PEM Grupo 3: Gradient Descend (Fletcher-Reeves)

Elaborado por:

Federico Banoy Restrepo

Juan David Rengifo Castro

Salomón Cardeño Luján

Problema

Implementar en MatLab el método de optimización seleccionado para una estructura arx111 $\left(G(z) = \frac{z\, b_1}{z + a_1}\right)$.

Graficar la convergencia y recorrido del algoritmo excitándolo con una señal PRBS y aplicando distintos niveles de ruido.

Grandiente descente

El **gradiente descendente** hace parte de los métodos de primer orden, los cuales emplean información sobre el gradiente de la **función de costo** $\nabla V(\boldsymbol{x}) = \frac{\partial}{\partial \boldsymbol{x}} V(\boldsymbol{x})$.

Este método se basa en la idea de iterativamente, *avanzar en la dirección de máxima reducción de la función de costo*. Para calcular dicha dirección se suele emplear el **vector de búsqueda,** denotado por $s(k)$, se define como el inverso aditivo del gradiente de la función de costos normalizado en norma euclidea.

$$\boldsymbol{s}(k) = -\frac{\nabla V(\boldsymbol{x}(k))}{\|\nabla V(\boldsymbol{x}(k))\|_2}$$

Así las cosas, el algoritmo considera la información de la iteración anterior para mejorar la estimación, tal y como se muestra a continuación

$$\boldsymbol{x}(k+1) = \boldsymbol{x}(k) + \alpha \cdot \boldsymbol{s}(k); \quad \boldsymbol{x}(0) = \boldsymbol{x}_0,$$

donde α es el *tamaño de paso* y x_0 las *condiciones iniciales* . Este parámetro es fundamental para el desempeño del método puesto que regula la velocidad de cambio en cada iteración. Un tamaño de paso muy pequeño es más preciso, pero puede resultar muy costoso en tiempo de computo. Por el contrario un tamaño de paso muy grande será muy eficiente en tiempo pero puede generar estimaciones muy poco precisas. Por lo tanto, elección de este parámetro no debe tomarse a la ligera. Cabe resaltar, que existen múltiples métodos para escoger un α apropiado y puede ser fijo o variable.

Fletcher-Reeves

En aras de lograr una buena estimación con una mejor convergencia, Fletcher & Reeves (1964) propusieron una modificación al gradiente descendente tradicional mediante el uso de gradientes conjugados, alcanzando así una convergencia cuadrática para funciones objetivos cuadráticas, tales como el error cuadrático medio (MSE).

El método modifica el vector de búsqueda como sigue:

$$\boldsymbol{s}(k) = -\nabla V(\boldsymbol{x}(k)) + \beta(k)\boldsymbol{s}(k-1),$$

donde,

$$\beta(k) = \frac{[\nabla V(\boldsymbol{x}(k))]^T[\nabla V(\boldsymbol{x}(k))]}{[\nabla V(\boldsymbol{x}(k-1))]^T[\nabla V(\boldsymbol{x}(k-1))]}$$

Antes de proceder a la implementación, vale la pena hacer algunos comentarios.

1. Se sugiere emplear un método para escoger valores iniciales adecuados.
2. El algoritmo es sensible a impresiciones numéricas, por lo tanto es recomendable reiniciar el algoritmo durante la optimización.
3. Es recomendable escoger un número de iteraciones máximas apropiado.

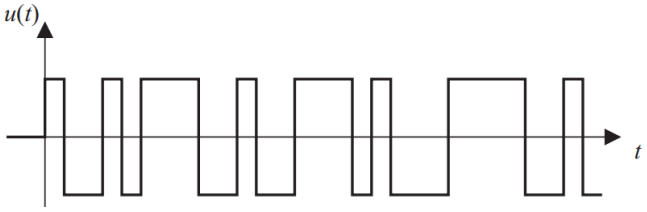
Lamentablemente, dadas las restricciones de tiempo, no se implementaron algoritmos para la elección de parámetros adecuados.

Nota: Una sucesión converge con orden q a ϵ si $\lim_{k\longrightarrow\infty} \frac{|x_{k+1} - \varepsilon|}{|x_k - \varepsilon|^q} = \mu > 0.$

Metodología

¿Qué tenemos?

