprender bien cuál es el problema que se intenta resolver, para ello se parte de la siguiente definición del modelado matemático de sistemas:

Un modelo matemático de un sistema dinámico se define como un conjunto de ecuaciones que representan la dinámica del sistema con precisión o, al menos, con una buena aproximación. Tenga presente que un modelo matemático no es único para un sistema determinado. Un sistema puede representarse en muchas formas diferentes, por lo que puede tener muchos modelos matemáticos, dependiendo de cada perspectiva [1].

Dicho de otra forma, modelar un sistema es poder describir con una expresión matemática su relación entrada-salida o causa-efecto. Esto limitándonos a un sistema de una sola entrada y una sola salida. De forma gráfica y ya trabajando en el dominio del tiempo discreto, en la Fig. 1 se puede observar dos señales donde $u(k)$ es la señal de entrada del sistema e $y(k)$ es la salida del mismo.



Fig. 1: Representación de un sistema o proceso de una entrada y una salida [1].

La expresión matemática que permite describir al proceso o sistema se conoce como función de transferencia. Dicha función se expresa en el dominio de la frecuencia, en este caso, en el dominio de la frecuencia compleja y discreta. Esto quiere decir que en vez de utilizar la variable k , que representa un número de secuencia de tiempo o de muestra, se utiliza la variable z , que representa lo mismo, pero en el dominio frecuencial. La forma que toma esta función de transferencia es la de un cociente de polinomios de la variable z , la misma puede observarse en la (1).

$$G_P(z) = \frac{Y(z)}{U(z)} = \frac{a_0 z^{N_a} + a_1 z^{N_a-1} + \dots + a_{N_a-1} z^1 + a_{N_a}}{z^{N_b} + b_1 z^{N_b-1} + \dots + b_{N_b-1} z^1 + b_{N_b}} \quad (1)$$

Donde: $G_P(z)$ es la función de transferencia del sistema o proceso; $Y(z)$ es la salida del sistema; $U(z)$ es la entrada del sistema; a_i son los coeficientes del numerador; N_a es el grado del polinomio del numerador o cantidad de coeficientes en el mismo; b_i son los coeficientes del denominador y N_b es el grado del polinomio del denominador o cantidad de coeficientes en el mismo.

Para obtener un sistema estable, es necesario que el grado del polinomio del denominador, N_b , sea mayor o igual al grado del polinomio del numerador, N_a . Proponiendo que sean iguales, es decir $N_b = N_a = N$, se puede simplificar la (1) en la (2). Es importante notar que el primer coeficiente del denominador, es decir, b_0 siempre es igual a 1 y no se lo expresa de forma explícita en la ecuación.

$$G_P(z) = \frac{Y(z)}{U(z)} = \frac{a_0 z^N + a_1 z^{N-1} + \dots + a_{N-1} z^1 + a_N}{z^N + b_1 z^{N-1} + \dots + b_{N-1} z^1 + b_N} \quad (2)$$

Aplicando la transformada Z inversa se puede obtener una ecuación recursiva del sistema a partir de la función de transferencia. La misma es la que se presenta en la (3) cabe aclarar que T es el periodo de muestreo.

$$y(kT) = a_0u(kT) + a_1u((k-1)T) + \dots + a_{N-1}u((k-N+1)T) + a_Nu((k-N)T) - b_1y((k-1)T) - \dots - b_{N-1}y((k-N+1)T) - b_Ny((k-N)T) \quad (3)$$

1.1 Base del procedimiento de identificación.

Si ahora se redefine a la (3), como una expresión para calcular una salida estimada del sistema, $\hat{y}(kT)$, es posible definir una nueva señal que cuantifique un error de identificación o de modelación. Dicha señal consistiría en la diferencia de la señal de salida real, $y(kT)$, y la salida estimada del sistema. Esto es lo que expresa la (4).

$$e(kT) = y(kT) - \hat{y}(kT) \quad (4)$$

En este punto es importante aclarar que $y(kT)$ es la señal que se mide del sistema y que $\hat{y}(kT)$ es la salida estimada en función de la entrada del sistema $u(kT)$ junto con muestras anteriores de la misma y de muestras anteriores de la salida, donde cada una de estas es multiplicada por sus respectivos coeficientes.

