Identificación "no paramétrica" de respuestas ante impulso de un sistema lineal invariante en el tiempo

Presentación en vídeo: http://personales.upv.es/asala/YT/V/impulid.html

Este código funcionó correctamente con Matlab **R2020b**

Objetivo: Identificar directamente la respuesta ante impulso (la respuesta ante la entrada $\{1, 0, 0, \dots\}$, esto es, una cierta secuencia $\{h_0, h_1, h_2, \dots\}$) de un sistema lineal discreto invariante en el tiempo, usando la fórmula de *convolución*.

Tabla de Contenidos

1.- Modelado y simulación para conseguir datos de entrenamiento

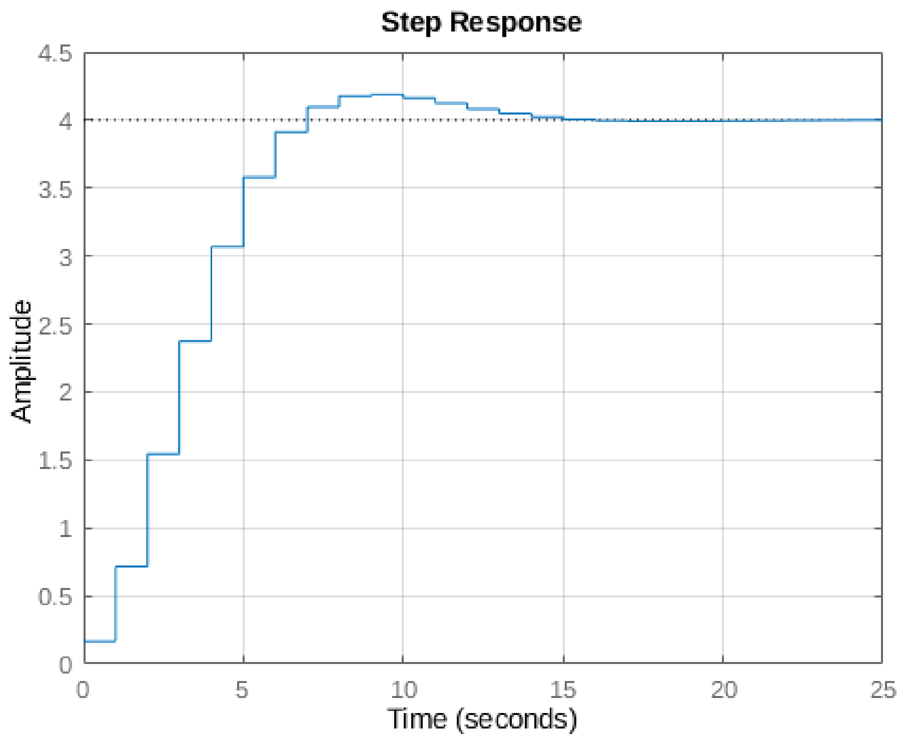
```
[b,a]=butter(2,.15);
G=tf(b*4,a,1)
```

```
G =

  0.165 z^2 + 0.33 z + 0.165
  --------------------------
    z^2 - 1.349 z + 0.514

Sample time: 1 seconds
Discrete-time transfer function.
```

