Obtención de características de régimen transitorio. Función **CaracteristicasParam.m**

Félix Monasterio-Huelin

22 de febrero de 2016

Índice

Índice **1**

Índice de Figuras **1**

Índice de Tablas **1**

1. Obtención de características numéricas de régimen transitotio, M_p, t_r, t_p y t_s, de la respuesta al escalón unidad de un sistema lineal dependiente de un único parámetro simbólico. Función CaracteristicasParam.m **2**

2. Ejemplo de utilización de la función CaracteristicasParam con múltipes curvas utilizando leyendas dinámicas **5**

2.1. Programa completo del ejemplo 8

Índice de Figuras

1.1. Salida gráfica del ejemplo de la Tabla 1.3 4

2.1. Ejemplo multicurva utilizando la función **CaracteristicasParam** 5

Índice de Tablas

1.1. Cabecera de la función **CaracteristicasParam** 3

1.2. Salidas de la función **CaracteristicasParam** 3

1.3. Ejemplo: Función **CaracteristicasParam** 4

2.1. Ejemplo multicurva utilizando la función **CaracteristicasParam.m** 8

1. Obtención de características numéricas de régimen transitotio, M_p, t_r, t_p y t_s, de la respuesta al escalón unidad de un sistema lineal dependiente de un único parámetro simbólico. Función CaracteristicasParam.m

En esta Sección se describe la función **CaracteristicasParam.m**. Es una función que recibe entradas simbólicas y genera salidas numéricas.

Esta función obtiene los vectores numéricos de las características de régimen transitorio de la respuesta de un sistema lineal arbitrario a una entrada escalón unidad, en función de un único parámetro que toma valores en un determinado intervalo con saltos discretos prefijados. Estos vectores se obtienen a partir de una representación simbólica del numerador y del denominador de la función de transferencia de tal sistema, pero dependiente de un único parámetro simbólico.

Las caracteríticas que calcula la función **CaracteristicasParam** son la sobreelongación máxima M_p, el tiempo de establecimiento t_s, el tiempo de pico t_p, y el tiempo de respuesta t_r.

La explicación del uso y funcionamiento de la función **CaracteristicasParam.m** se realiza aquí a través de un ejemplo.

Supongamos que se desean obtener gráficas de las características de régimen transitorio de la respuesta al escalón unidad del sistema lineal cuya función de transferencia viene dada por:

$$G(s) = \frac{\omega_n^2}{s^2 + \omega_n s + \omega_n^2} \tag{1.1}$$

Este ejemplo consiste en un sistema de segundo orden en su primera forma canónica con un coeficiente de amortiguamiento $\zeta = 0{,}5$.

El parámetro de variación seleccionado es ω_n. Para este ejemplo se sabe que M_p es independiente de ω_n, y que los tiempos t_p, t_r y t_s son inversamente proporcionales a ω_n,

$$M_p = e^{-\frac{\zeta}{\sqrt{1-\zeta^2}}\pi} \tag{1.2a}$$

$$t_s \approx \frac{\ln\left(\frac{1}{\nu\sqrt{1-\zeta^2}}\right)}{\zeta\omega_n} \tag{1.2b}$$

$$t_p = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}} \tag{1.2c}$$

$$t_p = \frac{\pi - \varphi}{\omega_n\sqrt{1-\zeta^2}} \tag{1.2d}$$

donde ν es la tolerancia, y

$$\sin\varphi = \sqrt{1-\zeta^2} \tag{1.3a}$$

$$\sin\varphi = \zeta \tag{1.3b}$$

Puesto que se va a utilizar como parámetro simbólico ω_n es necesario definirlo como tal, especificando además el nombre que será utilizado en las sustituciones numéricas:

```
syms Omegan
NomParam='Omegan';
```

En segundo lugar, la función de transferencia debe representarse en la forma de polinomios simbólicos del numerador y del denominador de $G(s)$ ya que la función **CaracteristicasParam** utiliza la función de Matlab **step** para realizar una simulación del sistema,

```
NumS=[Omegan^2];
DenS=[1 Omegan Omegan^2];
```