Por último, la suma del error cuadrático medio se puede definir como la (5).

$$E = \frac{\sum_{k=0}^{N_m-1} [e(kT)]^2}{N_m - 1} \quad (5)$$

Donde: E es el resultado de la suma del error cuadrático medio y N_m es la cantidad de muestras de las señales.

La premisa principal del modelado de sistemas consiste en hallar los coeficientes a_i y b_i que logran disminuir la suma del error cuadrático medio, en definitiva, es un problema de optimización multivariable.

Algo importante de lograr, además de disminuir E , es que los coeficientes a_i y b_i representen correctamente un sistema. Esto significa que, por ejemplo, los coeficientes a_i no tengan un valor negativo, dado que, estos coeficientes representan ganancias del sistema que generalmente no suelen ser negativas. Por otro lado, a partir de los coeficientes b_i se pueden hallar los polos del sistema que siempre deben estar dentro del círculo unitario del plano Z para que el sistema sea estable. Todos estos requisitos pueden evaluarse mediante funciones que calculen dicho error, la distancia de los polos al origen del plano Z y que verifiquen que los coeficientes del numerador sean positivos. Para categorizar una solución que no cumpla estos requisitos, es necesario un mecanismo o función que evalúe a dicha solución con un mérito o *fitness* relativamente grande, por ejemplo, un *fitness* de un valor de 10^6 . Como en este problema de optimización se intenta hallar una combinación de coeficientes que minimicen los parámetros mencionados anteriormente, un valor relativamente grande de desempeño ayuda a los algoritmos a converger en una solución útil. En definitiva, no sirve cualquier solución al problema de optimización, si no que los coeficientes tienen ciertas restricciones para ser una correcta representación del sistema.

1.2 Método de Mínimos Cuadrados Medios

Primeramente, se procederá a explicar cómo se puede resolver este problema una vez que se tienen las N muestras del escalón y la respuesta al escalón del sistema con el algoritmo de mínimos cuadrados. Todo esto es un breve resumen de lo presentado en la referencia [1] respecto a este tema.

Para hacer un planteamiento general del problema de modelación previamente descrito, consideraremos un proceso del que se obtienen observaciones de su salida y que dependen de un dato x , y que distintas observaciones proporcionan distintos pares $(y(k), x(k))$.

Sea un modelo de la forma:

$$\hat{y}(k) = \theta_1 \varphi_1(x_k) + \theta_2 \varphi_2(x_k) + \dots + \theta_n \varphi_n(x_k) \quad (6)$$

Donde, φ_j son funciones conocidas y θ_j los coeficientes del modelo a determinar. Es decir, φ_j son las entradas y salidas del sistema con sus respectivos atrasos y θ_j son los coeficientes a_i y b_i .

Considerando los N distintos pares $\{y(k), x(k)\}$ se puede introducir las siguientes notaciones matriciales y vectoriales.

$$\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}; \mathbf{\Phi} = \begin{bmatrix} \varphi_1(x_1) & \varphi_2(x_1) & \dots & \varphi_n(x_1) \\ \varphi_1(x_2) & \varphi_2(x_2) & \dots & \varphi_n(x_2) \\ \vdots & \vdots & & \vdots \\ \varphi_1(x_N) & \varphi_2(x_N) & \dots & \varphi_n(x_N) \end{bmatrix}; \mathbf{\Theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}; \mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix} \quad (7)$$

Entonces, se tiene un vector de salida real del sistema \mathbf{Y} de N filas, una matriz de funciones conocidas $\mathbf{\Phi}$ de N filas y n columnas, un vector de parámetros $\mathbf{\Theta}$ de n filas y por último un vector de error \mathbf{e} de N filas.

