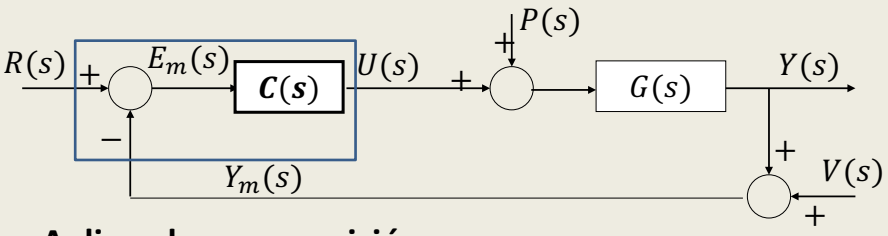
Diseño de PID basado en la respuesta en frecuencia: compromiso óptimo entre robustez, rapidez y amplificación del ruido

Roberto Sanchis Llopis
Universitat Jaume I

ÍNDICE

INTRODUCCIÓN. CONTROLADOR PID

Bucle de realimentación de la salida:



Aplicando superposición:

$$Y(s) = \frac{C(s)G(s)}{1 + C(s)G(s)} R(s) + \frac{G(s)}{1 + C(s)G(s)} P(s) - \frac{C(s)G(s)}{1 + C(s)G(s)} V(s)$$

Polos del sistema controlado: $1 + C(s)G(s) = 0$

INTRODUCCIÓN. CONTROLADOR PID

Controlador PID con filtro:

$$\boldsymbol{C(s) = K_p \left(1 + \frac{1}{T_i s} + \frac{T_d s}{1 + \frac{T_d}{N} s} \right)}$$

- Parámetros que lo definen: $\boldsymbol{K_p, T_i, T_d, N}$

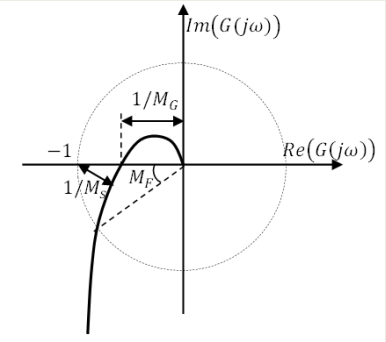
- Problema de diseño o ajuste del PID:
  selección de los valores de $\boldsymbol{K_p, T_i, T_d, N,}$ teniendo en cuenta
  - ***Robustez***
  - ***Rapidez***
  - ***Amplificación del ruido***

!!!! Objetivos en conflicto ¡¡¡¡

AJUSTE EN FRECUENCIA DE PID

Se basa en conseguir que el diagrama de Nyquist de C(s)G(s) cumpla requisitos de robustez:

- Margen de fase
- Margen de ganancia
- Margen de sensibilidad



AJUSTE EN FRECUENCIA DE PID EN LIBROS CLÁSICOS (DORF)

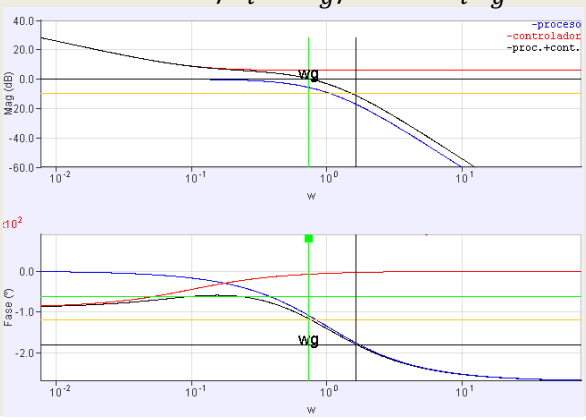
Ajuste de PI (compensador de retardo)

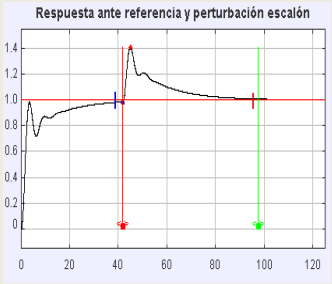
- Se basa en conseguir un margen de fase.
- Se busca ω_g donde el sistema tiene la fase necesaria con una corrección: -180 +M_f+5 a 12
- Se fija el cero (1/Ti) entre 1/6 y 1/10 de ω_g
- Se calcula la ganancia para que el margen de fase sea el deseado (el módulo de CG sea 1 a la frecuencia ω_g).

