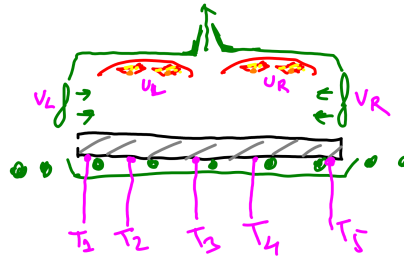


Mínimos cuadrados, pseudoinversa, ponderación, programación lineal, etc.: ejemplo "físico"

© 2026, Antonio Sala, Universitat Politècnica de València. Todos los derechos reservados.

Objetivo: comprobar las diferentes opciones de control sobre un sistema con 2 entradas y 5 salidas (posibles variables controladas).



Presentaciones en vídeo:

<https://personales.upv.es/asala/DocenciaOnline/Video/lsforn.html> [mínimos cuadrados]

<https://personales.upv.es/asala/DocenciaOnline/Video/linprforn.html> [programación lineal]

Tabla de Contenidos

Modelo.....	1
Problema 1: controlar "T" por mínimos cuadrados [no ponderados].....	2
Problema 2: Subconjunto de variables controladas.....	3
Problema 3: Mínimos cuadrados ponderados.....	4
Opción 4: control de la temperatura "media".....	5
Soluciones con programación lineal.....	7
2.1 Cota de error sobre salida.....	7
Suma del "valor absoluto" de los errores.....	8
Versión "linprog ponderada".....	8
Mantener T _{media} minimizando consumo en Kw.....	9
Conclusiones: entonces, ¿mínimos cuadrados o programación lineal?.....	10

Modelo

```
Gfull=[10 11 9 5 2;-5 -4 -3 -1 0;
        3 6 9 12 10;0 -1 -2 -3 -5]'
```

```
Gfull = 5x4
    10    -5     3     0
    11    -4     6    -1
     9    -3     9    -2
     5    -1    12    -3
     2     0    10    -5
```

*Ejemplo 1, sólo consideramos 2 actuadores:

```
G=Gfull(:, [1 3])
```

```
G = 5x2
    10     3
    11     6
     9     9
```

```
5 12
2 10
```

Problema 1: controlar "T" por mínimos cuadrados [no ponderados]

IMPORTANTE: variables controladas deben tener las mismas unidades, si no, hay que "escalar".

```
K=pinv(G)
```

```
K = 2x5
    0.0519    0.0448    0.0180   -0.0219   -0.0324
   -0.0280   -0.0149    0.0118    0.0476    0.0496
```

```
incTdeseado=[50 50 50 50 50]';
u=K*incTdeseado
```

```
u = 2x1
    3.0219
    3.3064
```

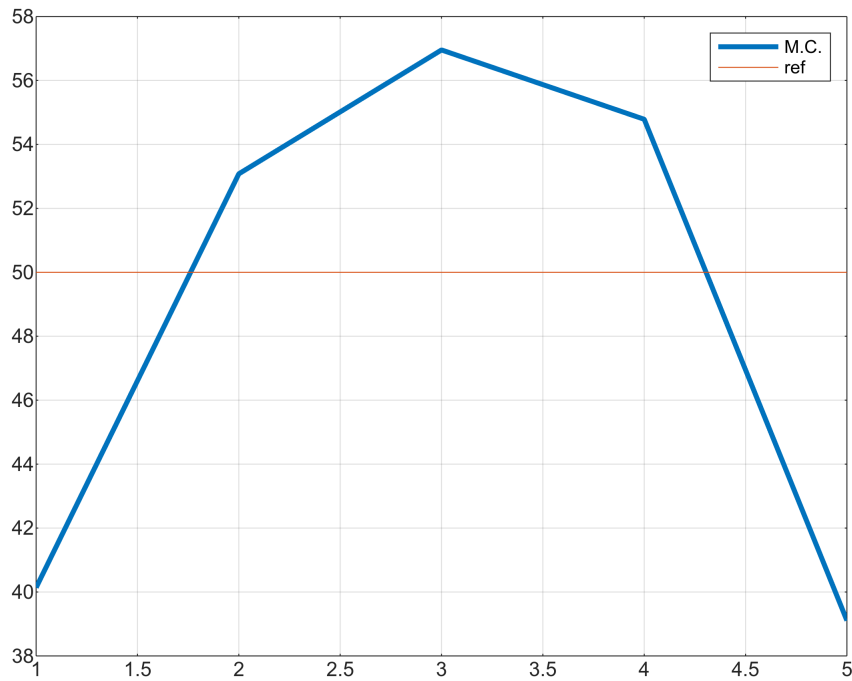
```
salidaLS=G*u
```

```
salidaLS = 5x1
    40.1384
    53.0795
    56.9549
    54.7863
    39.1078
```

```
error=incTdeseado-salidaLS
```

```
error = 5x1
    9.8616
   -3.0795
   -6.9549
   -4.7863
   10.8922
```

```
plot(salidaLS,LineWidth=2.5), hold on
plot(incTdeseado), hold off, grid on
legend("M.C.", "ref")
```