Teniendo en cuenta esta notación, se puede bautizar y definir al índice mediante el símbolo E y la (8) respectivamente.

$$E = \sum_{k=1}^N e^2(k) = \mathbf{e}^T \mathbf{e} = (\mathbf{Y} - \mathbf{\Phi}\mathbf{\Theta})^T (\mathbf{Y} - \mathbf{\Phi}\mathbf{\Theta}) \quad (8)$$

Si ahora se deriva al índice respecto de los parámetros y se iguala a cero dicha expresión, se puede deducir el valor de los parámetros que minimizan dicho índice con ayuda de reglas de derivación de matrices. Esto es lo que se presenta en la (9).

$$\begin{aligned} \frac{\partial V}{\partial \mathbf{\Theta}} &= \frac{\partial}{\partial \mathbf{\Theta}} [(\mathbf{Y} - \mathbf{\Phi}\mathbf{\Theta})^T (\mathbf{Y} - \mathbf{\Phi}\mathbf{\Theta})] = \frac{\partial}{\partial \mathbf{\Theta}} [\mathbf{Y}^T \mathbf{Y} - \mathbf{Y}^T \mathbf{\Phi}\mathbf{\Theta} - \mathbf{\Theta}^T \mathbf{\Phi}^T \mathbf{Y} + \mathbf{\Theta}^T \mathbf{\Phi}^T \mathbf{\Phi}\mathbf{\Theta}] \\ \frac{\partial V}{\partial \mathbf{\Theta}} &= -\mathbf{\Phi}^T \mathbf{Y} - \mathbf{\Phi}^T \mathbf{Y} + \mathbf{\Phi}^T \mathbf{\Phi}\mathbf{\Theta} + \mathbf{\Phi}^T \mathbf{\Phi}\mathbf{\Theta} = -2\mathbf{\Phi}^T \mathbf{Y} + 2\mathbf{\Phi}^T \mathbf{\Phi}\mathbf{\Theta} \\ -2\mathbf{\Phi}^T \mathbf{Y} + 2\mathbf{\Phi}^T \mathbf{\Phi}\mathbf{\Theta} &= \mathbf{0} \rightarrow \mathbf{\Theta} = (\mathbf{\Phi}^T \mathbf{\Phi})^{-1} \mathbf{\Phi}^T \mathbf{Y} \\ \mathbf{\Theta} &= (\mathbf{\Phi}^T \mathbf{\Phi})^{-1} \mathbf{\Phi}^T \mathbf{Y} \end{aligned} \quad (9)$$

Ahora bien, para implementar esto de forma iterativa se puede considerar la matriz Φ y el vector \mathbf{y} para k pares $\{y(h), x(h)\}$ con $h = 1 \dots k$ y utilizando la notación Φ_k e \mathbf{Y}_k podemos escribir.

$$\mathbf{Y}_k = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix}; \Phi_k = \begin{bmatrix} \varphi_1(x_1) & \varphi_2(x_1) & \dots & \varphi_n(x_1) \\ \varphi_1(x_2) & \varphi_2(x_2) & \dots & \varphi_n(x_2) \\ \vdots & \vdots & & \vdots \\ \varphi_1(x_k) & \varphi_2(x_k) & \dots & \varphi_n(x_k) \end{bmatrix}$$

Al recibir un nuevo par de datos $(y(k+1), x(k+1))$ puede escribirse en la forma:

$$\mathbf{Y}_{k+1} = \begin{bmatrix} \mathbf{y}_k \\ y(k+1) \end{bmatrix}; \Phi_{k+1} = \begin{bmatrix} \Phi_k \\ \varphi_{k+1} \end{bmatrix}$$