AJUSTE EN FRECUENCIA DE PID EN LIBROS CLÁSICOS (DORF)

Ajuste de PI (compensador de retardo)

El cero se pone lejos de ω_g para que el retardo de fase "no moleste": $1/T_i = \omega_g/10 \rightarrow T_i\omega_g = 10$





AJUSTE EN FRECUENCIA DE PID EN LIBROS CLÁSICOS (DORF)

Ajuste de PID con filtro (compensador de adelanto-retardo)

- Se decide el adelanto y se pone el término derivador con filtro para máximo adelanto en ω_g.
- Se fija el término de retardo igual que antes, a 1/6 a 1/10 de ω_g para que "no moleste".

$$T_i\omega_g = 10$$

NUEVO AJUSTE EN FRECUENCIA DE PID

Ajuste de PI en frecuencia

- Se usa el parámetro de ajuste: $\boldsymbol{a = T_i \omega_g}$
- Si se fija $\boldsymbol{a}$ hay un único PI con el Mf deseado.
  - El argumento del controlador depende solo de $\boldsymbol{a}$:
    $$\arg\left(C(j\omega_g)\right) = \arg\left(K_p(1+\frac{1}{j\omega_g T_i})\right) = \arg\left(1+\frac{1}{ja}\right) = f(a)$$
  - Se busca $\boldsymbol{\omega_g}$ donde el sistema tiene la fase exacta necesaria:
    $$\arg\left(G(j\omega_g)\right) = -180º + M_{F,des} - \arg\left(C(j\omega_g)\right) \rightarrow \boldsymbol{\omega_g}$$
  - Se calcula $\boldsymbol{T_i = \frac{a}{\omega_g}}$
  - Se calcula la ganancia $\boldsymbol{K_p}$ para que $|G(j\omega_g)||C(j\omega_g)| = 1$

Sanchis, Roberto , Romero, Julio A. and Balaguer, Pedro. 'Tuning of PID controllers based on simplified single parameter optimisation', International Journal of Control, 2010.

NUEVO AJUSTE EN FRECUENCIA DE PID

Ajuste de PI en frecuencia

- ¿Cuál es el valor adecuado de $\boldsymbol{a = T_i \omega_g}$?
- Se busca $\boldsymbol{a}$ para rapidez máxima→ IAE mínima
- Asegurando robustez (M_F y M_G , o M_S)

NUEVO AJUSTE EN FRECUENCIA DE PID

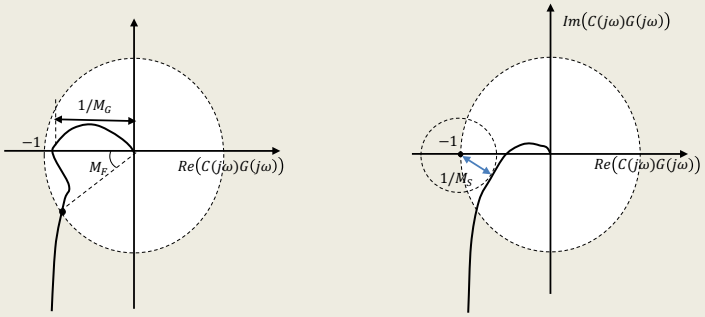
Ajuste de PI en frecuencia

3 formas de definir la robustez (M_F y M_G , o M_S)

$M_F = M_{F,req}$
$M_G \geq M_{G,req}$

$M_F = M_{F,req}$
$M_S \leq M_{S,req}$

$M_S = M_{S,req}$



NUEVO AJUSTE EN FRECUENCIA DE PID

Ajuste de PI en frecuencia

- Se busca a para rapidez máxima→ IAE mínima
- Asegurando robustez (M_F y M_G , o M_S)

Minimizar a IAE (o ISE o ITAE)

Sujeto a: $M_F = M_{F,d}$

$M_G \geq M_{G,d}$

!!!!!! El controlador PI más rápido ¡¡¡¡¡

NUEVO AJUSTE EN FRECUENCIA DE PID

Ajuste de PI en frecuencia

•Se busca a para rapidez máxima→ IAE mínima

• Asegurando robustez (M_F y M_G, o M_S)

Minimizar IAE (o ISE o ITAE)

a

Sujeto a: $M_F = M_{F,d}$

$M_S \leq M_{S,d}$

!!!!!! El controlador PI más rápido ¡¡¡¡¡

NUEVO AJUSTE EN FRECUENCIA DE PID

Ajuste de PI en frecuencia

•Se busca a para rapidez máxima→ IAE mínima

• Asegurando robustez (M_F y M_G, o M_S)

Minimizar IAE (o ISE o ITAE)

a

Sujeto a: $M_S = M_{S,d}$

!!!!!! El controlador PI más rápido ¡¡¡¡¡

NUEVO AJUSTE EN FRECUENCIA DE PID

Ajuste de PI en frecuencia

¿Y la amplificación de ruido de alta frecuencia?

- En el PI la amplificación de alta frecuencia es $\mathbf{K_p}$
- Si el PI más rápido amplifica mucho el ruido → se reduce α respecto del óptimo para hacerlo más lento y reducir $\mathbf{K_p}$.
- Si el PI más rápido amplifica poco el ruido → se puede usar un PID para respuesta más rápida.

NUEVO AJUSTE EN FRECUENCIA DE PID

Ajuste de PI en frecuencia

Herramienta de diseño de PI

- Calcula el PI para un $\mathbf{M_f}$ y un valor de $\boldsymbol{\alpha}$
- Calcula automáticamente el PI óptimo
- Si la robustez se define con M_s, itera con M_f hasta que M_s es el deseado.