1. Función de transferencia discreta $G(z) = \frac{z\, b_1}{z + a_1}$ asociada a un ARX110.
2. Entrada de señal discreta binaria PRBS.



Plan

1. Definición explicita del modelo.
2. Cálculo analítico del vector de busqueda
3. Implementación.
4. Análisis de convergencia.

Solución

Modelo

La estructura de un modelo arx110 ($a, b,$ exponente de q en el numerador) es la siguiente

$$y(k) = \frac{b_1 q^{-0}}{1 + a_1 q^{-1}} u(k) + \frac{1}{1 + a_1 q^{-1}} e(k)$$

$$y(k) + a_1 y(k-1) = b_1 u(k) + e(k)$$

$$y(k) = -a_1 y(k-1) + b_1 u(k) + e(k)$$

Estimación del error (e en el código)

De la expresión anterior, el error $e(t)$ se puede estimar como

$$e(k) = y(k) + a_1 y(k-1) - b_1 u(k)$$

Gradiente de la función de costo (S1 y S2 en el código)

Sabemos que la función de costo en nuestro caso es $V(\theta) = \frac{1}{N}\sum_{t=1}^{N} e^2(t)$, lo cual implica que

$$\frac{\partial V}{\partial a_1} = \frac{2}{N}\sum_{t=1}^{N} e(t)\frac{\partial e(t)}{\partial a_1}, \qquad \frac{\partial V}{\partial b_1} = \frac{2}{N}\sum_{t=1}^{N} e(t)\frac{\partial e(t)}{\partial b_1}$$

Luego, como $\frac{\partial e(t)}{\partial a_1} = y(t-1)$ y $\frac{\partial e(t)}{\partial b_1} = -u(t)$, entonces

$$\frac{\partial V}{\partial a_1} = \frac{2}{N}\sum_{t=1}^{N} e(t)y(t-1) = \frac{2}{N}S_1(k), \qquad \frac{\partial V}{\partial b_1} = -\frac{2}{N}\sum_{t=1}^{N} e(t)u(t) = -\frac{2}{N}S_2(k)$$

$$\nabla V(\boldsymbol{x}(k)) = \left[\frac{2}{N}S_1(k), -\frac{2}{N}S_2(k)\right]^T.$$

Estimación de los parámetros a_1 y b_1

Realizamos la estimación de los parámetros del modelos, es decir, $\theta = [a_1, b_1]$.

Iteración 1

Para la primera iteración empleamos el gradiente descendentes básico, es decir,

$$a_1(k+1) = a_1(k) + \alpha_k s_a(k), \text{ donde } s_a(k) = -\frac{\partial V}{\partial a_1}$$

$$b_1(k+1) = b_1(k) + \alpha_k s_b(k), \text{ donde } s_b(k) = -\frac{\partial V}{\partial b_1}.$$

Iteración 2 en adelante

Desde la segunda iteración empleamos la propuesta de Fletcher-Reeves, pues antes no tiene sentido. Veamos cual es el vector de busqueda.

$$a_1(k+1) = a_1(k) + \alpha_k s_a(k)$$

$$b_1(k+1) = b_1(k) + \alpha_k s_b(k)$$

donde:

$$s_a(k) = -\frac{\partial V}{\partial a_1} + \beta(k) s_a(k-1)$$

$$s_b(k) = -\frac{\partial V}{\partial b_1} + \beta(k) s_b(k-1)$$

$$\beta(k) = \frac{\left[\frac{\partial V}{\partial a_1}, \frac{\partial V}{\partial b_1}\right]^T \cdot \left[\frac{\partial V}{\partial a_1}, \frac{\partial V}{\partial b_1}\right]^T}{\left[\frac{\partial V}{\partial a_1}(k-1), \frac{\partial V}{\partial b_1}(k-1)\right]^T \cdot \left[\frac{\partial V}{\partial a_1}(k-1), \frac{\partial V}{\partial b_1}(k-1)\right]^T}$$