Donde: $\varphi_{k+1} = \{\varphi_1(x_{k+1}), \varphi_2(x_{k+1}), \dots, \varphi_n(x_{k+1})\}$. Entonces, la solución al problema de optimización queda definida por la (10).

$$\begin{aligned} \Theta_{k+1} &= (\Phi_{k+1}^T \Phi_{k+1})^{-1} \Phi_{k+1}^T \mathbf{Y}_{k+1} \\ \Theta_{k+1} &= (\Phi_k^T \Phi_k + \varphi_{k+1}^T \varphi_{k+1})^{-1} (\Phi_k^T \mathbf{y}_k + \varphi_{k+1}^T Y(k+1)) \end{aligned} \quad (10)$$

El enfoque teórico de los algoritmos LMS resuelve el problema de optimización enfocándose exclusivamente en disminuir la suma del error cuadrático medio, ECM. Como ya se había hecho mención anteriormente, no basta solo con lograr ese objetivo, sino que los coeficientes a los que converja el algoritmo deben poder representar correctamente al sistema. Es difícil asegurar que los algoritmos LMS no converjan a una solución de un sistema inestable y por ello, se proponen como primera alternativa los algoritmos genéticos.

1.3 Algoritmo Genético

A continuación, se presenta un breve resumen de la referencia [2] respecto a este tema. Estos algoritmos son métodos de optimización metaheurísticos, también conocidos como técnicas estocásticas o probabilísticas. Fueron introducidos por Holland [3] y desarrollados por Goldberg [4], estos algoritmos emplean una heurística basada en la teoría de la evolución genética de Charles Darwin [5]. Comienzan con una población inicial de soluciones potenciales, que se mejora iterativamente. Cada iteración, llamada generación, busca mejorar el desempeño de sus predecesores.

Cada solución posible se denomina individuo, y está compuesta por genes que representan las variables del problema a resolver, en este caso son los coeficientes del polinomio en la función transferencia de la planta que se busca modelar. La eficacia de cada solución se evalúa mediante la función objetivo que se desea optimizar. Para este caso, dicha función objetivo es la suma del ECM junto con la distancia de los polos al origen del plano Z, para asegurar estabilidad, y la verificación de que los coeficientes del numerador no sean negativos.

Durante cada ciclo del algoritmo, se generan nuevos individuos mediante operaciones genéticas como selección, cruce y mutación. La operación de selección consiste, en primer lugar, tomar

aleatoriamente 3 individuos y seleccionar al mejor de ellos para participar en la siguiente operación, esto también bajo una cierta probabilidad. La segunda operación consiste en creación de 2 individuos hijos de 2 individuos padres mediante un promedio entre cada uno de los genes, o en este caso, los coeficientes. Cabe resaltar que se obtienen dos individuos hijos porque dicho promedio se realiza de forma tal que un hijo se parece más a un padre que al otro. La última operación es la de mutación, la cual consiste en generar una alteración al individuo bajo cierta probabilidad, en este caso lo que se realiza es sumar un pequeño valor a cada coeficiente de forma aleatoria. Estos son los mecanismos que le permiten al algoritmo recorrer el hiperespacio de soluciones para hallar una solución lo suficientemente buena.

1.4 Búsqueda Tabú

El algoritmo de búsqueda Tabú es uno de los métodos basados en trayectoria, dado que el proceso de búsqueda determina una trayectoria en el hiperespacio de soluciones. A continuación, se presenta un resumen de la referencia [6] respecto a este tema.