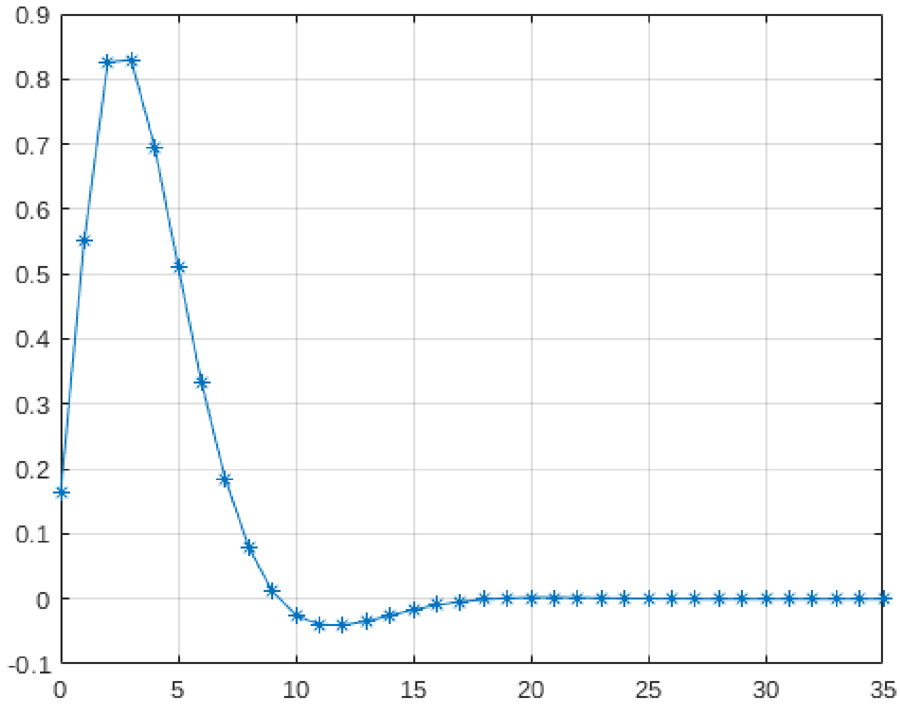
```
step(G), grid on
```



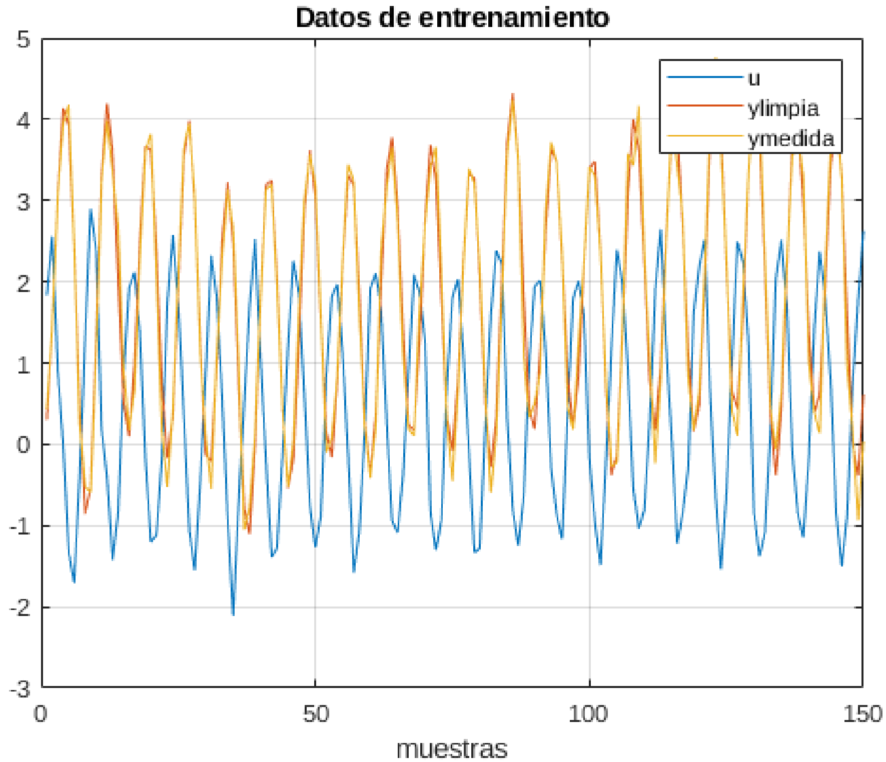

```
orden=36;ejeTh=0:(orden-1);
```

Calculemos la respuesta ante impulso $\{u_k\} = \{1, 0, 0, \dots\}$:

```
thperfecto=impulse(G,orden-1);
plot(ejeTh,thperfecto,'Marker','*'), grid on
```




```
N=150;
u=1.9*sin(0.85*(1:N))'+0.2*randn(N,1)+0.3*(1:N)'/N+0.3;
dtruido=0.25;
ylimpia=lsim(G,u);
y=ylimpia+dtruido*randn(N,1);
plot([u ylimpia y]), legend('u','ylimpia','ymedida'), grid on, title("Datos de entrenam
```



2.- Identificación de respuesta impulsional por mínimos cuadrados

Si la respuesta ante $\{1, 0, 0, \dots\}$ es $\{h_0, h_1, h_2, \dots\}$, denominada *respuesta impulsional*, entonces:

