

Criterios de robustez para sistemas lineales

José Franco Carlos Mario Vélez

Noviembre de 2010

1. Introducción

Al construir un modelo de un sistema, éste siempre incluye algún grado de incertidumbre con respecto al sistema real y a su entorno. El sistema puede cambiar de manera inesperada o estar sujeto a perturbaciones inesperadas. Debido a factores como los cambios en los parámetros, las dinámicas y retardos no modeladas, los cambios en los puntos de operación, el ruido en los sensores o las perturbaciones no predichas, el modelo de un proceso no puede ser una representación completamente acertada del sistema real [1].

La incertidumbre se refiere a la diferencia entre el comportamiento de un modelo y la realidad. En general, para representar esta incertidumbre y poder lidiar con ella, es necesario utilizar, más que un único modelo, un conjunto de modelos que permita incluir dinámicas de la planta que no están explícitamente representadas en la estructura del modelo [5].

Al diseñar un sistema de control para un sistema donde la presencia de incertidumbre es significativa, es necesario tener en cuenta esta representación para minimizar el efecto de los errores que presenta el modelo frente a la realidad en lo que respecta al regulador. Se dice que un sistema de control es robusto si es insensible a las diferencias entre el sistema real y al modelo utilizado para diseñarlo [4].

En este trabajo se expone la forma de representar la incertidumbre y analizar la robustez de sistemas SISO, luego se extienden estos conceptos a sistemas MIMO y se presentan ejemplos en MATLAB usando el toolbox de control robusto.

2. Incertidumbre y robustez para sistemas SISO

Los sistemas SISO son perfectos para introducir las representaciones de incertidumbre y los criterios de robustez ya que facilitan la visualización de estos usando condiciones y métodos gráficos en el dominio de la frecuencia que son bien conocidos, y permiten generalizar los conceptos a sistemas MIMO más adelante.

2.1. Representación de la incertidumbre

Las diferentes fuentes de incertidumbre que afectan a un modelo pueden resumirse en dos clases generales de incertidumbre: La incertidumbre paramétrica, donde cada parámetro β_i está contenido en un intervalo $[\beta_{\min}, \beta_{\max}]$; o incertidumbre causada por dinámicas no modeladas. [4].

Cualquiera de estos tipos de incertidumbres, o una combinación de ambos, se puede representar mediante perturbaciones multiplicativas [4] aplicadas a una planta nominal, de la forma

$$G_p(s) = G(s)(1 + w_I(s)\Delta_I(s)), \quad \|\Delta_I\|_\infty \leq 1$$

Donde w_I simplemente es una función ponderadora, G es la planta nominal y la norma \mathcal{H}_∞ se calcula como [2]

$$\|G\|_\infty = \sup_{\omega} \sigma_{\max}[G(i\omega)]$$

En este caso Δ_I es cualquier función de transferencia estable que cumpla con tener norma \mathcal{H}_∞ menor o igual que 1.

El tipo de incertidumbre esperada determina la elección de w_I , que varía desde ser una constante real para incertidumbre en los parámetros, a ser una función de transferencia sencilla para representar polos no modelados. Skogestad muestra diferentes w_I para diferentes tipos de incertidumbre [4].

2.2. Estabilidad robusta

Se define que un sistema es robustamente estable cuando cada modelo en el conjunto de plantas posibles es estable [4]. A partir del criterio de estabilidad de Nyquist, esto equivale a que el diagrama de Nyquist no rodee el punto $(-1, 0)$ para ninguna planta en este conjunto.