La idea general detrás de los algoritmos metaheurísticos basados en búsquedas locales es evaluar los puntos en las proximidades del agente de búsqueda, dicho agente sería un individuo o una combinación de los coeficientes, y localizar la mejor solución posible en dicho conjunto. En este punto, el agente se reubicará en la posición recién identificada. Este procedimiento continúa dicha trayectoria hasta que llega a un punto donde no encuentra un mejor movimiento, es decir, encuentra un punto óptimo. El problema radica en que dicho punto sea un óptimo local y no uno global, por ello, se adapta el algoritmo para que también permita el movimiento a un punto en la proximidad que no sea el mejor. Esto se hace para permitirle el explorar otras regiones del hiperespacio de soluciones. Ese mecanismo es una función similar a la memoria que le permite registrar los movimientos que hizo con anterioridad y evitar repetirlos. Esta última característica es lo que le da el nombre al algoritmo. La lista donde se almacenan los anteriores movimientos o los anteriores puntos que visitó se llama lista Tabú. Recibe ese nombre porque justamente se le prohíbe al algoritmo regresar por dichos puntos.

El algoritmo cuenta con tres etapas: La primera es la de inicialización, donde se ubica de forma aleatoria un punto en el hiperespacio de soluciones. La segunda es la etapa de búsqueda, donde se evalúa cual es el mejor movimiento a realizar y que el mismo no esté dentro de la lista Tabú. Cabe aclarar que el movimiento consiste en desplazar el punto una distancia μ en una de sus direcciones, es decir, incrementar o decrementar solo uno de los coeficientes. Por lo tanto, todos los posibles movimientos se forman a partir de incrementar o decrementar cada uno de los coeficientes, resultando en $2(N_a + N_b)$ posibles movimientos. Por otro lado, es vital que el tamaño de la lista Tabú, representado con la variable δ , sea lo suficientemente grande como para hallar un óptimo global junto con una cantidad suficiente de iteraciones del propio algoritmo, es decir, una cantidad suficiente de movimientos en el hiperespacio. La tercera etapa es la de finalización y consiste básicamente en encontrar una solución que cumpla algún criterio, por ejemplo, haber realizado un número máximo de iteraciones o movimientos en el hiperespacio de soluciones.

2 Adquisición y Preprocesamiento de Señales

Con el objetivo de evaluar el desempeño de los algoritmos en un sistema real, se realizó el siguiente ensayo para obtener la respuesta al escalón de un motor de CC. Las dos señales necesarias para el proceso, el escalón de entrada y la respuesta de la salida del mismo, se obtuvieron de una experiencia de laboratorio, realizada en la asignatura “Sistemas de Control 1”, cuyo el objetivo era modelar sistemas a partir de diferentes métodos clásicos. La experiencia se realizó utilizándose un motor CC Remssi SR42-12200, que se muestra en la en la Fig. 2 donde también se pueden observar los instrumentos y fuentes de alimentación utilizadas para realizar el ensayo.

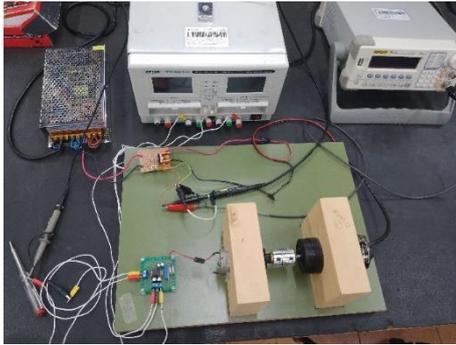


Fig. 2: Conexiones del Ensayo.

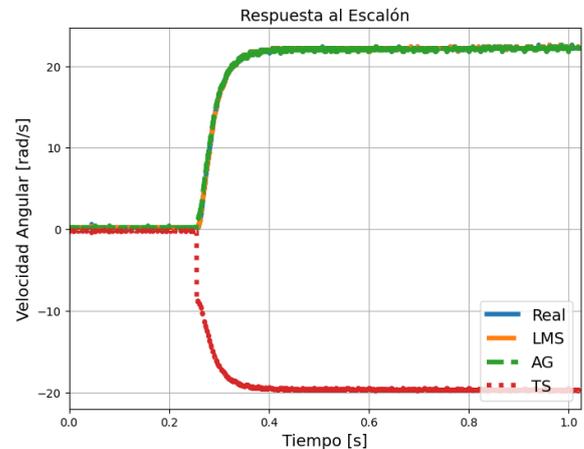


Fig. 3: Respuesta al escalón con dos coeficientes.