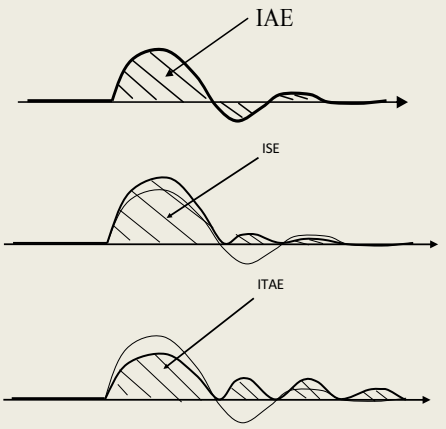
NUEVO AJUSTE EN FRECUENCIA DE PID

Ajuste de PI en frecuencia

IAE

ISE

ITAE



$$IAE \approx IE = \frac{1}{K_i}$$

Si poco oscilatorio

NUEVO AJUSTE EN FRECUENCIA DE PID

Ajuste de PI

Ejemplo

$$G(s) = \frac{3}{(1+0.5s)(1+0.25s)}$$

- Diseñar controlador P para un Mf=60º:
- Diseñar PI con a=10.
- Diseñar PI con a=0.5.
- Buscar PI que minimiza IAE (ISE o ITAE).

NUEVO AJUSTE EN FRECUENCIA DE PID

Ajuste de PI

Ejemplo

$$G(s) = \frac{3}{(1+0.5s)(1+0.25s)}e^{-s}$$

Diseñar controlador PI para:

- $M_f = 60º$, $M_G \geq 9.5$ dB
- Buscar PI que minimiza IAE.

NUEVO AJUSTE EN FRECUENCIA DE PID

Ajuste de PID en frecuencia

- Se usa el parámetro de ajuste: $\boldsymbol{a = T_i\omega_g}$
- Si se fija $\boldsymbol{a}$, $\beta = \frac{T_i}{T_d}$, y N, hay un único PID con el M_f deseado.
  - El argumento del controlador depende solo de $\boldsymbol{a}$:
    $$\arg\left(C(j\omega_g)\right) = \arctan\left(\frac{a(N\beta+1)}{N\beta - a^2(1+N)}\right) - \arctan\left(\frac{a}{N\beta}\right) - 90º = f(a)$$
  - Se busca $\boldsymbol{\omega_g}$ donde el sistema tiene la fase exacta necesaria:
    $$\arg\left(G(j\omega_g)\right) = -180º + M_{F,des} - \arg\left(C(j\omega_g)\right) \rightarrow \boldsymbol{\omega_g}$$
  - Se calcula $\boldsymbol{T_i = \frac{a}{\omega_g}}$, $T_d = \frac{T_i}{\beta}$
  - Se calcula la ganancia $\boldsymbol{K_p}$ para que $|G(j\omega_g)||C(j\omega_g)| = 1$

NUEVO AJUSTE EN FRECUENCIA DE PID

Ajuste de PID en frecuencia

- ¿Cuál es el valor adecuado de $\boldsymbol{a = T_i \omega_g}$?

- Se busca a para rapidez máxima→ IAE mínima

- Asegurando robustez (M_F y M_G , o M_S)

NUEVO AJUSTE EN FRECUENCIA DE PID

Ajuste de PID en frecuencia

¿Y la amplificación de ruido?

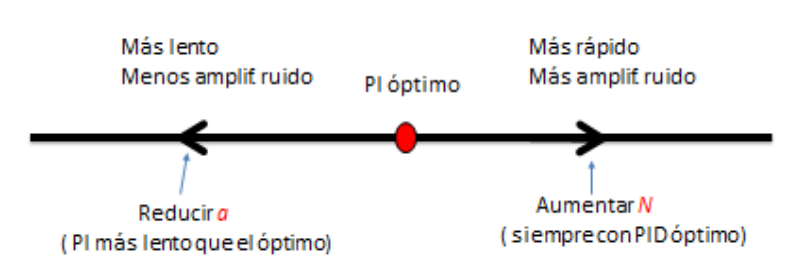
- En el PID la amplificación del ruido es **$K_P(1+N)$**

- Si el PID más rápido amplifica mucho el ruido → se reduce **N** y se recalcula el óptimo.

NUEVO AJUSTE EN FRECUENCIA DE PID

Ajuste de PID en frecuencia

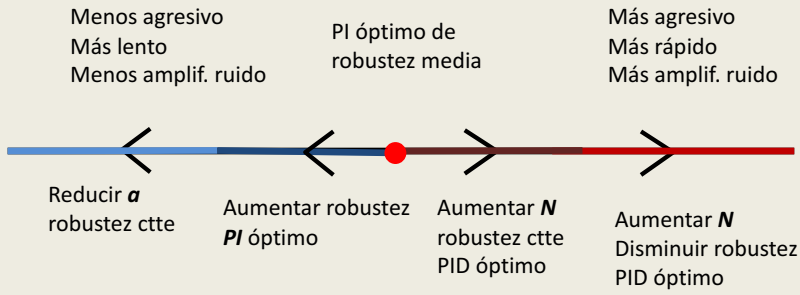
Línea de ajuste para robustez fija



NUEVO AJUSTE EN FRECUENCIA DE PID

Ajuste de PID en frecuencia

Línea de ajuste de una dimensión



NUEVO AJUSTE EN FRECUENCIA DE PID

Ajuste de PID en frecuencia

¿Y cómo se fija el valor de $\beta = \frac{T_i}{T_d}$?