- por invarianza en el tiempo, ante $\{0, 1, 0, 0, \dots\}$ es $\{0, h_0, h_1, h_2, \dots\}$,
- por linealidad, ante $\{u_0, 0, 0, \dots\}$ es $\{u_0 h_0, u_0 h_1, u_0 h_2, \dots\}$,
- por linealidad (superposición), ante $\{u_0, u_1, 0, 0, \dots\}$ es $\{u_0 h_0, u_0 h_1, u_0 h_2, \dots\}$ + $\{0, u_1 h_0, u_1 h_1, u_1 h_2, \dots\}$, porque $\{u_0, u_1, 0, 0, \dots\} = \{u_0, 0, 0, 0, \dots\} + \{0, u_1, 0, 0, \dots\}$
- por linealidad (superposición), ante $\{u_0, u_1, u_2, u_3, \dots\}$ es $\{y_0 = u_0 h_0, y_1 = u_0 h_1 + u_1 h_0, y_2 = u_0 h_2 + u_1 h_1 + u_2 h_0, y_3 = u_0 h_3 + u_1 h_2 + u_2 h_1 + u_3 h_0, \dots\}$, que se denomina **fórmula de convolución discreta** para el cálculo de la respuesta temporal.

Escribiendo la última expresión (fórmula de convolución) en forma de matriz:

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \end{pmatrix} = \begin{pmatrix} u_0 & 0 & 0 & \dots & 0 \\ u_1 & u_0 & 0 & \dots & 0 \\ u_2 & u_1 & u_0 & \dots & 0 \\ \vdots & & \vdots & & \vdots \end{pmatrix} \begin{pmatrix} h_0 \\ h_1 \\ h_2 \\ \vdots \end{pmatrix} = T \cdot \mathbf{h}$$

ello permite:

1) simular el sistema, conocidos $\mathbf{h}$ y datos de entrada $\{u_0, u_1, u_2, u_3, \dots\}$

2) estimar $\mathbf{h}$ dados unos datos de entrada $\{u_0, u_1, u_2, u_3, \dots\}$ y salida $\{y_0, y_1, y_2, y_3, \dots\}$ por mínimos cuadrados.

Como en un sistema estable los elementos de la respuesta impulsional tienden a cero, se limita el número de elementos de $\mathbf{h}$ (y de columnas de T) a un tamaño "razonable" N.

Es equivalente a identificar una función de transferencia **FIR**

$$G(z) = h_0 + h_1 z^{-1} + h_2 z^{-2} + \dots + h_N z^{-N}$$

```
T=toeplitz(u,[u(1) zeros(1,orden-1)])
```

```
T = 150×36
    1.8370         0         0         0         0         0         0         0 ···
    2.5549    1.8370         0         0         0         0         0         0
    0.9138    2.5549    1.8370         0         0         0         0         0
   -0.0051    0.9138    2.5549    1.8370         0         0         0         0
   -1.3267   -0.0051    0.9138    2.5549    1.8370         0         0         0
   -1.7086   -1.3267   -0.0051    0.9138    2.5549    1.8370         0         0
   -0.3941   -1.7086   -1.3267   -0.0051    0.9138    2.5549    1.8370         0
    1.3233   -0.3941   -1.7086   -1.3267   -0.0051    0.9138    2.5549    1.8370
    2.8943    1.3233   -0.3941   -1.7086   -1.3267   -0.0051    0.9138    2.5549
    2.3910    2.8943    1.3233   -0.3941   -1.7086   -1.3267   -0.0051    0.9138
      ⋮
```

```
th=pinv(T)*y
```

```
th = 36×1
    0.1231
    0.5775
    0.8029
    0.8198
    0.6778
    0.4653
    0.4231
    0.1873
    0.0848
   -0.0991
      ⋮
```

Este es, obviamente, el modelo que mejor ajusta los datos de entrenamiento (en términos de error cuadrático)

```
std_err=sqrt(sum((y-T*th).^2/(N-orden))) %desv. típica error
```

```
std_err = 0.2436
```

```
varth=std_err^2*inv(T'*T);
trace(varth) %variación total parámetros estimados
```

```
ans = 0.2509
```