3 Análisis de resultados

El desempeño de los algoritmos se evalúa en función de tres métricas. La primera de ellas es la suma del error cuadrático medio que se obtiene con cada sistema que modela cada algoritmo. La segunda consiste en verificar si el sistema que modelaron es estable, en este caso, se indica con un binario de “VERDADERO” si es estable y “FALSO” en caso contrario. La última es el tiempo que tarda cada algoritmo en estimar los coeficientes del sistema. También, resulta interesante observar los valores de los coeficientes a fin de detectar comportamientos erróneos tales como ignorar la señal de entrada, es decir, que los coeficientes a_i tomen valores cercanos a 10^{-2} . La implementación de todos los algoritmos propuestos se realiza en el lenguaje de programación Python [7] y se pueden observar los mismos junto con los resultados que se presentarán a continuación en la referencia [8]. Dicho link lleva a un Notebook de GoogleColab donde se realizó todo el procesamiento de señales.

3.1 Desempeño con 2 Coeficientes: a_0 y b_1

Para el caso de utilizar solo un coeficiente en el numerador y denominador se obtienen los resultados de la Tabla 1 y la Fig. 3. En los mismos se observa que el algoritmo TS tuvo el peor desempeño dado el gran valor de ECM que obtuvo sumado a que la forma de la señal de salida presenta un valor de régimen estacionario menor al real, y además resulta invertida. Respecto a AG y LMS, presentaron un muy buen comportamiento, dado el bajo valor de ECM, pero si se observan los valores de los coeficientes se puede apreciar que simplemente están copiando la salida anterior, debido a que el valor de b_1 es cercano a la unidad, y que ignoran la señal de entrada, dado que el valor de a_0 es despreciable. Dicho de otra forma, la salida estimada actual se está igualando a la salida real atrasada en una muestra, como se expresa en al (12). Esto último significa que el modelo al que converge los algoritmos no presenta una relación entre su señal de entrada y salida, como se expresa

en la (12).De todas formas, todas las soluciones propuestas presentan un modelo de sistema estable y el AG fue el algoritmo que más tiempo demoró en dar sus respectivos coeficientes.

$$\hat{y}(k) = a_0u(kT) + b_1y((k - 1)T) \cong y((k - 1)T) \quad (11)$$

$$G_P(z) = \frac{Y(z)}{U(z)} = \frac{a_0z^{N_a}}{z^{N_b} + b_1z^{N_b-1}} \cong \frac{0}{z^{N_b} + b_1z^{N_b-1}} \cong 0 \quad (12)$$

Tabla 1: Resultados con dos coeficientes.

Función	Coeficientes				ECM	Estabilidad	Tiempo (s)
	a_0	a_1	b_1	b_2			
TS	-0,7313	0,6949	0,3275	-0,4899	253,0803	VERDADERO	29,06
AG	0,3275	0,0160	-0,5307	-0,2788	0,3538	VERDADERO	308,20
LMS	0,0000	0,0132	-0,5268	-0,4662	0,0024	VERDADERO	5,38

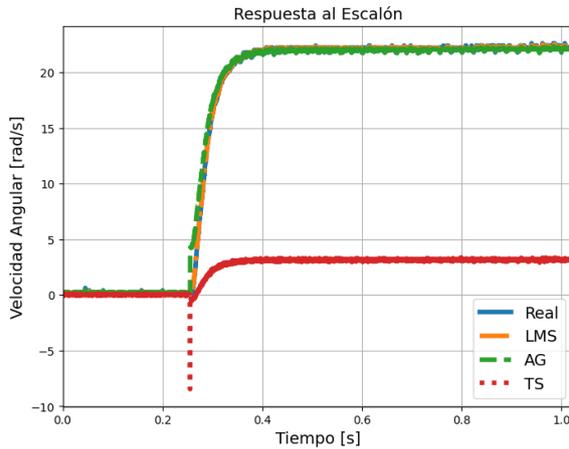


Fig. 4: Respuesta al escalón con cuatro coeficientes.

