

# Optimización con fmincon/ga de parámetros de modelos de Simulink

## Table of Contents

Simulemos un control con unos parámetros cualquiera: .....	1
Optimicemos con una búsqueda local a partir de esos parámetros [optimization toolbox] .....	1
Optimicemos con un algoritmo de búsqueda aleatoria dirigida global (algoritmo genético, global optimization toolbox) .....	2

En este fichero, se parte de un sintonizado inicial de un regulador PI y se busca optimizar un índice de coste costeacum sobre esos parámetros. Se compara la salida de fmincon (búsqueda local) con ga (búsqueda global).

Este código función correctamente en Matlab **R2019b**

## Simulemos un control con unos parámetros cualquiera:

```
Kp=0.9; Ki=1.2; %converge a mínimo global
%Kp=6.9; Ki=5.0; %se queda en mínimo local 1.4 con fmincon en caso
    robusto!
%Kp=6.9; Ki=6.0; %se queda en mínimo local 3.2 con fmincon en caso
    robusto!
%Kp=2.7; Ki=1.2; %el óptimo o casi
figure(1)
clf
tic
J0=costeacum([Kp Ki], 'yes');
title('Sintonizado inicial J='+string(J0))
drawnow

CotasMinimas=[0.1 0.1];
CotasMaximas=[8 6];
```

## Optimicemos con una búsqueda local a partir de esos parámetros [optimization toolbox]

```
T0=tic;tic
[KpKi_opt1,J_opt1]=fmincon(@costeacum,[Kp Ki],[[],[],[]],
[],CotasMinimas,CotasMaximas)
T1=toc(T0)

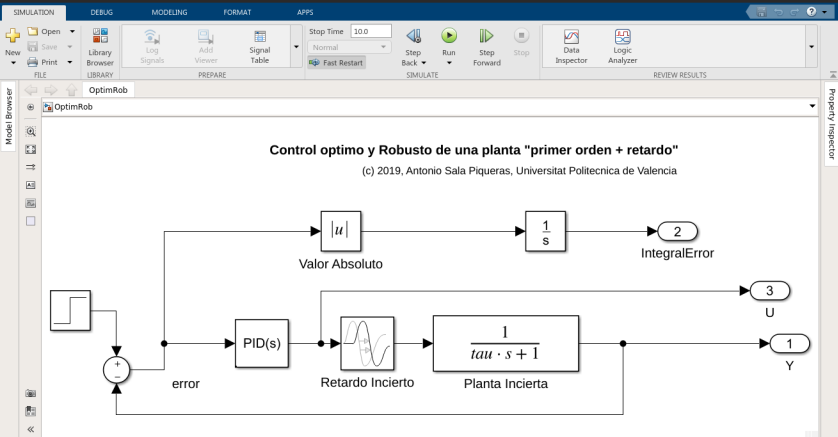
% Grafiquemos un control con los parámetros óptimos de "fmincon"
figure(2)
clf
tic
```

```
J1=costeacum(KpKi_opt1,'yes');  
hold on  
SimulaUnControl(Kp,Ki,'yes','.r');  
hold off  
title('Sintonizado fmincon J='+string(J1))  
drawnow  
  
return
```

## Optimicemos con un algoritmo de búsqueda aleatoria dirigida global (algoritmo genético, global optimization toolbox)

```
opts=optimoptions('ga','MaxTime',239,'FunctionTolerance',5e-5);  
% cuatro minutos máximo tiempo de ejecución (aprox.)  
  
NumParametrosAOptimizar=2;  
  
T0=tic;tic  
[KpKi_opt2,J_opt2]=ga(@costeacum, NumParametrosAOptimizar,[],[],[],  
[],CotasMinimas,CotasMaximas,[],opts)  
T2=toc(T0)  
  
% Ejecutemos un control con los parámetros óptimos de "ga"  
figure(3)  
clf  
tic  
costeacum(KpKi_opt2,'yes');  
hold on  
SimulaUnControl(Kp,Ki,'yes','.r');  
hold off  
title('Sintonizado ga J='+string(J_opt2))  
T1  
T2  
J_opt1  
J_opt2  
KpKi_opt1  
KpKi_opt2
```

*Published with MATLAB® R2019b*



---

```

function simout=SimulaUnControl(Kp, Ki, detalles, LineStyle)

    arguments %Validación de argumentos, Matlab R2019b o superior.
        Kp double
        Ki double
        detalles (1,:) char {mustBeMember(detalles,{'yes','no'})}
    = 'no'
        LineStyle char = 'k'
    end
    model='OptimRob';

    caso='robusto';
    if(strcmp(caso,'robusto'))
        lista_pruebas=diag([2,0.3])*[1.2 1.2 1 0.8 0.8; ...
                                     1.3 0.7 1 1.3 0.7];
    else
        lista_pruebas=[2;.3]; %caso nominal
    end

    NRuns=size(lista_pruebas,2);
    for i=1:NRuns
        in(i)=Simulink.SimulationInput(model);
        in(i)=in(i).setVariable('tau',lista_pruebas(1,i));
        in(i)=in(i).setVariable('d',lista_pruebas(2,i));
        in(i)=in(i).setVariable('Kp',Kp);
        in(i)=in(i).setVariable('Ki',Ki);
    end

    if(strcmp(detalles,'yes'))
        simout=sim(in, 'ShowProgress', 'on');
        for i=1:NRuns
            tmp=simout(i).yout; %SimulationData
            tmpval=tmp{1}.Values;
            subplot(2,1,1)
            hold on
            plot(tmpval,LineStyle,'LineWidth',2)
            tmpval=tmp{3}.Values;
            subplot(2,1,2)
            hold on
            plot(tmpval,LineStyle)
        end
        hold off
        grid on
        ylabel('Acc. Control')
        ylim([0 3.2])
        subplot(2,1,1)
        hold off
        grid on
        ylabel('Salida')
        ylim([0 1.3])
    else
        simout=sim(in, 'ShowProgress', 'off');
    end

```

---

---

```
end  
end
```

*Published with MATLAB® R2019b*

---

```
function J=costeacum(params,detalle)
    arguments %Validación de argumentos, Matlab R2019b o superior.
        params (1,2) double
        detalle (1,:) char {mustBeMember(detalle,{'yes','no'})} = 'no'
    end
    simout=SimulaUnControl(params(1),params(2), detalle);
    NRuns=length(simout);
    cost=zeros(1,NRuns);
    for i=1:NRuns
        salidassimuladas=simout(i).yout; %SimulationData
        IntegralError=salidassimuladas{2}.Values.Data(end);
        Pico=max(salidassimuladas{1}.Values.Data);
        cost(i)=0.5*Pico+0.5*IntegralError;
    end
    J=max(cost);
    fprintf('J=%g [Kp=%g, Ki=%g] -- best: %g, worst: %g -- Time: %g\n', ...
        J,params(1), params(2), min(cost), max(cost),toc);
end
```

*Published with MATLAB® R2019b*