En el diagrama de Nyquist, se grafican las partes real e imaginaria de la función de ganancia de lazo $L(i\omega)$ variando la frecuencia ω como se observa en la Fig. 1. Es claro que si el sistema no es nominalmente estable, entonces no será robustamente estable, por lo que esta condición equivale a evaluar si para cada frecuencia ω , la distancia entre $L(i\omega)$ y el punto $(-1, 0)$ es mayor que el radio generado por $L(i\omega)w_I(i\omega)$:

$$\begin{aligned} \text{RS} &\Leftrightarrow |w_I L| < |1 + L| \quad \forall \omega \in \mathbb{R}^+ \\ &\Leftrightarrow \left| \frac{w_I L}{1 + L} \right| < 1 \quad \forall \omega \in \mathbb{R}^+ \\ &\Leftrightarrow \|w_I T\|_\infty < 1 \end{aligned}$$

Donde $T = \frac{L}{1+L}$ es la función de transferencia en lazo cerrado del modelo nominal.

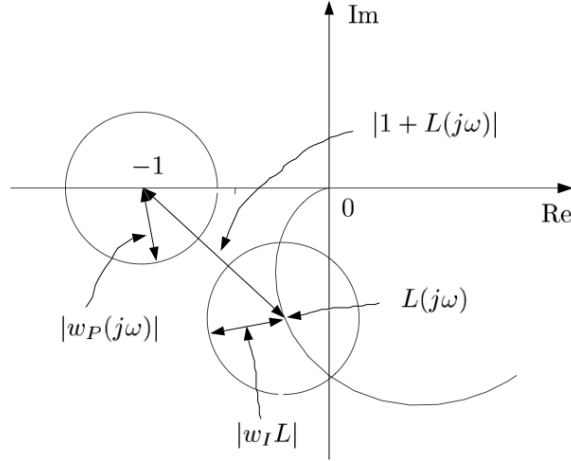


Fig 1: Ilustración de la estabilidad robusta SISO

3. Incertidumbre y robustez para sistemas MIMO

Aunque los sistemas SISO presentan una manera sencilla de visualizar los criterios de robustez para plantas básicas, las plantas de múltiples entradas y múltiples salidas presentan dificultades adicionales que deben ser consideradas para analizar la robustez de una categoría más general de sistemas.

3.1. Representación de la incertidumbre

De manera similar al caso de los sistemas SISO, para los sistemas MIMO se representa la incertidumbre utilizando perturbaciones multiplicativas. La estructura general con la que se representan los sistemas con incertidumbre se presenta en la Fig. 2. En este caso, Δ es una matriz diagonal por bloques donde cada bloque representa una fuente de incertidumbre.

Para analizar la estabilidad robusta del sistema, se utiliza la forma $M\Delta$ que se observa en la Fig. 3. Esta forma se obtiene a partir de cerrar el lazo¹ para considerar únicamente los efectos de $u\Delta$ y $y\Delta$.

En MATLAB, la incertidumbre paramétrica se representa usando el comando `ureal('nombre', valorNominal, propiedad, argumento)`, donde `propiedad` puede tomar los siguientes valores: `PlusMinus` para una desviación aditiva, `Range` para especificar un rango donde está el parámetro, o `Percentage` para establecer una desviación en porcentaje. Posteriormente, estos objetos `ureal`

¹Primero se aplica una transformación lineal inferior para obtener una matriz $N = F_l(P, K) = P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}$, y luego se toma únicamente la parte de N que involucra a $u\Delta$ y $y\Delta$ (N_{11}). En MATLAB, se puede usar el comando `[M,Delta] = lftdata(A)`, donde **A** es el modelo representado mediante un objeto `uss`.