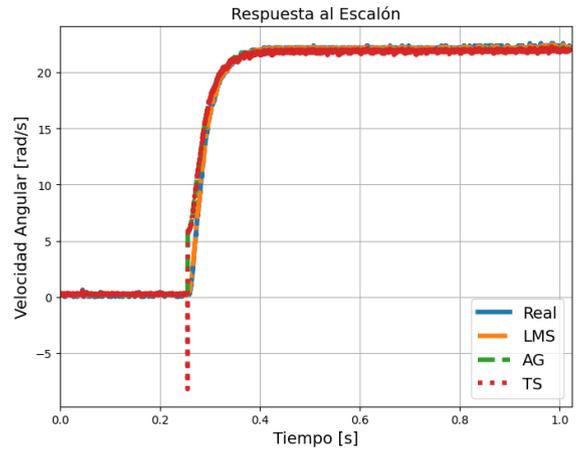


Fig. 5: Respuesta al escalón con cuatro coeficientes.

3.2 Desempeño con 4 Coeficientes: a_0 , a_1 , b_1 y b_2

Los resultados obtenidos con este caso se pueden observar en la Tabla 2 y la Fig. 4. En los mismos se observa nuevamente que el algoritmo TS tuvo el peor desempeño dado el gran valor de ECM que obtuvo junto con que ahora la señal de salida parece solo estar atenuada. Respecto a LMS, pareciera que funcionó, dado el bajo valor de ECM, pero si se observa los valores de los coeficientes, se puede apreciar que otra vez simplemente están copiando la salida anterior, debido a que el valor de la suma de b_1 y b_2 es cercano a la unidad, y que ignoran la señal de entrada, dado que el valor de a_0 y a_1 es despreciable. En cambio, El AG tiene características similares, pero no está ignorando la señal de entrada, dado que a_0 no tiene un valor despreciable y a_1 sí. Además, que la suma de los coeficientes del denominador no se acerca tanto a la unidad. Igual que en el caso anterior, todas las soluciones propuestas presentan un modelo de sistema estable y el AG fue el que más tiempo demoró en dar sus respectivos coeficientes.

Tabla 2: Resultados con cuatro coeficientes.

Función	Coeficientes				ECM	Estabilidad	Tiempo (s)
	a_0	a_1	b_1	b_2			
TS	-0,7313	0,6949	0,3275	-0,4899	253,0803	VERDADERO	29,06
AG	0,3275	0,0160	-0,5307	-0,2788	0,3538	VERDADERO	308,20
LMS	0,0000	0,0132	-0,5268	-0,4662	0,0024	VERDADERO	5,38

3.3 Desempeño con 6 Coeficientes, a_0 , a_1 , a_2 , b_1 , b_2 y b_3

En el último caso, los resultados obtenidos utilizando 6 coeficientes se pueden observar en la Tabla 3 y la Fig. 5. En los mismos se observa otra vez que, el algoritmo TS tuvo el peor desempeño, dado que el valor de ECM fue levemente superior al obtenido por AG. La señal de salida estimada por TS ahora logra aproximarse bastante a la señal de salida real, durante gran parte del transitorio y del régimen estacionario, aunque ni bien empieza a crecer la señal de salida real se observa que la estimada toma valores negativos. Respecto a LMS, se observa el mismo comportamiento que en los anteriores casos, obtiene un bajo valor de ECM pero se está ignorando la señal de entrada, dado que los coeficientes a_i toman valores despreciables, y simplemente se está replicando la salida real del sistema, porque la suma de los coeficientes b_i es cercana a la unidad. Igual que el caso anterior, el AG tiene características similares, pero no está ignorando la señal de entrada, dado que a_1 y a_2 no tiene un valor despreciable y a_0 sí. Además, la suma de los coeficientes del denominador no se acerca tanto a la unidad. Igual que en el caso anterior, todas las soluciones propuestas presentan un modelo de sistema estable y el AG fue el que más tiempo demoró en dar sus respectivos coeficientes.