$$= \frac{\frac{\partial V}{\partial a_1}^2 + \frac{\partial V}{\partial b_1}^2}{\frac{\partial V}{\partial a_1}(k-1)^2 + \frac{\partial V}{\partial b_1}(k-1)^2} = \frac{\left(\frac{2}{N}\sum_{t=1}^{N} e(t)y(t-1)\right)^2 + \left(-\frac{2}{N}\sum_{t=1}^{N} e(t)u(t)\right)^2}{\left(\frac{2}{N}\sum_{t=1}^{N} e(t,k-1)y(t-1,k-1)\right)^2 + \left(-\frac{2}{N}\sum_{t=1}^{N} e(t,k-1)u(t,k-1)\right)^2}$$

$$= \frac{\left(\sum_{t=1}^{N} e(t)y(t-1)\right)^2 - \left(\sum_{t=1}^{N} e(t)u(t)\right)^2}{\left(\sum_{t=1}^{N} e(t,k-1)y(t-1,k-1)\right)^2 - \left(\sum_{t=1}^{N} e(t,k-1)y(t-1,k-1)\right)^2} = \frac{(S_1(k))^2 - (S_2(k))^2}{(S_1(k-1))^2 - (S_2(k-1))^2}$$

Así tenemos por vector de busqueda,

$$s(k) = [a_1(k) + \alpha(k)s_a(k),\ b_1(k) + \alpha(k)s_b(k)].$$

Tamaño de paso α_k

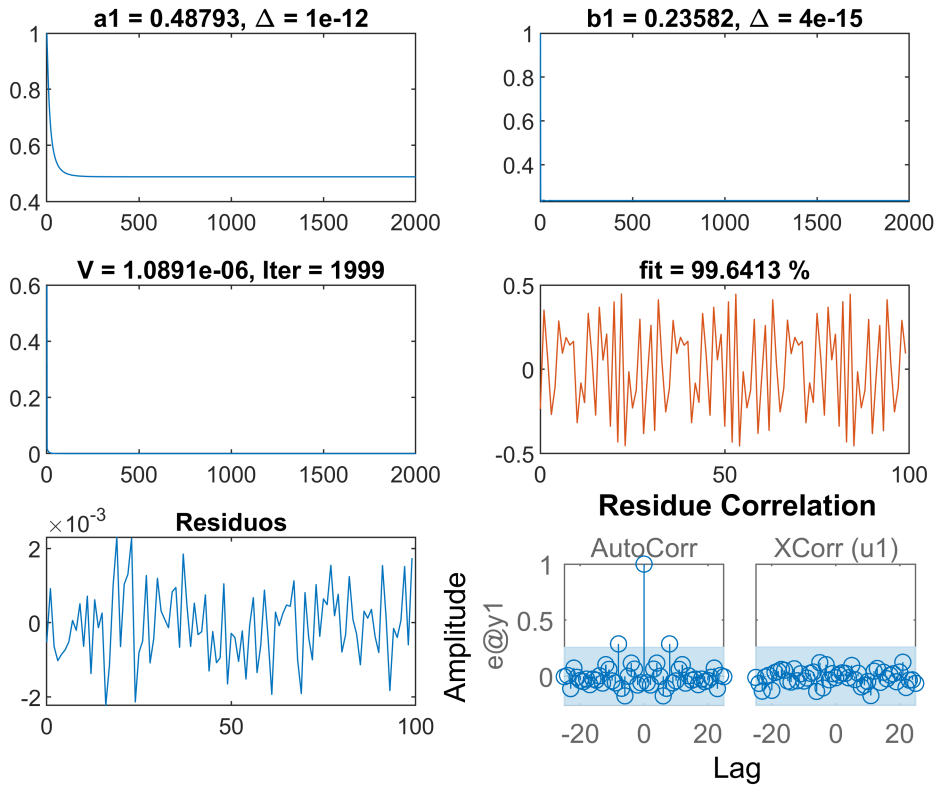
De acuerdo con el análisis de convergencia empleado para el método de gradiente descendente es posible que el tamaño de paso $\alpha_k < \frac{1}{2k}$. En este caso, se considera la elección heurística de $\alpha_k = \frac{1}{2k}$ (aún si se trata de la extensión del método), donde k es la iteración. Esto resulta menos costoso computacionalmente y más simple que utilizar un método de optimización de una dimensión para encontrar el α_k óptimo, lo cual es el procedimiento usual.