El parámetro libre ω_n, representará el eje de abcisas de las representaciones gráficas que se realicen, por lo que deben especificarse sus límites de variación para la realización de la simulación del sistema:

```
ParamMin=0.1;
ParamMax=20;
incParam=0.1;
ParamLim=[ParamMin, incParam, ParamMax];
```

Para la obtención de la respuesta en el tiempo del sistema $y(t)$, la función **CaracteristicasParam** utiliza la función de Matlab **step** con un tiempo final definido por el usuario,

```
Tfinal=16;
```

Puesto que la función **CaracteristicasParam** calcula el tiempo de establecimiento t_s, es necesario especificar la tolerancia deseada,

```
nuN=0.02;
```

La cabecera de la función **CaracteristicasParam** tiene la forma escrita en la Tabla 1.1:

```
function [err,xParam,MpParam,tsParam,tpParam,trParam]=
         CaracteristicasParam(NomParam,NumS,DenS,nuN,ParamLim,Tfinal)
```

Tabla 1.1: Cabecera de la función **CaracteristicasParam**

La salidas de la función **CaracteristicasParam**, mostradas en la Tabla 1.2, son vectores numéricos de longitud,

$$\text{length(xParam)} = \frac{ParamMax - ParamMin}{incParam} + 1 \tag{1.4}$$

```
Salidas:
        xParam: vector de variación de Param
        MpParam: vector Mp para el intervalo de valores de Param
        tsParam: vector ts para el intervalo de valores de Param
        tpParam: vector tp para el intervalo de valores de Param
        trParam: vector tr para el intervalo de valores de Param
```

Tabla 1.2: Salidas de la función **CaracteristicasParam**

A partir de estos vectores es posible realizar gráficas utilizando, por ejemplo, la función plot.

```
[err,xParam,MpParam,tsParam,tpParam,trParam]=
 CaracteristicasParam(NomParam,NumS,DenS,nuN,ParamLim,Tfinal);
figure
plot(xParam,MpParam,'b');
hold on
plot(xParam,tpParam,'r');
plot(xParam,trParam,'g');
plot(xParam,tsParam,'m');
legend('Mp','tp','tr','ts')
xlabel('\omega_n')
title('Respuesta al escalón unidad')
```

En la Tabla 1.3 está el conjunto de instrucciones del ejemplo anterior para ejecutar en el entorno de Matlab.

```
syms Omegan
NomParam='Omegan';
NumS=[Omegan^2];
DenS=[1 Omegan Omegan^2];
nuN=0.02;
ParamLim=[0.1 0.1 20];
Tfinal=16;
[err,xParam,MpParam,tsParam,tpParam,trParam]=
 CaracteristicasParam(NomParam,NumS,DenS,nuN,ParamLim,Tfinal);
figure
plot(xParam,MpParam,'b');
hold on
plot(xParam,tpParam,'r');
plot(xParam,trParam,'g');
plot(xParam,tsParam,'m');
legend('Mp','tp','tr','ts')
xlabel('\omega_n')
title('Respuesta al escalón unidad')
```

