

Mínimos cuadrados ponderados (introducción a su uso en control)

Caso 2: exceso de variables manipuladas

Presentación en vídeo: <http://personales.upv.es/asala/YTV/mcpu.html>

© 2021, Antonio Sala Piqueras. Universitat Politècnica de València. Todos los derechos reservados.

*Este código ejecutó correctamente en Matlab R2020b

Objetivo: Cuando se quiere controlar "pocas cosas" con muchas VM (más entradas que salidas) en principio hay infinitas soluciones, eso es bueno (en muchos casos se querrá hacer una "selección" y usar menos actuadores automáticos --caros y más propensos a fallar que los "manuales"--). El enfoque básico es mínimos cuadrados (minimizar $\sum_{i=1}^N u_i^2$ sujeto a la restricción de igualdad $r = Gu$) pero cuando cada variable tiene unas unidades físicas y unos objetivos de precisión debe cambiarse a mínimos cuadrados **ponderados**. El objetivo de este material es ilustrar el significado de dichas ponderaciones.

Tabla de Contenidos

Mínimos cuadrados ponderados (introducción a su uso en control).....	1
Caso 2: exceso de variables manipuladas.....	1
2.-Exceso de entradas (variables manipuladas).....	1
2.1- Mínimos cuadrados no ponderados.....	1
2.2- Pesos de entrada.....	2
1.3- Escalado basado en umbrales de saturación (equivalente).....	3
Conclusiones.....	4

2.-Exceso de entradas (variables manipuladas)

Consideremos un sistema (estático, por simplicidad) $y = Gu$ con:

```
G=[1 10 .4 1;2 9 0 4], Nact=size(G,2);
```

```
G = 2x4
    1.0000    10.0000    0.4000    1.0000
    2.0000     9.0000         0     4.0000
```

donde para controlar dos salidas tenemos cuatro actuadores.

2.1- Mínimos cuadrados no ponderados

Si deseamos (objetivo de control) valores de referencia:

```
ref=[4 2]';
```

la solución mínimo cuadrática (mínima norma de u entre las infinitas soluciones) es:

```
uopt=pinv(G)*ref
```

```
uopt = 4×1
-0.1337
 0.4563
 0.0769
-0.4598
```

y la salida alcanzada es:

```
y=G*uopt
```

```
y = 2×1
 4.0000
 2.0000
```

con un esfuerzo de control dado por:

```
norm(uopt) %la óptima
```

```
ans = 0.6658
```

aunque, como hemos dicho, el número de soluciones es infinito:

```
otrau=uopt+null(G)*randn(2,1)
```

```
otrau = 4×1
-0.1726
 0.5031
-0.7820
-0.5457
```

```
G*otrau
```

```
ans = 2×1
 4.0000
 2.0000
```

```
norm(otrau)
```

```
ans = 1.0919
```

2.2- Pesos de entrada

Deseamos no utilizar tanto el actuador 4, por lo que podemos poner unos pesos de modo que

minimicemos $\sum_{i=1}^4 (w_i u_i)^2$:

```
Peso_u=diag([1 1 1 100]);
Gesc=G/Peso_u
```

```
Gesc = 2×4
 1.0000 10.0000 0.4000 0.0100
 2.0000 9.0000 0 0.0400
```

```
uopt2=Peso_u\pinv(Gesc)*ref
```

```
uopt2 = 4×1
-1.3019
```

```
0.5117
0.4633
-0.0004
```

```
G*uopt2
```

```
ans = 2×1
4.0000
2.0000
```

Coincide con fórmulas en libros de mínimos cuadrados ponderados (usando el cuadrado del peso):

```
W=inv(Peso_u^2);
uopt2_otravez=W*G'*inv(G*W*G')*ref
```

```
uopt2_otravez = 4×1
-1.3019
0.5117
0.4633
-0.0004
```

1.3- Escalado basado en umbrales de saturación (equivalente)

```
Incremento_U_aceptable=[4 4 4 0.5]%reducir si >1 criterio1 abajo
```

```
Incremento_U_aceptable = 1×4
4.0000 4.0000 4.0000 0.5000
```

```
Matriz_esc_u=diag(Incremento_U_aceptable);
% cambio de variable: u=Matriz_esc_u*uesc
```

Proponemos:

```
Gesc=G*Matriz_esc_u; % Gesc*uesc=G*Matriz_esc_u*uesc
```

La solución del problema de mínimos cuadrados con ese peso:

```
uesc_opt2=pinv(Gesc)*ref %en unidades "escaladas" +/- 1
```

```
uesc_opt2 = 4×1
-0.2921
0.1263
0.1048
-0.1058
```

```
uopt2=Matriz_esc_u*uesc_opt2 %en unidades físicas
```

```
uopt2 = 4×1
-1.1683
0.5054
0.4191
-0.0529
```

```
y2=G*uopt2 %en unidades físicas
```

```
y2 = 2×1
4.0000
```