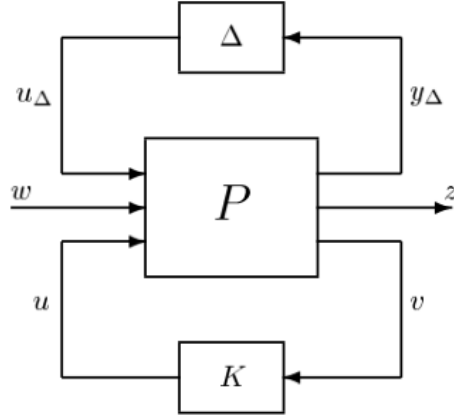


Fig 2: Configuración general para sistemas con incertidumbre

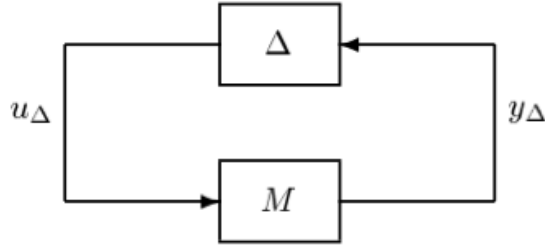


Fig 3: Estructura $M\Delta$

se pueden tratar como valores reales, creando por ejemplo matrices para invocar la función $\text{ss}(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ y hacer así un modelo en espacio de estados con incertidumbre (El objeto resultante en este caso sería un objeto uss).

3.2. Estabilidad robusta

La extensión del criterio de estabilidad de Nyquist a los sistemas MIMO está dada por la siguiente condición [4]: Dado un conjunto convexo de perturbaciones Δ acotado bajo la norma \mathcal{H}_∞ , el sistema modelado con la estructura $M\Delta$ de la Fig. 3 es robustamente estable sí y sólo sí el diagrama de Nyquist de $\det(I - M\Delta(s))$ no rodea el origen, para cualquier $\Delta \in \Delta$. De manera equivalente,

$$\forall \Delta \in \Delta, \forall \omega \in \mathbb{R}^+, \det(I - M\Delta(i\omega)) \neq 0$$

El problema con esta condición es que no indica qué tan robusto es el sistema: únicamente señala si es robustamente estable o no. Para perturbaciones diagonales por bloques, existen dos conceptos útiles a la hora de evaluar qué tan insensible a las perturbaciones es el sistema: El valor singular estructurado μ y el margen de estabilidad robusta k_m . Estos se definen como sigue:

$$k_m = \min\{k \in \mathbb{R}^+ : \det(I - kM\Delta) = 0\}$$

$$\mu = \frac{1}{k_m}$$

Mientras más grande es el margen de estabilidad robusta, puede soportar una cantidad más grande de la incertidumbre modelada. Con esta definición en mente, se puede expresar una forma adicional de la condición de estabilidad robusta:

$$\text{RS} \Leftrightarrow k_m > 1$$

Si el margen de estabilidad robusta es menor que 1, el sistema es estable para todas las perturbaciones que no alejen a los parámetros de sus valores nominales en un $(k_m)100\%$ o más pero no es estable para todas las perturbaciones modeladas[3].

En MATLAB, se puede usar la función `robuststab` para realizar un análisis de robustez de un objeto `uss`. Esta función recibe como argumento al sistema, y retorna cuatro valores: el margen de estabilidad, un conjunto de parámetros que desestabilizan al sistema, un reporte e información adicional². Para el margen de estabilidad se entregan las cotas superior e inferior, y la frecuencia en la que ocurre la desestabilización. En el reporte, se ve adicionalmente cuales son los parámetros que más influyen en el margen de estabilidad. Adicionalmente, en la estructura de información adicional se almacenan las cotas de μ para diferentes frecuencias.

4. Ejemplo usando MATLAB

Para ilustrar los conceptos vistos, se presenta un ejemplo utilizando MATLAB y el toolbox de control robusto. El sistema a analizar es un péndulo invertido. Para empezar, se definen los parámetros a utilizar en el modelo nominal:

$M = 0,5$, la masa del carro, $m = 0,2$, la masa del brazo, $b = 0,1$, la fricción en el carro, $I = 0,006$, el momento de inercia del péndulo, $g = 9,8$, la gravedad, y $l = 0,3$, la longitud del péndulo (hasta el centro de masa).