Tabla 1.3: Ejemplo: Función **CaracteristicasParam**

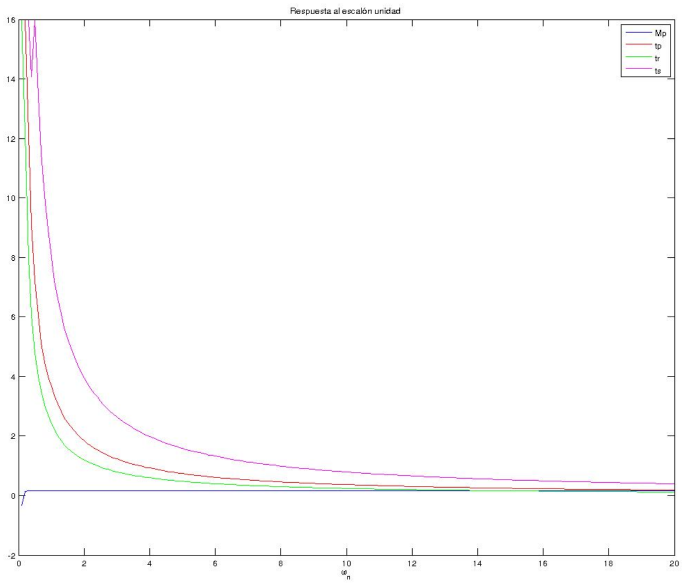


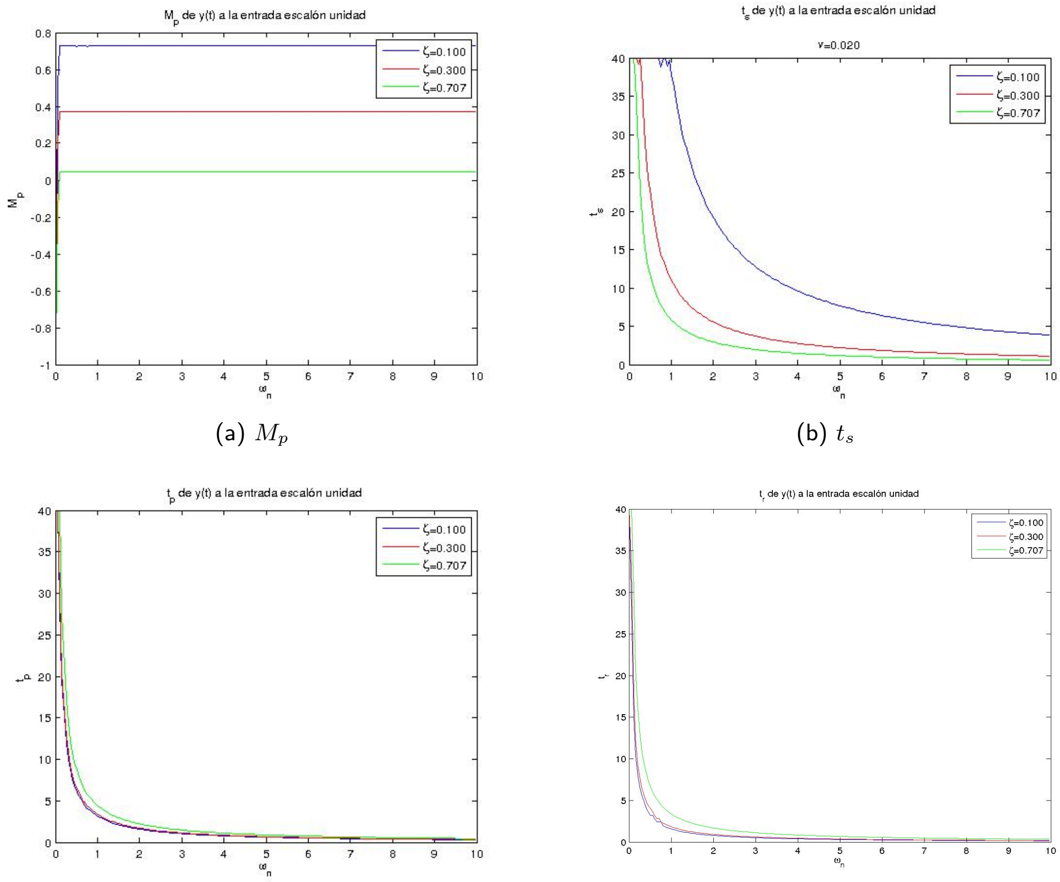
Figura 1.1: Salida gráfica del ejemplo de la Tabla 1.3

En la Figura 1.1 se muestra el resultado de ejecutar las anteriores instrucciones. Podemos apreciar que, como era de esperar, $t_r < t_p < t_s$ para el mismo valor de ω_n, además de que son funciones decrecientes en ω_n. También se aprecia que M_p es independiente de ω_n.