Tabla 3: Resultados con cuatro coeficientes.

Función	Coeficientes						ECM	Estabilidad	Tiempo (s)
	a_0	a_1	a_2	b_1	b_2	b_3			
TS	-0,7313	0,6949	0,5175	-0,6499	-0,0091	-0,0710	0,7071	VERDADERO	57,45
AG	0,0230	0,3445	0,1055	-0,3226	-0,4144	-0,0004	0,6660	VERDADERO	453,05
LMS	-0,0039	0,0055	0,0188	-0,3887	-0,3340	-0,2665	0,0034	VERDADERO	4,09

4 Conclusiones

El desarrollo de este trabajo brinda las siguientes conclusiones:

1. Los tres algoritmos propuestos demuestran que son capaces de resolver el problema de optimización desde un punto de vista teórico, es decir, simplemente enfocándose en reducir la suma del error cuadrático medio.
2. Existe más de una solución o más de una combinación de coeficientes que logran resolver el problema. Por lo tanto, se puede afirmar que dentro del hiperespacio de soluciones hay más de un punto crítico que se acerca lo suficiente a una solución aceptable.
3. Dado que el problema de optimización es sobre el modelado de sistemas, no sirve cualquier solución o cualquier combinación de coeficientes.
4. El algoritmo de mínimos cuadrados nunca logra converger a una solución útil del sistema analizado dado que converge en la mejor solución para reducir el ECM. Dicha solución es la de ignorar la señal de entrada del sistema y replicar el valor anterior de la salida del

- sistema. Para el uso que se pretende dar al algoritmo, no es útil que el mismo converja a un modelo de sistema que no sigue el principio de relación causa-efecto. Por ello, el algoritmo LMS no será útil para modelar sistemas que representen la dinámica de un motor de CC.
5. El algoritmo de búsqueda Tabú, bajo parámetros idénticos a los del algoritmo genético, obtuvo los peores resultados en la suma del error cuadrático medio, aunque demoró menos en dar sus respectivos coeficientes a comparación con el algoritmo genético. Como este último demostró ser más robusto, en el sentido de obtener una solución útil independientemente de la cantidad de coeficientes, se lo elige como mejor propuesta.

Bibliografía

- [1] A. N. Reviriego, C. P. Cámara y J. M. M. Sánchez, Conceptos Básicos de Filtrado, Estimación e Identificación., Primera ed., Madrid: Universidad Nacional de Educación a Distancia, 2014.
- [2] D. G. Reina, T. A. Córdoba y Á. Rodríguez del Nozal, Algoritmos Genéticos con PYTHON, Primera ed., Akoma: Marcombo, 2020.
- [3] J. H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, 1992.
- [4] D. E. GOLDBERG, Genetic algorithms in search, optimization, and machine learning, Boston, 1989.
- [5] C. Darwin, The origin of species, New York, 1909.
- [6] Z. A. Babak, COMPUTATIONAL INTELLIGENCE-BASED OPTIMIZATION ALGORITHMS, Primera ed., Boca Raton, Florida: CRC Press, 2024.
- [7] P. S. Foundation, «Python Language Reference, version 3.12.4,» [En línea]. Available: <https://www.python.org/>. [Último acceso: 20 07 2024].
- [8] J. J. Berger, «Comparación de algoritmos de optimización. GoogleColab,» [En línea]. Available: https://colab.research.google.com/drive/1rDU_cGxGpd6JwKMzF639PiB1__eRIIPG?usp=sharing. [Último acceso: 25 07 2024].