Se toma la ecuación de estado linealizada del péndulo para obtener las matrices A , B , C y D :

²`[stabmarg,destabunc,report,info] = robuststab(sys)`

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{(I+ml^2)b}{p} & \frac{m^2gl^2}{p} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{mlb}{p} & 0 & \frac{ml}{p} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ \frac{(I+ml^2)}{p} \\ 0 \\ \frac{ml}{p} \end{bmatrix}$$

$$C = I_4$$

$$D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Donde $p = I(M + m) + Mml^2$. Se invoca la función `ss` para crear un modelo de espacio de estados: `pend = ss(A, B, C, D)`; , tras lo que se utiliza la función `place` para crear un control por realimentación de estado para el modelo nominal:

```
polos = [-1.1, -1.6, -3.2, -2.9];
Kn = place(A, B, polos);
fbpend = feedback(pend, Kn);
```

Luego se crea el modelo incierto. Supóngase que se estima que los parámetros M, m, b, I, g , y l tienen una incertidumbre estimada de 20 %, 20 %, 20 %, 20 %, 0,1 %³, y 20 % respectivamente. Para modelar estos parámetros inciertos, se usa la función `ureal`:

```
uM = ureal('M', M, 'Percent', 20);
um = ureal('m', m, 'Percent', 20);
ub = ureal('b', b, 'Percent', 20);
ui = ureal('I', I, 'Percent', 20);
ug = ureal('g', g, 'Percent', 0.1);
ul = ureal('l', l, 'Percent', 20);
up = ui*(uM+um)+uM*um*ul^2;
```

Las matrices inciertas de la ecuación de estado `uA` y `uB` se crean exactamente como antes, pero se utilizan los parámetros inciertos. Se utilizan las mismas matrices $C = I_4$ y $D = 0_{4 \times 1}$ para crear un modelo en espacio de estados con incertidumbre, llamando una vez más la función `ss`, y se cierra el lazo con el control generado para el sistema nominal para evaluar la robustez del sistema

³La gravedad tiene un bajo nivel de incertidumbre.

Parámetro	Valor nominal	Valor desestabilizante	Diferencia relativa
M	0.5	0,14	-71 %
m	0.2	$9,6 \times 10^{-15}$	-100 %
b	0.1	0,17	69 %
I	0.006	$2,9 \times 10^{-16}$	-100 %
g	9.8	9,84	0,004 %
l	0.3	0,15	-49 %

Tabla 1: Valores nominales de los parámetros y valores desestabilizantes entregados por `robuststab`

en lazo cerrado:

```
upend = ss(uA, uB, C, D);
ufbpend = feedback(upend, Kn);
```

Teniendo ya el modelo en espacio de estados con incertidumbre, se procede a hacer el análisis de robustez usando la función `robuststab`:

```
[stabmarg, destabunc, report, info] = robuststab(ufbpend);
```

De `stabmarg` se obtienen los límites superior e inferior del margen de estabilidad robusta⁴: $1,5538 \leq k_m \leq 5$. Ya que cualquier elemento del intervalo es mayor que 1, el sistema es robustamente estable.

En `destabunc` se encuentra un conjunto de parámetros que causan inestabilidad en el sistema. Estos valores pueden verse en la Tabla 1, donde se observa que exceptuando al parámetro `g`, el valor desestabilizante de cada parámetro difiere en una cantidad mayor que la incertidumbre especificada. Para verificar que esta combinación de parámetros en efecto desestabiliza al sistema, se puede usar el comando `usubs` para remplazarlos en el sistema con incertidumbre, y así obtener un objeto `ss`, para luego evaluar la estabilidad del sistema con la función `po1e`. Sin embargo, las matrices que se obtienen en este caso al hacerlo presentan problemas numéricos (el valor p , que aparece en el denominador de algunos elementos de las matrices A y B , es casi 0 para estos valores de m y I), por lo que MATLAB arroja un error.