Problema 2: Subconjunto de variables controladas

```
TrozoG=G([2 4],:)
```

```
TrozoG = 2x2
    11     6
     5    12
```

```
%RGA=TrozoG.*inv(TrozoG')
u2=inv(TrozoG)*[50;50]
```

```
u2 = 2x1
    2.9412
    2.9412
```

```
y2=G*u2
```

```
y2 = 5x1
    38.2353
    50.0000
    52.9412
    50.0000
    35.2941
```

```
[salidaLS y2]
```

```
ans = 5x2
    40.1384    38.2353
    53.0795    50.0000
    56.9549    52.9412
    54.7863    50.0000
    39.1078    35.2941
```

```
errs=incTdeseado-ans
```

```
errs = 5x2
```

```

9.8616    11.7647
-3.0795     0
-6.9549   -2.9412
-4.7863    0.0000
10.8922   14.7059

```

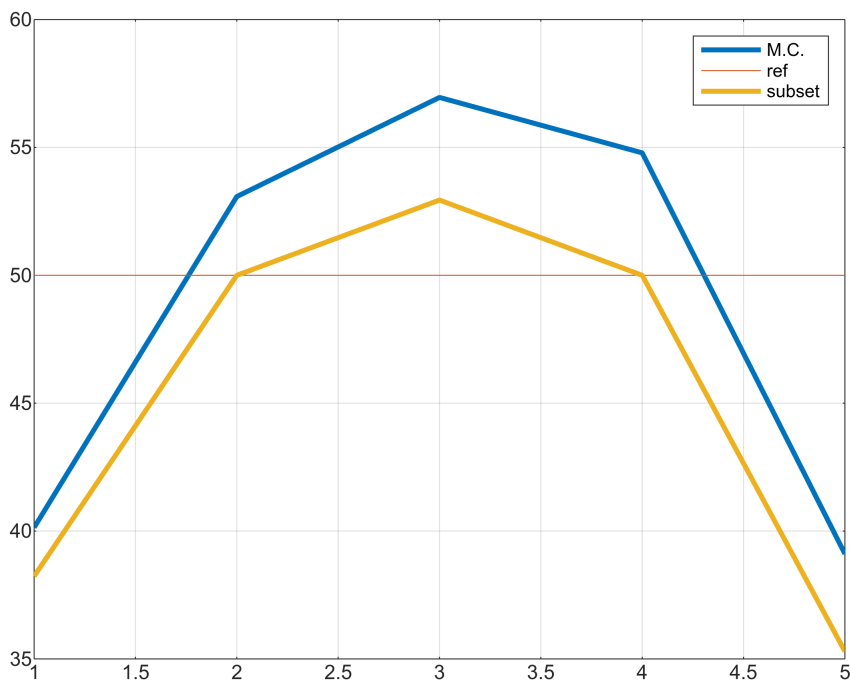
```
[norm(errs(:,1)) norm(errs(:,2))]
```

```
ans = 1x2
    17.2236    19.0610
```

```

plot(salidaLS,LineWidth=2.5), hold on
plot(incTdeseado),
plot(y2,LineWidth=2.5)
hold off, grid on, legend("M.C.", "ref", "subset")

```



Problema 3: Mínimos cuadrados ponderados

```

Pesos=[1 5 15 5 1];
%Pesos=[1 1e6 1 1e6 1];
Gesc=diag(Pesos)*G;
incTdeseadoESC=diag(Pesos)*incTdeseado;
u=pinv(Gesc)*incTdeseadoESC