- Se toma $\frac{T_i}{T_d}$=4 por defecto.
- En algunos casos, reducir o aumentar β permite mejorar las prestaciones

NUEVO AJUSTE EN FRECUENCIA DE PID

Ajuste de PID en frecuencia

Herramienta de diseño de PID

- Calcula el PID para un **M_f** ,un valor de ***α,*** de **β** y de ***N.***
- Calcula automáticamente el PID óptimo
- Si la robustez se define con M_s, itera con M_f hasta que M_s es el deseado.

NUEVO AJUSTE EN FRECUENCIA DE PID

Ajuste de PID

Ejemplo

$$G(s) = \frac{1}{(1+s)^3}$$

Diseñar un PID para conseguir:

- Mf = 60º, Mg ≥ 9.5dB.
- Amplificación del ruido de alta frecuencia menor de 10.
- IAE mínima.

NUEVO AJUSTE EN FRECUENCIA DE PID

Ajuste de PID

Ejemplo

$$G(s) = \frac{1}{(1+s)^3}$$

Diseñar un PID para conseguir:

- Ms = 1.5.
- $e_v \leq 1 \rightarrow \boldsymbol{K_i} = \frac{1}{e_v G(0)} \geq 1$
- Amplificación del ruido de alta frecuencia mínima.

NUEVO AJUSTE EN FRECUENCIA DE PID

Ajuste de PID

Ejemplo $G(s) = \dfrac{2}{(1+10s)(1+s)^2}$

Diseñar un PID para conseguir:

- Mf = 60º, Mg ≥ 9.5dB
- ts,98 ≈ 10 seg ⟶ $\omega_g \approx \dfrac{16.8 - 0.2M_f}{t_{s,98}} = 0.48\ rad/s$
- Amplificación del ruido de alta frecuencia mínima.

COMPROMISO DE 3 FACTORES EN PID

Los 3 factores de diseño en frecuencia:

- **Robustez**. Se puede fijar de tres formas:
  - $M_F = M_{F,d}, M_G \geq M_{G,d}$: Si no hay retardo, M_G no afecta.
  - $M_F = M_{F,d}$, $M_S \leq M_{S,d}$
  - $M_S = M_{S,d}$
- **Rapidez de respuesta**.
  - Para optimizar: **IAE**, ISE, ITAE, $\boldsymbol{K_i} = \dfrac{1}{e_v G(0)}$
  - Para fijar: $\boldsymbol{K_i}$, $\boldsymbol{t_{s98}}$, (se puede aproximar: $\boldsymbol{\omega_g} \approx \dfrac{16.8-0.2M_f}{t_{s,98}}$)
- **Amplificación del ruido**. $C(\infty) = K_p$ para el PI, $C(\infty) = K_p(1+N)$ para el PID ó el PD.

COMPROMISO DE 3 FACTORES EN PID

Estrategia de diseño 1: maximizar rapidez.

Fijar robustez y $C(\infty)$, y maximizar rapidez (Ki, IAE, ISE o ITAE).

1. Se calcula el PI más rápido. Si $C(\infty)$ adecuada **FIN.**
2. Si $C(\infty)$ excesiva → reducir α hasta que $C(\infty)$ adecuada. **FIN**
3. Si $C(\infty)$ pequeña → calcular PID más rápido con N=0.1. Aumentar N y recalcular PID más rápido hasta que $C(\infty)$ adecuada. **FIN**

COMPROMISO DE 3 FACTORES EN PID

Estrategia 2: minimizar amplificación del ruido.

Fijar robustez y rapidez (t_{s98} o K_i), y minimizar $C(\infty)$.

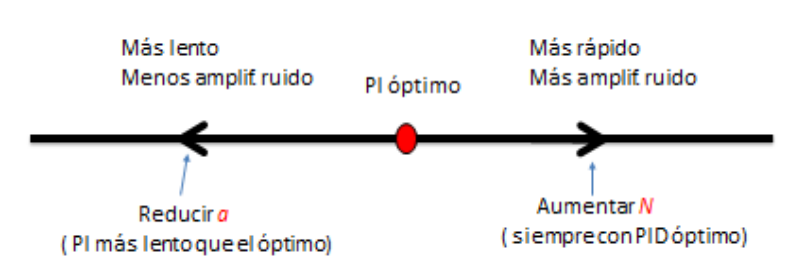
$$\omega_g \approx \frac{16.8 - 0.2M_f}{t_{s,98}}$$

1. Se calcula el PI más rápido. Si rapidez adecuada **FIN**
2. Si rapidez excesiva → reducir α hasta rapidez adecuada. **FIN**
3. Si rapidez insuficiente → calcular PID más rápido con N=0.1. Aumentar N y recalcular PID más rápido hasta que rapidez adecuada. **FIN**

COMPROMISO DE 3 FACTORES EN PID

Estrategias de diseño 1 y 2.

