

Cuestionario Práctica 2

	Apellidos	Nombre
1	Antolinos García	Diego Ismael
2	Ruz Nieto	Andrés

1. Escriba su implementación de *policy iteration*, incluyendo su política inicial (valor de u para cada estado i del canal)

```
N = length(H);
max_iter = 10000;
iter = 0;
sigue = true;

policy = [3;1;7;5;2;8;6;4;3;1;7];

g = [];
P = [];
I = eye(N);
for i = 1:N
    u = policy(i);
    g = [g;Gu{u}(i)];
    P = [P;Pu{u}(i,:)];
end

Jpolicy = (I-P)\g

initial_cost = Jpolicy;
politica_actual = policy;

while sigue

    J = zeros(N,length(U))
    for u = 1:length(U)
        J(:,u) = Gu{u} + Pu{u}*Jpolicy;
    end

    [TJ nuevapolitica] = min(J,[],2);

    g = [];
    P = [];

    for i = 1:N
        u = nuevapolitica(i);
        g = [g;Gu{u}(i)];
        P = [P;Pu{u}(i,:)];
    end

    nuevajpolitica = (I-P)\g;
```

```

if (nuevapolitica==politica_actual)
    politicaoptima = politica_actual;
    sigue = false;
else
    politica_actual = nuevapolitica;
    Jpolicy = nuevajpolitica;
end

iter = iter + 1;
if (iter == max_iter)
    break;
end

end

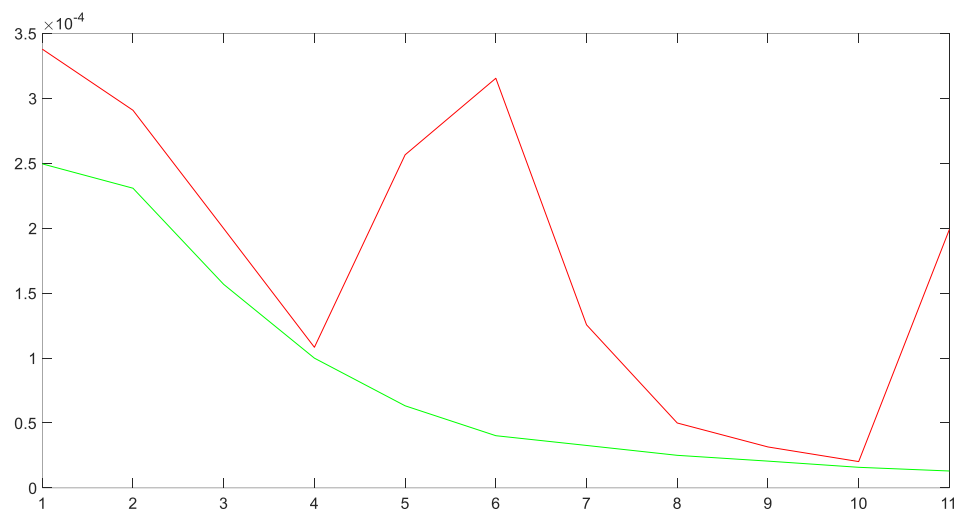
```

2. Escriba la **política óptima** que ha obtenido al ejecutar *policy iteration*. ¿Cuántas iteraciones ha necesitado su algoritmo para encontrar una política óptima?

Hemos necesitado 3 iteraciones

```
politicaoptima = [1;1;6;5;4;3;3;2;2;1;1];
```

3. Represente en una misma gráfica el coste asociado a su política inicial y el **coste óptimo** obtenido con *policy iteration*