```

```
u = 2x1
    2.7325
    2.8842
```

```

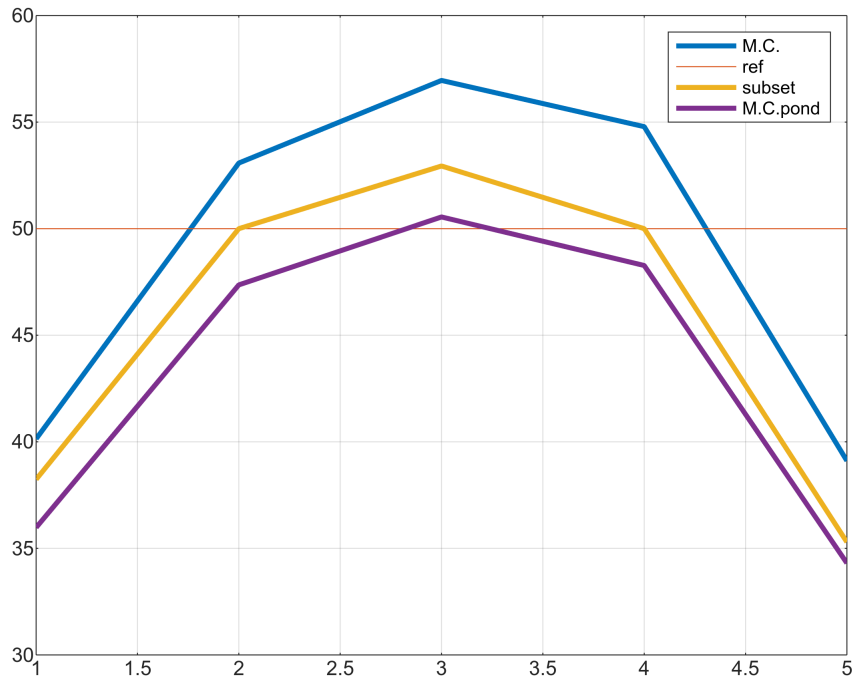
yLSpond=G*u;
[yLSpond salidaLS]

```

```
ans = 5x2
    35.9771    40.1384
    47.3621    53.0795
    50.5496    56.9549
```

```
48.2723    54.7863
34.3066    39.1078
```

```
plot(salidaLS,LineWidth=2.5), hold on
plot(incTdeseado),
plot(y2,LineWidth=2.5)
plot(yLSpond,LineWidth=2.5)
hold off, grid on, legend("M.C.", "ref", "subset", "M.C.pond")
```



Opción 4: control de la temperatura "media"

```
Gmed=[1 1 1 1 1]*G/5
```

```
Gmed = 1x2
    7.4000    8.0000
```

Me sobran actuadores... Podría utilizar sólo un actuador:

```
u1=inv(Gmed(1))*50
```

```
u1 =
    6.7568
```

El perfil de temperaturas sería:

```
yprueba=G*[u1;0]
```

```
yprueba = 5x1
    67.5676
    74.3243
    60.8108
    33.7838
    13.5135
```

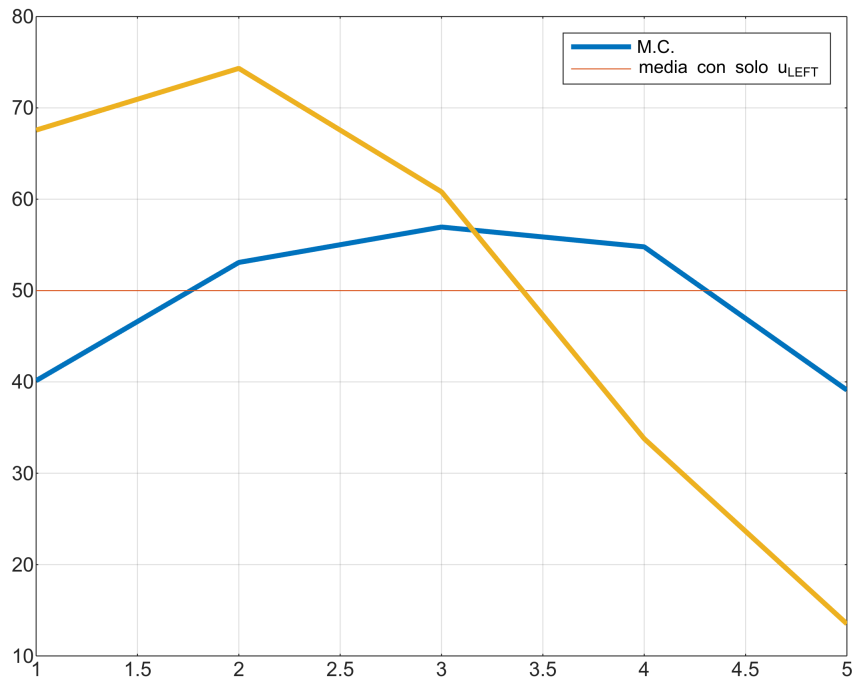
```
mean(yprueba)
```

```
ans =  
50.0000
```

```
mean(salidaLS)
```

```
ans =  
48.8134
```

```
plot(salidaLS,LineWidth=2.5), hold on  
plot(incTdeseado),  
plot(yprueba,LineWidth=2.5), hold off, grid on  
legend("M.C.", "media con solo u_{LEFT}")
```