Se empieza en el punto medio de la línea de ajuste (PI óptimo) y se mueve a izquierda o derecha:



COMPROMISO DE 3 FACTORES EN PID

Estrategia 3: maximizar robustez:

Fijar $C(\infty)$ y rapidez (t_{s98} o K_i), y maximizar robustez.

1. Se fija robustez máxima de la tabla:

M_f (º)	75	70	65	60	55	50	45	40
$M_{G,min}$ (dB)	12.7	11.6	10.5	9.5	8.6	7.7	6.8	6

M_S	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2

2. Se calcula el PID más rápido con ese valor de $C(\infty)$.
3. Si rapidez suficiente → Ese PID es la solución. **FIN**
4. Si rapidez insuficiente → tomar siguiente valor de robustez de la tabla y volver al paso 2

COMPROMISO DE 3 FACTORES EN PID

Ejemplo

$$G(s) = \frac{2}{(1+10s)(1+s)^2}$$

Diseñar un PID para conseguir:

- Ms mínimo
- $e_v \leq 0.5 \rightarrow \boldsymbol{K_i} = \frac{1}{e_v G(0)} \geq 1$
- Amplificación del ruido de alta frecuencia menor de 20.

HERRAMIENTAS DE PID BASADAS EN JAVA

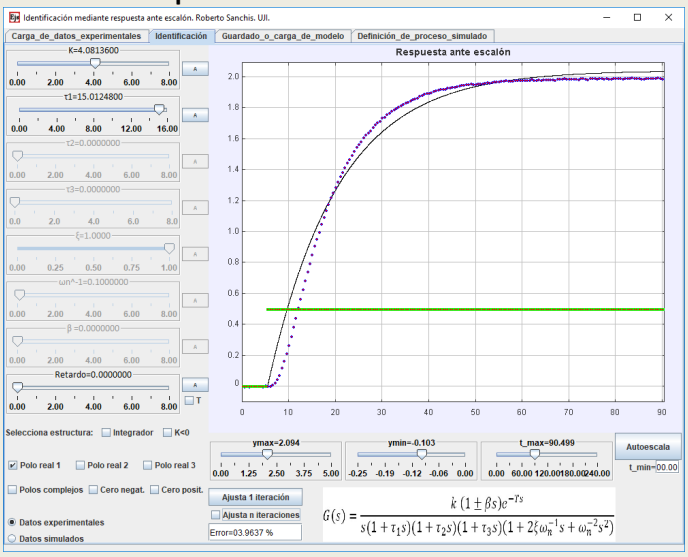
Herramientas libres para PID

- Identificación experimental
- Ajuste experimental de PID
- Diseño basado en modelo de PID
- Firmware Arduino y aplicación interfaz para PC
- Identificación en frecuencia
- Otras aplicaciones (Send on Delta)