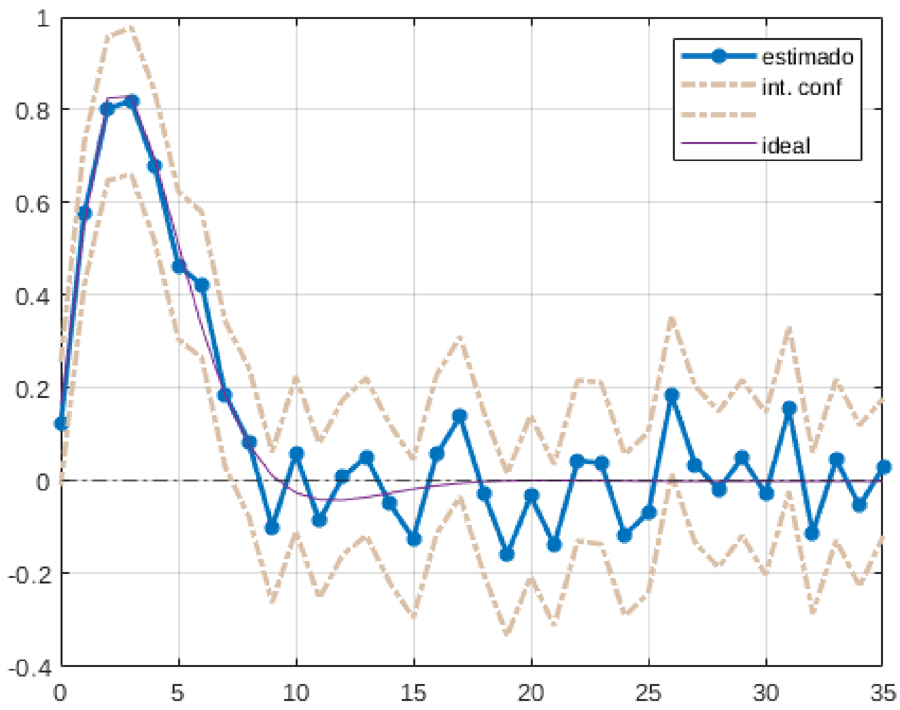
```
max(sqrt(eig(varth))) %desv. típ. componentes principales error parámetros
```

```
ans = 0.1580
```

```
std_th=sqrt(diag(varth));
max(std_th)
```

```
ans = 0.0890
```

```
plotIncierto(ejeTh,th,std_th)
hold on
plot(ejeTh,thperfecto)
hold off, yline(0,'-.')
legend('estimado','int. conf','','ideal')
```



Pruebo con ARX:

```
zz=zeros(orden,1);
modelo=arx(iddata([zz; y],[zz; u],1),[0 orden 0],arxOptions('InitialCondition','zero'))
```

```
modelo =
Discrete-time FIR model: y(t) =  B(z)u(t) + e(t)

  B(z) = 0.1231 + 0.5775 z^-1 + 0.8029 z^-2 + 0.8198 z^-3 + 0.6778 z^-4 + 0.4653 z^-5 + 0.4231 z^-6 + 0.18
          + 0.08482 z^-8 - 0.09908 z^-9 + 0.05882 z^-10 - 0.08501 z^-11 + 0.008625 z^-12 + 0.05275 z^-13
          - 0.04809 z^-14 - 0.1233 z^-15 + 0.05821 z^-16 + 0.1392 z^-17 - 0.02474 z^-18 - 0.1583 z^-19
          - 0.0316 z^-20 - 0.1373 z^-21 + 0.04438 z^-22 + 0.0388 z^-23 - 0.1159 z^-24 - 0.0684 z^-25 + 0.1
          + 0.03615 z^-27 - 0.01726 z^-28 + 0.05044 z^-29 - 0.02661 z^-30 + 0.1557 z^-31 - 0.1118 z^-32
                                                              + 0.04634 z^-33 - 0.05259 z^-34 + 0.02987 z

Sample time: 1 seconds
```

```
Parameterization:
   Polynomial orders:   nb=36   nk=0
   Number of free coefficients: 36
   Use "polydata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using ARX on time domain data.
Fit to estimation data: 87.61%
FPE: 0.0813, MSE: 0.03638
```

Conclusiones

Aunque la identificación directa de la respuesta impulsional (la respuesta ante la entrada $\{1, 0, 0, \ldots\}$, una secuencia $\{h_0, h_1, h_2, \ldots\}$) basada en datos de entrada-salida es usualmente catalogada como un método "no paramétrico" en muchos libros, en este caso utiliza las mismas fórmulas de mínimos cuadrados clásicos: los elementos de la respuesta impulsional pueden ser considerados como los parámetros a estimar.

La respuesta tiene teóricamente "duración infinita" pero debe limitarse. Como cuantos más elementos de la misma se estiman mayor es la varianza del estimado debida al ruido, el número de parámetros a estimar no debe ser "excesivo".

```
function plotIncierto(X,Y,DesvTip)
plot(X,Y,'LineWidth',2,'Marker','*')
hold on
plot(X,Y+2*DesvTip,'-.','Color',[.85 .75 .65],'LineWidth',2)
plot(X,Y-2*DesvTip,'-.','Color',[.85 .75 .65],'LineWidth',2)
hold off, grid on
end
```