2. Ejemplo de utilización de la función CaracteristicasParam con múltipes curvas utilizando leyendas dinámicas

En esta Sección se utiliza la función **CaracteristicasParam** descrita en la Sección 1 para realizar un conjunto de figuras. Cada una de ellas presenta las curvas de cada característica de régimen transitorio, M_p, t_p, t_r, t_s, de la respuesta al escalón de un sistema lineal para diferentes valores de un parámetro numérico. El eje de abcisas es el parámetro simbólico que requiere la función **CaracteristicasParam.m**. Cada figura lleva asociada una leyenda creada dinámicamente, que indica a qué valor del parámetro numérico se corresponde cada curva.

El programa completo se muestra en la subsección 2.1, pero se va a describir aquí paso a paso. En la Figura 2.1 se muestran los resultados del ejemplo que se describe a continuación.



(a) M_p (b) t_s

(c) t_p (d) t_r

Figura 2.1: Ejemplo multicurva utilizando la función **CaracteristicasParam**

La función de transferencia que se va estudiar tiene la forma,

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{2.1}$$

En primer lugar, debe indicarse que ζ y ω_n son variables simbólicas, y puesto que se utilizará la función **CaracteristicasParam.m** es necesario definir el nombre del parámetro simbólico que representará el eje de abcisas de las representaciones gráficas,

```
syms Zeta Omegan
NomParam='Omegan';
NomParamB=strcat('\omega_n');
```

Con estas declaraciones ya puede definirse la función de transferencia en la forma de dos polinomios,

```
        NumS=[Omegan^2];
        DenS=[1 2*Zeta*Omegan Omegan^2];
```

Cada Figura presentará una leyenda creada dinámicamente con un conjunto de valores de ζ predefinido. Esto puede hacerse creando una lista de valores de ζ. Conviene también crear una lista con los colores que se desea que tenga cada curva. La utilización de listas facilita la programación, ya que permite realizar iteraciones con la función de Matlab **for**,

```
        ListaZeta=[0.1,0.3,0.707];
        ListaColorZeta=['brg'];
```

Se crearán cuatro figuras, cada una de ellas asociada a una característica de régimen transitorio. Es necesario distinguir cada una de ellas, lo que puede hacerse asociando un "handler" distinto. Puesto que se hará un programa iterativo, conviene crear una lista ordenada de los identificadores de cada figura. Además, cada figura tendrá su propio título, por lo que, de nuevo, conviene crear una lista ordenada de nombres que las distinga,

```
    hMp=figure;
    hts=figure;
    htp=figure;
    htr=figure;
    ListaHandlers=[hMp,hts,htp,htr];
    ListaNomCurva={'M_p','t_s','t_p','t_r'};
```

La función **CaracteristicasParam** solo permite como entrada polinomios dependientes de un único parámetro simbólico, por lo que deberá crearse un bucle que genere un valor de ζ que permita hacer una sustitución numérica en cada polinomio. Conviene, también, crearse una lista de vectores de datos para ser utilizada en cada una de las figuras,

```
        ........
        ........
    for Zeta=ListaZeta
        NumSA=subs(NumS,'Zeta',Zeta);
        DenSA=subs(DenS,'Zeta',Zeta);
        [err,xParam,MpParam,tsParam,tpParam,trParam]=
          CaracteristicasParam(NomParam,NumSA,DenSA,nuN,ParamLim,Tfinal);
        ListaVectorFigura=[MpParam', tsParam', tpParam', trParam'];
        ........
        ........
     end% for Zeta
```

Todas las figuras tendrán la misma leyenda creada dinámicamente, con el formato $\zeta =$ al valor concreto de ζ de la lista creada anteriormente, con el color predefinido en la lista de colores creada anteriormente.

```
        ........
        ........
    LeyendaZeta=strcat('\zeta=');
    ind=0;
    for Zeta=ListaZeta
        ........
        ........
        ind=ind+1;
        ColorCurva=ListaColorZeta(ind);
```

```
        ZetaS=num2str(Zeta,'%10.3f');
        Leyenda=strcat(LeyendaZeta,ZetaS);
        for i=1:4
           figure(ListaHandlers(i))
           plot(xParam,ListaVectorFigura(:,i)','Color',ColorCurva,
                'DisplayName',Leyenda)
                ........
                ........
        end% for i
    end% for Zeta
```