https://sites.google.com/a/uji.es/freepidtools/

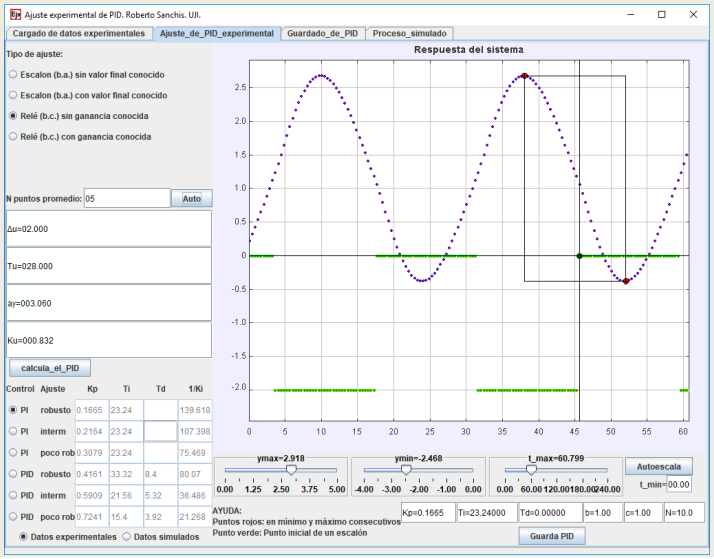
HERRAMIENTAS DE PID BASADAS EN JAVA

Identificación experimental



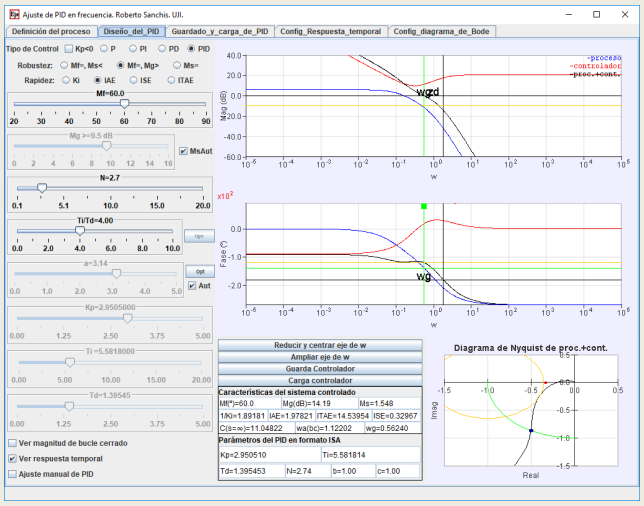
HERRAMIENTAS DE PID BASADAS EN JAVA

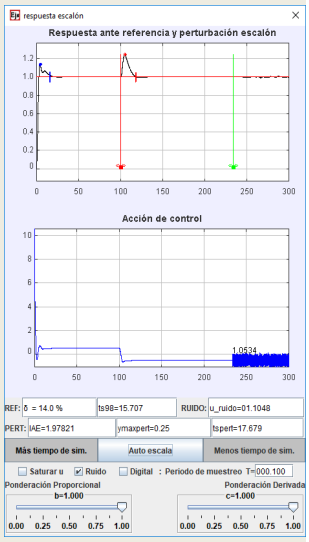
Ajuste experimental de PID



HERRAMIENTAS DE PID BASADAS EN JAVA

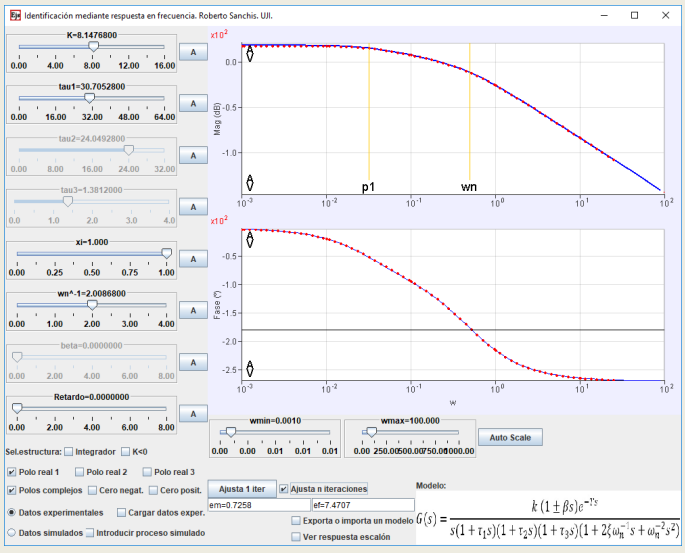
Diseño de PID basado en modelo





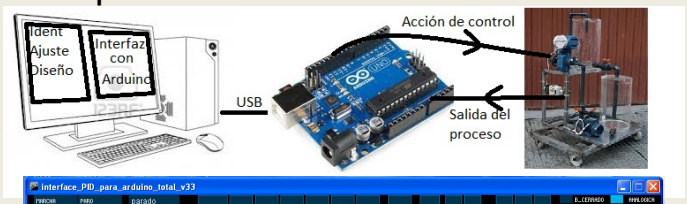
HERRAMIENTAS DE PID BASADAS EN JAVA

Identificación en frecuencia



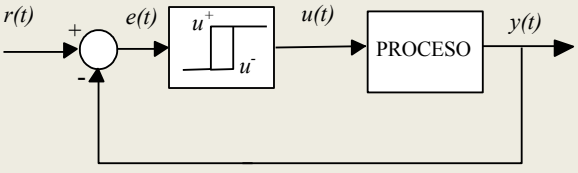
HERRAMIENTAS DE PID BASADAS EN JAVA

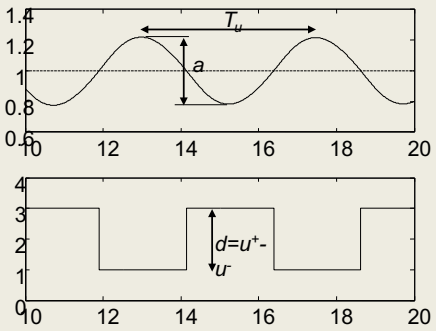
Interfaz para Arduino



AUTOAJUSTE DE PID

Método clásico de relé:





AUTOAJUSTE DE PID

Autoajuste de PI

- Se aproxima el diagrama de Bode (magnitud y fase) por una línea recta:
  - Experimento relé, y relé con retardo→ 2 puntos del diagrama de Bode.
  - Aproximación de Bode con líneas rectas.
- Se diseña el PI que maximiza K_i con $M_f = M_{f,r}$, $M_G \geq M_{G,r}$
- Si ω_g está lejos de los 2 puntos, se calcula un 3er punto con relé y más retardo, y se recalcula el PI.

Julio Ariel Romero, Roberto Sanchis, Pedro Balaguer. PI and PID auto-tuning procedure based on simplified single parameter optimization. Journal of Process Control. 21 (2011) 840–851.

AUTOAJUSTE DE PID

Autoajuste de PID

- Se fija $T_i/T_d=4$, y se elige N.
- Se procede exactamente igual que con el PI.

Parámetros que definen el autoajuste

- $\boldsymbol{M_{f,r}}$ (45º - 60º)
- $\boldsymbol{M_{G,r}}$ (6 dB - 9 dB)
- $\boldsymbol{N}$, parámetro de filtro del derivador (5 – 10)
- Amplitud del relé
- Histéresis del relé (algo mayor que el ruido)
- Fase del segundo punto (-130º)
- Factor de condición de cálculo de 3er punto (1)

AUTOAJUSTE DE PID

Autoajuste de PID

- Alternativamente, se elije la robustez, y se elige C(∞), y se diseña el PID para maximizar K_i.