Parte avanzada.

```
%datos partida
K = 1.38e-23; %constante de Boltzmann
T = 290; %temperatura de ruido en la antena
F = 12; %figura de ruido del receptor (en dB)
g = 10; %ganancia del transmisor y del receptor (en dB)
d = 1000; %distancia en metros
lP = 1500; %tamaño total de la trama en bits
R = [100000 200000 300000 400000 500000]; %tasa inicial en bits/s
N_o = K*(T+T*(10^(F/10)-1)); %densidad espectral de ruido

vector_p_tx = [4 6 8 10 12 14 16 18]; %vector de posibles potencias
de transmisión en dBm

%matriz de transición del canal radio

Pradio = [0.33 1-0.33 0 0 0 0 0 0 0 0 0;
0.12 1-0.12-0.2 0.2 0 0 0 0 0 0 0;
0 0.15 1-0.15-0.18 0.18 0 0 0 0 0 0;
0 0 0.18 1-0.15-0.18 0.15 0 0 0 0 0;
0 0 0 0.19 1-0.19-0.14 0.14 0 0 0 0;
0 0 0 0 0.2 1-0.2-0.13 0.13 0 0 0;
0 0 0 0 0 0.21 1-0.21-0.12 0.12 0 0;
0 0 0 0 0 0 0.22 1-0.22-0.11 0.11 0;
0 0 0 0 0 0 0 0.23 1-0.23-0.1 0.1 0;
0 0 0 0 0 0 0 0 0.24 1-0.24-0.09 0.09;
0 0 0 0 0 0 0 0 0.2 1-0.2;];

%ganancias del canal en cada estado
H = [-80 -76 -73 -71 -69 -67 -66 -65 -64 -63 -62];

%U conjunto de acciones de control
U = 1:length(vector_p_tx);
Pu = cell(1,length(R)); %conjunto de matrices de markov
Gu = cell(1,length(R)); %conjunto de vectores de coste

for r = 1:length(R)
    Pu_aux = cell(1,length(U));
    Gu_aux = cell(1,length(U));
    for u = U
        p_tx = vector_p_tx(u);
        p_tx_W = 10^(p_tx/10)*0.001; %potencia de transmisión en
W
        vector_p_rx =
(p_tx_W*10^(g/10)*10^(g/10)/d^2).*10.^(H./10); %vector de potencias
recibidas en cada estado
        vector_BER = min(1, 0.5*exp(-
vector_p_rx/(R(r)*N_o))./sqrt(pi()*vector_p_rx/(R(r)*N_o))); %vector de
tasas de error en cada estado
        vector_Pr = (1-vector_BER).^lP; %vector de probabilidad
de recepción de trama en cada estado
        vector_PNrx = (1-vector_Pr)'; %vector de probabilidad de
no recepción de trama en cada estado
        Pu_aux{u} = Pradio.*repmat(vector_PNrx,1,11); %matriz de
probabilidad de transición para el control u
        Gu_aux{u} = (lP*p_tx_W/R(r))*ones(11,1); %matriz de coste
en cada estado para el control u
    end
    Pu{r}=Pu_aux;
```

```

        Gu{r}=Gu_aux;
    end

    %%%%POLICY ITERATION%%%%%%%%

    %Atención MUY IMPORTANTE en un SSP se elimina de la matriz el estado
    de
    %terminación

    N = length(H);
    max_iter = 10000;
    iter = 0;
    sigue = true;

    policy = [3;1;7;5;2;8;6;4;3;1;7]; %inserte aquí su politica inicial
    (vector columna)

    politicas = cell(1,length(R));
    for r = 1:length(R)
        Pu_aux = cell(1,length(U));
        Gu_aux = cell(1,length(U));

        Pu_aux=Pu{r};
        Gu_aux=Gu{r};

        g = [];
        P = [];
        I = eye(N);
        for i = 1:N
            u = policy(i);
            g = [g;Gu_aux{u}(i)];
            P = [P;Pu_aux{u}(i,:)];
        end

        Jpolicy = (I-P)\g%obtenga la J de la politica inicial

        initial_cost = Jpolicy;
        politica_actual = policy;

        %%%%
        % implemente aquí el bucle de policy iteration
        %%%%
        while sigue

            J = zeros(N,length(U),length(R))
            for u = 1:length(U)
                for R = 1:length(R)
                    J(:,u) = Gu_aux{u} + Pu_aux{u}*Jpolicy;
                end
            end

            [TJ nuevapolitica] = min(J,[],2);

            g = [];
            P = [];

            for i = 1:N
                u = nuevapolitica(i);
                g = [g;Gu_aux{u}(i)];
                P = [P;Pu_aux{u}(i,:)];
            end
        end
    end
end

```

```

nuevajpolitica = (I-P)\g;

if (nuevapolitica==politica_actual)
    politicaoptima = politica_actual;
    sigue = false;
else
    politica_actual = nuevapolitica;
    Jpolicy = nuevajpolitica;
end

iter = iter + 1;
if (iter == max_iter)
    break;
end
end
politicas{r} = Jpolicy;
end

%% Represente el coste de la política óptima y la de la su política
inicial
% plot(initial_cost,'r');
hold;
for i=1:length(politicas)
    plot(politicas{i});
end

```