2.0000

Si cada elemento de u_{esc_opt2} (abs) es menor de 1, está por debajo del umbral, y la solución sería válida.

```
max(abs(u_esc_opt2)) %si realmente nos interesa esta geometría, deberíamos plantearlo co
```

```
ans = 0.2921
```

Si $(\sum_{i=1}^4 u_{esc,i}^2)^{1/2} < 1$ entonces también todos los elementos están por debajo de su umbral:

```
crit1=norm(u_esc_opt2)
```

```
crit1 = 0.3513
```

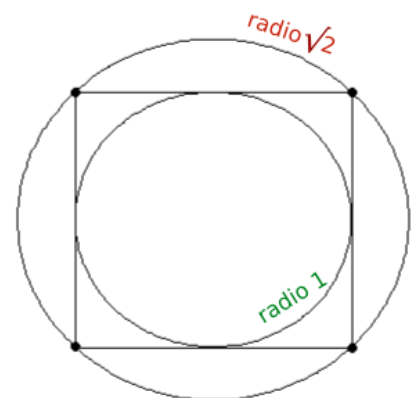
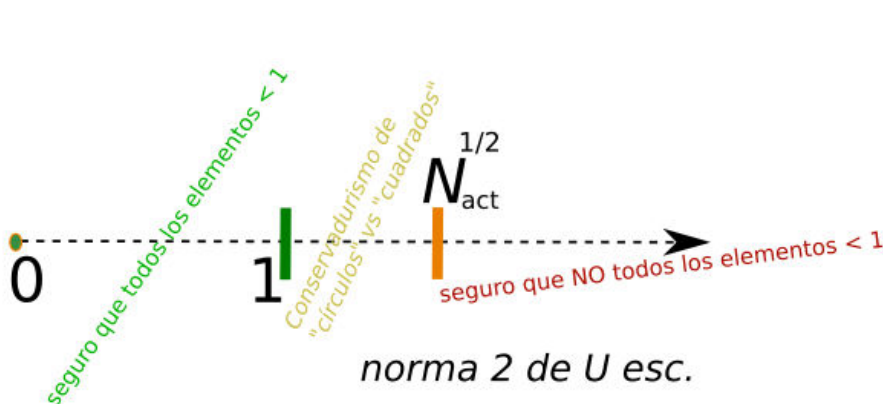
aunque sumar $\sqrt{1+1+\dots+1} = \sqrt{N_{act}}$ puede tener valor mayor a 1 sin superar individualmente el límite.

```
crit2=norm(u_esc_opt2)/sqrt(Nact)
```

```
crit2 = 0.1757
```

- Si $crit1 < 1$ entonces *todos* los actuadores por *debajo* del umbral ($crit1 < 1$ es cond. **suficiente** para asegurar que todo **SÍ** está **por debajo** del umbral)

- Si $crit2 > 1$ (o sea, $crit1 > \sqrt{N}$) entonces *seguro* que alguno de los actuadores *sobrepasa* el umbral ($crit1 > \sqrt{N}$ es cond. **suficiente** para certificar que **NO** se cumplen los umbrales; $crit1 \leq \sqrt{N}$ es condición **necesaria** para que todos los errores estén **por debajo** del umbral aceptable.)



Conclusiones

Cuando se quiere controlar "pocas cosas" con muchas VM (más entradas que salidas) en principio hay infinitas soluciones, eso es bueno (en muchos casos se querrá hacer una "selección" y usar menos actuadores automáticos). El enfoque básico es mínimos cuadrados (minimizar $\sum_{i=1}^N u_i^2$ sujeto a la restricción de igualdad $r = Gu$) pero cuando cada variable tiene unas unidades físicas y unos objetivos de precisión debe cambiarse a mínimos cuadrados **ponderados**.

En un primer enfoque, se trata de minimizar $\sum_{i=1}^N (w_i u_i)^2$, dejando a criterio del usuario el valor del peso w_i . Cuanto mayor sea cada peso w_i , más cercano a cero se desea ese actuador (a expensas de otros).

En un segundo enfoque, lo que se deja a criterio del usuario es especificar un cierto umbral máximo del actuador, por ejemplo como "incremento hasta saturación".

Las ventajas del segundo enfoque umbrales son:

- que trabajamos en unidades "físicas" en dichos umbrales, y por lo tanto, resulta trivial que unas variables manipuladas sean en voltios, otras milivoltios, y otras litros... con su especificación de saturación (en sus unidades).
- que de forma rápida (aproximada por un posible error en un factor $\sqrt{N}$) podemos asegurar que se cumplen las especificaciones (actuador < umbral) si la norma del vector de "entradas escalado" es menor a 1.