AUTOAJUSTE DE PID

Algoritmo de autoajuste

1. Experimento relé sin retardo (ampl. μ, hist. h)

$$M_1 = \pi A_1/4\mu \qquad \phi_1 = -\pi - arg\left(\frac{4\mu}{\pi A_1}\left(\sqrt{1-\left(\frac{h}{A_1}\right)^2} - j\frac{h}{A_1}\right)\right)$$

2. Experimento relé con retardo $L_2 = ((-\phi_1 + \phi)\phi_1)/\phi\omega_1$

$$M_2 = \pi A_2/4\mu \qquad \phi_2 = L_2\omega_2 - \pi - arg\left(\frac{4\mu}{\pi A_2}\left(\sqrt{1-\left(\frac{h}{A_2}\right)^2} - j\frac{h}{A_2}\right)\right)$$

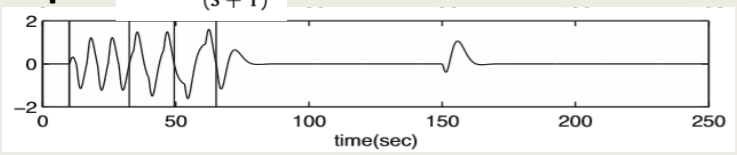
3. Ajusta PID con Bode línea recta
4. Si $\log(\omega_2/\omega_{gt1}) > r\log(\omega_1/\omega_2)$, exp. relé con $L_3 = \frac{(\phi_2-\phi_1)\log\left(\frac{\omega_1}{\omega_{gt1}}\right)}{\omega_{gt1}\log\left(\frac{\omega_1}{\omega_2}\right)}$

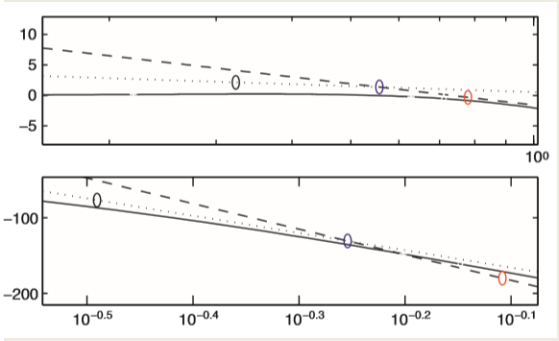
$$M_3 = \pi A_3/4\mu \qquad \phi_3 = L_3\omega_3 - \pi - arg\left(\frac{4\mu}{\pi A_3}\left(\sqrt{1-\left(\frac{h}{A_3}\right)^2} - j\frac{h}{A_3}\right)\right)$$

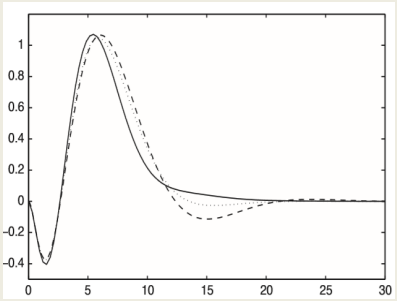
5. Ajusta PID con Bode nueva línea recta