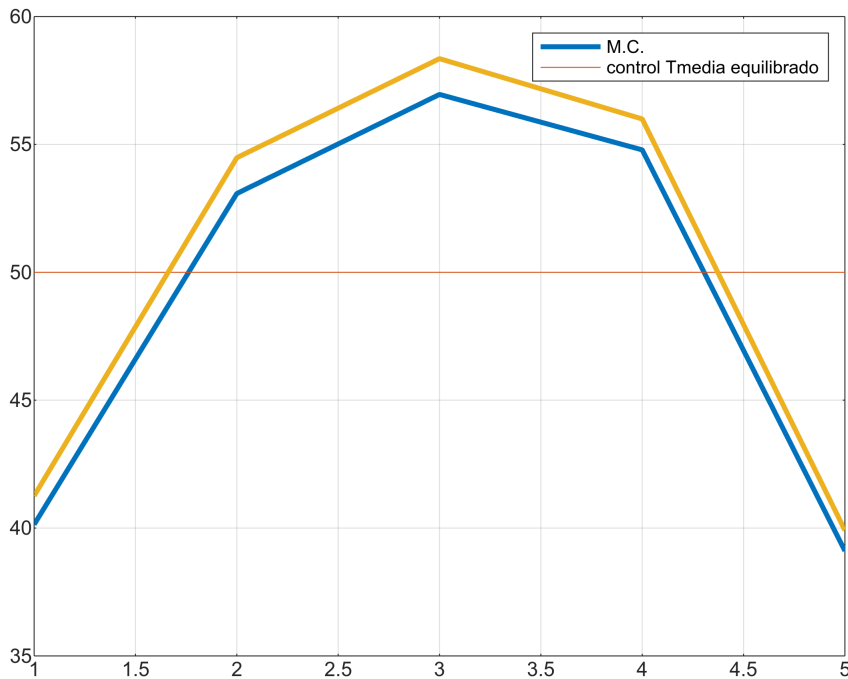


Si quiero "repartir" el esfuerzo de control, puedo minimizar $u_L^2 + u_R^2$ que consigan la temperatura "media" deseada. **IMPORTANTE:** actuadores deben tener las mismas unidades, si no, hay que "escalar".

```
u_LSbalancing=pinv(Gmed)*50
```

```
u_LSbalancing = 2x1  
3.1155  
3.3681
```

```
plot(salidaLS,LineWidth=2.5), hold on  
plot(incTdeseado),  
plot(G*u_LSbalancing,LineWidth=2.5), hold off, grid on  
legend("M.C.", "control Tmedia equilibrado")
```



*Todas las respuestas son "relativamente parecidas", pero cada selección de variables controladas/índice de coste hará que el sistema de control reaccione de forma diferente ante "perturbaciones" sobre todo si son marcadamente asimétricas.

Soluciones con programación lineal

2.1 Cota de error sobre salida

Podríamos querer minimizar el máximo error, no el "error RMS":

minimizar γ sujeto a $e = r - G \cdot u$, $-\gamma \leq e \leq +\gamma$.

Eso lo haremos con PROGRAMACIÓN LINEAL.