En `report` se describen los resultados, entregando qué tanta incertidumbre modelada puede tolerar el sistema, qué características tiene la combinación de parámetros desestabilizante y en qué frecuencia se encuentra la inestabilidad, y las sensibilidades frente a cada parámetro. Para este caso en particular se tiene lo siguiente:

- El sistema tolera el 155 % de la incertidumbre modelada.

⁴`stabmarg` es una estructura donde `UpperBound` es la cota superior del margen de estabilidad, `LowerBound` es la cota inferior y `DestabilizingFrequency` es la frecuencia en la que ocurre la inestabilidad para algún conjunto de parámetros.

Parámetro	Sensibilidad	Aumentar en un 25% cambia k_m en
M	44 %	- 11 %
m	20 %	- 25 %
b	7 %	- 2 %
I	7 %	- 2 %
g	1 %	0 %
l	37 %	- 9 %

Tabla 2: Sensibilidad asociada a cada parámetro

- La combinación desestabilizante de los parámetros representa un 500 % de la incertidumbre modelada.
- La sensibilidad con respecto a cada parámetro está registrada en la Tabla 2

Es posible que se quiera observar el comportamiento de $\mu(i\omega)$ frecuencia por frecuencia para ver cómo se comporta la estabilidad del modelo frente a cambios en las frecuencias. Esta información frecuencial se encuentra en la estructura `info`, donde `info.Frequency` son las frecuencias usadas en el análisis, e `info.MussvBnds` tiene las cotas inferiores y superiores del valor singular estructurado μ . Para ver esto gráficamente, se puede usar el comando `semilogx(info.MussvBnds)`. El resultado se observa en la Fig. 4, donde es evidente que las condiciones de estabilidad robusta se cumplen para todas las frecuencias analizadas.

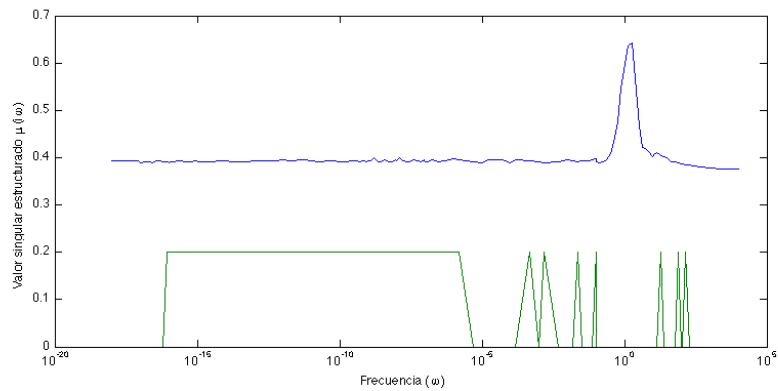


Fig 4: Valor singular estructurado vs. frecuencia (en escala logarítmica)

5. Conclusiones

El análisis de estabilidad robusta es una herramienta poderosa para evaluar la eficiencia de cualquier controlador cuando existe incertidumbre en los parámetros o en las dinámicas del modelo utilizado para el diseño. No sólo tiene una fuerte fundamentación teórica sino que además su aplicación es sencilla gracias a paquetes de software como MATLAB.

Se realizó el análisis de sensibilidad de un péndulo invertido con incertidumbre utilizando el toolbox de control de MATLAB, ilustrando las capacidades de esta herramienta e interpretando los resultados obtenidos.

Referencias

- [1] R. Dorf and H. Bishop. *Modern Control Systems*. Pearson Education, Inc., 2008.
- [2] L. Lublin, S. Grocott, and M. Athans. \mathcal{H}_2 (LQG) and \mathcal{H}_∞ control. In W. Levine, editor, *The Control Handbook*, pages 651–661. CRC Press, 1999.
- [3] The Mathworks, Inc. *MATLAB Documentation*, 2010.
- [4] S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control*. John Wiley & Sons, 1996.
- [5] K. Zhou, C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice Hall, 1996.