Código

```
% Simulación OE
```

```
a1 = 0.4876; b1 = 0.2357;
A_error = 0.001; % Amplitud del error
N = 100; % Número de datos experimentales
z = tf('z', 'Ts', 1); Gd = b1*z/(z+a1); % Modelo
u = 2*(prbs(5,N, [1 0 1 0 0])-0.5); % Entrada experimental PRBS
y = lsim(Gd,u,(0:1:N-1)); % Salida experimental
rng(1); % Para generar siempre la misma secuencia aleatoria
y = y + A_error*randn(N,1); % Señal con ruido (modelo OE de perturbación)

% Método numérico del gradiente descendente (Fletcher-Reeves) con modelo ARX110
M = 2000; % Número máximo de iteraciones
a1 = zeros(1,M); b1 = zeros(1,M); V = zeros(1,M);
a1(1) = 1; % Valor inicial de a1
b1(1) = 1; % Valor inicial de b1
tol = 1e-12; % Criterio de parada según exactitud de los parámetros
Vmin = 1e-12; % Criterio de parada según la función de coste

for i=1:M-1 % Proceso de estimación de los parémetros
    S1 = 0; S2 = 0; V1 = 0;
    S1_ant = 0; S2_ant = 0;
    for j=2:N
        e = y(j) + a1(i)*y(j-1) - b1(i)*u(j); % Predicción del error
        S1 = S1 + e*y(j-1);
        S2 = S2 + e*u(j);
        V1 = V1 + e^2;

        if i >= 2
            e_ant = y(j) + a1(i-1)*y(j-1) - b1(i-1)*u(j); % Predicción del error
            S1_ant = S1_ant + e_ant*y(j-1);
            S2_ant = S2_ant + e_ant*u(j);
        end

    end
    V(i) = V1/N; % Función de coste
    if V(i) < Vmin
        a1 = a1(1:i);
        b1 = b1(1:i);
        V = V(1:i);
        iter = i;
        break
    end
    if i==1
        s_a = -(2*S1/N);
        s_b = -(-2*S2/N);
    else
        beta = (S1^2-S2^2)/(S1_ant^2-S2_ant^2);
        s_a = -(2*S1/N) + beta*s_a;
        s_b = -(-2*S2/N) + beta*s_b;
    end
    alphak = 1/(2*i); % Longitud de paso dinámica
    a1(i+1) = a1(i) + alphak*s_a;
    b1(i+1) = b1(i) + alphak*s_b;
    if ((abs(a1(i+1)-a1(i)) <= tol) && (abs(b1(i+1)-b1(i)) <= tol))
        a1 = a1(1:i+1);
```

```
        b1 = b1(1:i+1);
        V1 = 0;
        iter = i+1;
        for j=2:N
            e = y(j) + a1(i+1)*y(j-1) - b1(i+1)*u(j);
            V1 = V1 + e^2;
        end
        V(i+1) = V1/N;
        V = V(1:i+1);
        break
    end
    iter = M-1;
end

arx110 = z*b1(iter)/(z + a1(iter));
[yest, t] = lsim(arx110,u,(0:1:N-1));
test_data = iddata(y, u', 1);
res = y - yest;
fit = (1 - norm(y - yest)/norm(y - mean(y)) )*100;
figure("Name",'Resultados')
subplot(3,2,2), plot(b1), title([ 'b1 = ' num2str(b1(iter)) ', \Delta = ' num2str( abs( b1(iter
subplot(3,2,1), plot(a1), title([ 'a1 = ' num2str(a1(iter)) ', \Delta = ' num2str( abs( a1(iter
subplot(3,2,3), plot(V), title(['V = ' num2str(V(iter)) , ', Iter = ' num2str(iter)])
subplot(3,2,4), plot(t,y,t,yest), title(['fit = ' num2str(fit) ' %'])
subplot(3,2,5), plot(t,res), title('Residuos')
subplot(3,2,6), resid(test_data, arx110);
```



Bibliografia

FLETCHER, R., & REEVES, C. 1964 Function minimization by conjugate gradients. Comput. J. 7,149-154.

- Link oficial
- Sci-Hub

GORDON, G., & TIBSHIRANI, R. Fall 2012 10-725 Optimization course Lecture 5: Gradient Descent Revisited, School of Computer Science, Carneige Mellon University.

- Link oficial