- Las variables de decisión serán $\xi = (\gamma, u_L, u_R)^T$. El coste será $[1 \ 0 \ 0] \cdot \xi$.
- Las restricciones serán, siendo $\mathbf{1}$ una columna de cinco "unos".

$$(1) \quad r - Gu \leq \gamma, \text{ o sea } -\gamma * \mathbf{1} - Gu \leq -r$$

$$(2) \quad -\gamma \leq r - Gu, \text{ o sea } -\gamma * \mathbf{1} + Gu \leq +r$$

```
Aineq1=[-ones(5,1) -G];bineq1=-incTdeseado;
Aineq2=[-ones(5,1) G];bineq2=incTdeseado;
Aineq=[Aineq1;Aineq2];bineq=[bineq1;bineq2];
sol_linprog=linprog([1 0 0],Aineq,bineq)
```

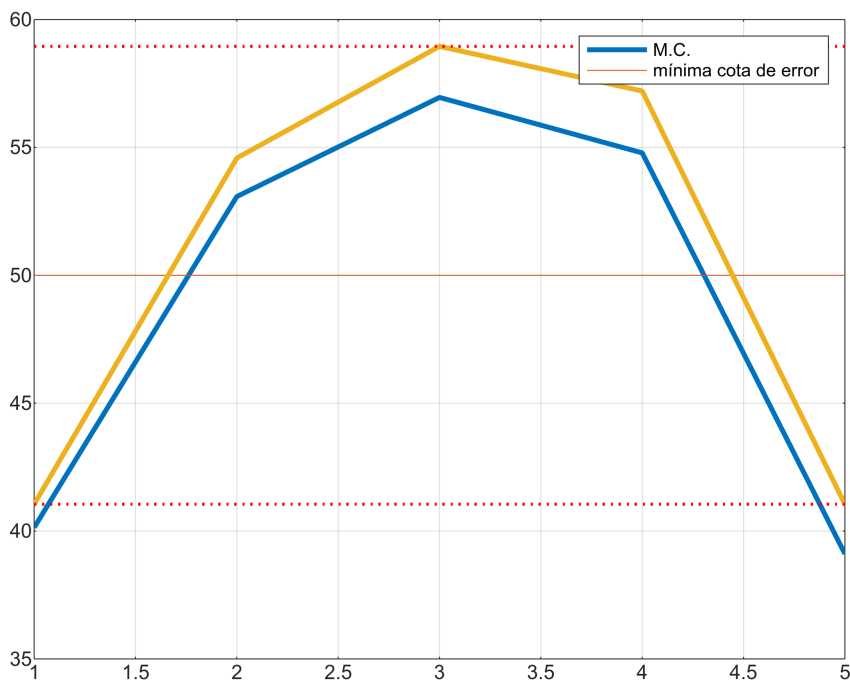
```
Optimal solution found.
sol_linprog = 3x1
    8.9520
    3.0568
```

3.4934

```
gamma_linprog=sol_linprog(1);  
u_linprog=sol_linprog(2:3)
```

```
u_linprog = 2×1  
 3.0568  
 3.4934
```

```
y_linprog=G*u_linprog;  
plot(salidaLS,LineWidth=2.5), hold on  
plot(incTdeseado),  
plot(y_linprog,LineWidth=2.5), hold off, grid on  
yline(incTdeseado+gamma_linprog,':r',LineWidth=1.5)  
yline(incTdeseado-gamma_linprog,':r',LineWidth=1.5)  
legend("M.C.", "mínima cota de error")
```



Suma del "valor absoluto" de los errores

$$e = r - G \cdot u, \quad e = e_{neg} + e_{pos}, \quad e_{neg} \leq 0, \quad e_{pos} \geq 0.$$

minimizar $e_{pos} - e_{neg}$.

Versión "linprog ponderada"

Podría poner una cota máxima de error para cada temperatura... o "apretar más" en temperaturas centrales... incluso ponderar diferentemente los errores por encima o por debajo de la referencia.

Por ejemplo, cambiar $-\gamma \leq e \leq +\gamma$ por $-W_+ \cdot \gamma \leq e \leq +W_- \cdot \gamma$.

O, minimizar $W_+ \cdot e_{pos} - W_- \cdot e_{neg}$.

*Se dejan los detalles al lector, por brevedad.

Mantener Tmedia minimizando consumo en Kw

Podríamos querer minimizar el coste energético "en kW" que es lo que viene en la factura y no en "kW²".

minimizar $u_L + u_R$ sujeto a $0 = r_{med} - G_{med} \cdot u$

También lo haremos con PROGRAMACIÓN LINEAL.