Para la creación de la leyenda dinámica es necesario utilizar un recurso de Matlab que por desgracia no está bien documentado, **legend('-DynamicLegend')**. Nos interesa también generar un título distinto para cada figura, así como los nombres que deben aparecer en los ejes de las graficas. Se ha tenido en cuenta que el título para la figura de t_s debe contener el valor de la tolerancia ν ya que el valor de t_s depende de ella. Toda esta información es necesario crearla una sola vez, ya que va asociada a cada figura.

```
        ........
        ........
   nuS=num2str(nuN,'%10.3f');
   TituloNuTs=strcat('\nu=',nuS);
   ind=0;
   for Zeta=ListaZeta
        ........
        ........
        ind=ind+1;
        ........
        ........
        for i=1:4
                ........
                ........
           if ind==1
              NomCurva=ListaNomCurva{i};
              TituloA=strcat(NomCurva,' de y(t) a la entrada escalón unidad');
                 if i==2 %ts
                   Titulo={TituloA,'    ',TituloNuTs};
                 else
                   Titulo={TituloA};
                 end
              xlabel(NomParamB)
              ylabel(NomCurva)
              title(Titulo)
              legend('-DynamicLegend')
              hold all
           end% if ind
        end% for i
   end% for Zeta
```

2.1. Programa completo del ejemplo

```
function CaracteristicasParamEjemplo
%%%%
delete(findall(0,'Type','figure')) % cierra todas las figuras
%%%%
   ListaZeta=[0.1,0.3,0.707];
   ListaColorZeta=['brgmk'];
   Tfinal=40;
   nuN=0.02;
   ParamMin=0.01;
   ParamMax=10;
   incParam=0.05;
   ParamLim=[ParamMin, incParam, ParamMax];
   syms Zeta Omegan
   NomParam='Omegan';
   NomParamB=strcat('\omega_n');
   NumS=[Omegan^2];
   DenS=[1 2*Zeta*Omegan Omegan^2];
   hMp=figure;
   hts=figure;
   htp=figure;
   htr=figure;
   ListaHandlers=[hMp,hts,htp,htr];
   ListaNomCurva={'M_p','t_s','t_p','t_r'};
   nuS=num2str(nuN,'%10.3f');
   TituloNuTs=strcat('\nu=',nuS);
   LeyendaZeta=strcat('\zeta=');
   ind=0;
   for Zeta=ListaZeta
        NumSA=subs(NumS,'Zeta',Zeta);
        DenSA=subs(DenS,'Zeta',Zeta);
        [err,xParam,MpParam,tsParam,tpParam,trParam]=
           CaracteristicasParam(NomParam,NumSA,DenSA,nuN,ParamLim,Tfinal);
        ListaVectorFigura=[MpParam', tsParam', tpParam', trParam'];
        ind=ind+1;
        ColorCurva=ListaColorZeta(ind);
        ZetaS=num2str(Zeta,'%10.3f');
        Leyenda=strcat(LeyendaZeta,ZetaS);
        for i=1:4
           figure(ListaHandlers(i));
           plot(xParam,ListaVectorFigura(:,i)','Color',ColorCurva,
                'DisplayName',Leyenda)
           if ind==1
                 NomCurva=ListaNomCurva{i};
                 TituloA=strcat(NomCurva,' de y(t) a la entrada escalón unidad');
                 if i==2 %ts
                   Titulo={TituloA,'    ',TituloNuTs};
                 else
                   Titulo={TituloA};
                 end
                 xlabel(NomParamB)
                 ylabel(NomCurva)
                 title(Titulo)
                 legend('-DynamicLegend');
                 hold all
           end
        end
   end
end
```

Tabla 2.1: Ejemplo multicurva utilizando la función **CaracteristicasParam.m**