- Las variables de decisión serán $\xi = (u_L, u_R)^T$. El coste será $[1 \ 1] \cdot \xi$.
- Las restricciones serán $G_{med}u = r_{med}$ (igualdad) y que los actuadores estén en cierto rango $0 \leq u_L \leq 5$, $0 \leq u_R \leq 5$

```
u_linprog2=linprog([1 1],[[],[]],[Gmed,50],[0 0],[5 5])
```

```
Optimal solution found.
```

```
u_linprog2 = 2x1
```

```
    1.3514
```

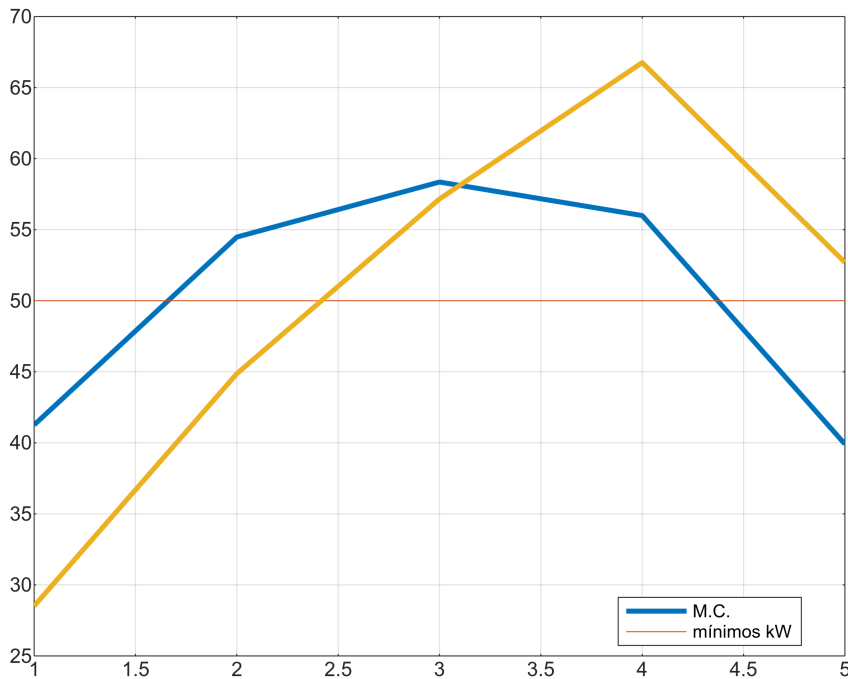
```
    5.0000
```

Como era intuitivamente esperable, satura el actuador de más ganancia (°C/kW).

```
y_linprog2=G*u_linprog2;  
mean(y_linprog)
```

```
ans =  
50.5677
```

```
plot(G*u_LSbalancing,LineWidth=2.5), hold on  
plot(incTdeseado),  
plot(y_linprog2,LineWidth=2.5), hold off, grid on  
legend("M.C.", "mínimos kW",Location="best")
```



Conclusiones: entonces, ¿mínimos cuadrados o programación lineal?

¿Por qué muchos libros se centran en "mínimos cuadrados"? Porque para modelos lineales tiene una solución muy atractiva para control:

- Tiene una forma explícita (matriz pseudoinversa) que no requiere optimización numérica "en línea".
- Esa solución "explícita" es LINEAL $u=K*ref...$
- En caso "dinámico" acabará siendo el control LQR y filtro de Kalman, y será un controlador LINEAL $K(s)$.

Pero si en mi aplicación concreta mis requisitos económicos o limitaciones técnicas no "cuadran" del todo con las suposiciones de los mínimos cuadrados, bueno, pues tendré que utilizar programación lineal o cuadrática (o no lineal) para obtener el punto de operación (si es constante), o "control predictivo con índice de coste económico" en el caso dinámico.

En muchas aplicaciones se usa:

- optimización linprog/quadprog u otras en el cálculo de un **punto de operación** (en bucle abierto).
- **Control lineal en bucle cerrado** para mantener dicho punto de operación (PID o, en caso de que el PID no sea suficiente, optimización de índices **cuadráticos** en control LQR/LQG o predictivo), por sus ventajas teóricas para probar estabilidad y prestaciones.