AUTOAJUSTE DE PID

Ejemplo $G_8(s) = \dfrac{1-2s}{(s+1)^3}$

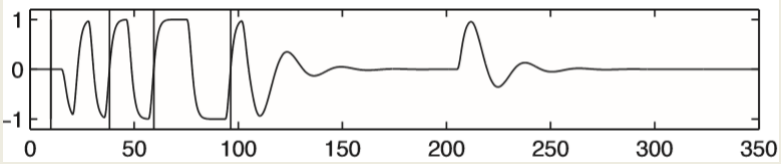


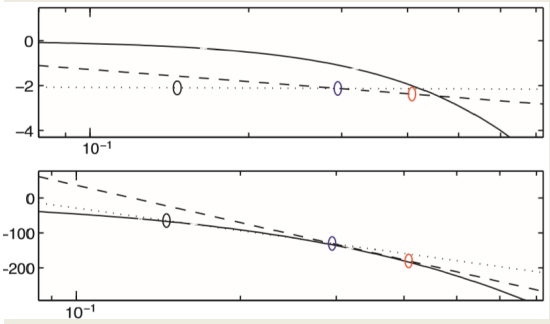


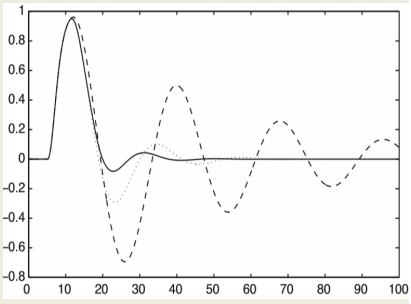


AUTOAJUSTE DE PID

Ejemplo $G_2(s) = \dfrac{1}{(s+1)^3}e^{-5s}$



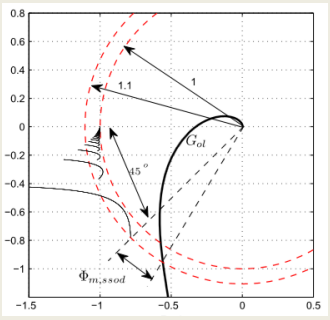


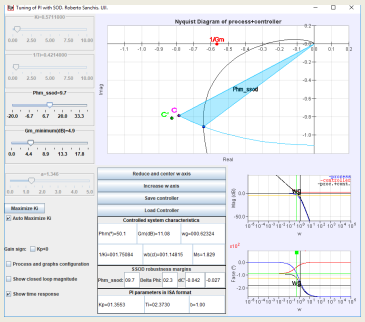


PID BASADO EN EVENTOS

PID con muestreo Send on Delta

- Se utiliza la función descriptiva para predecir ciclos límites, y para diseñar el PID y evitarlos



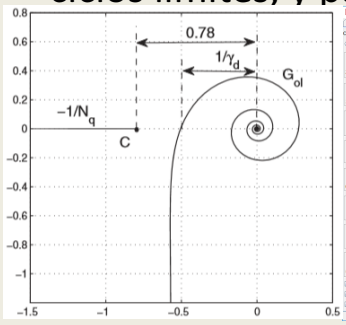


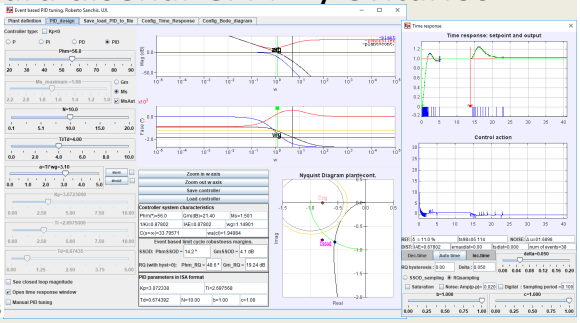
- Válido para procesos muy filtrantes

PID BASADO EN EVENTOS

PID con muestreo RQ

- Se utiliza la función descriptiva para predecir ciclos límites, y para diseñar el PID y evitarlos



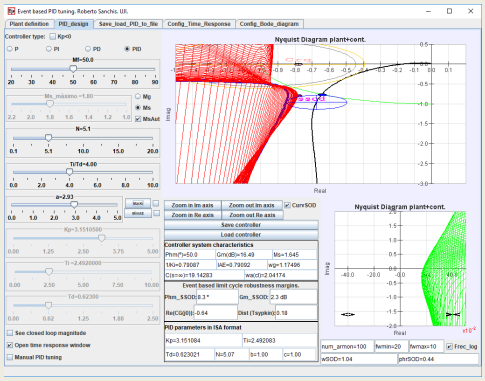


- Válido para procesos muy filtrantes

PID BASADO EN EVENTOS

PID con muestreo Send on Delta

- Se utiliza la aproximación de Tsypkin para predecir ciclos límites, y para diseñar el PID y evitarlos



- Válido para todo tipo de procesos

REFERENCIAS

- Sanchis, Roberto , Romero, Julio A. and Balaguer, Pedro. 'Tuning of PID controllers based on simplified single parameter optimisation', International Journal of Control, 2010. DOI: 10.1080/00207179.2010.495162. URL: http://dx.doi.org/10.1080/00207179.2010.495162
- Julio Ariel Romero, Roberto Sanchis, Pedro Balaguer. PI and PID auto-tuning procedure based on simplified single parameter optimization. Journal of Process Control. 21 (2011) 840–851.
- Julio Ariel Romero Pérez, Roberto Sanchis Llopis. Benchmark para la Evaluación de Algoritmos de Auto-ajuste de Controladores PID. REVISTA IBEROAMERICANA DE AUTOMATICA E INFORMATICA INDUSTRIAL. Num. 1. Vol. 8 . pp. 112-117. 2011.
- Julio Ariel Romero Pérez, Roberto Sanchis Llopis. A new method for tuning PI controllers with symmetric send-on-delta sampling strategy. ISA TRANSACTIONS . Vol. 64 . pp. 161-173. 2016.
- Julio Ariel Romero Pérez, Roberto Sanchis Llopis. Tuning and robustness analysis of event-based PID controllers under different event generation strategies. INTERNATIONAL JOURNAL OF CONTROL . pp. 1-38. 2017.

REFERENCIAS

- Roberto Sanchis Llopis, Julio Ariel Romero Pérez. A software tool for the design and simulation of PID with event based sampling. 3rd International Conference on Event-Based Control, Communication and Signal Processing (EBCCSP 2017). Madeira (Portugal). 24-05-2017.
- Julio Ariel Romero Pérez, Roberto Sanchis Llopis. Analysis of a simple rule for tuning SSOD based PIDs. 2nd International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP). 2016. Cracovia (Polònia). 13-06-2016.
- Julio Ariel Romero Pérez, Roberto Sanchis Llopis, Ignacio Peñarrocha Alós. A simple rule for tuning Event-based PID controllers with Symmetric Send-On Delta sampling strategy. 2014 IEEE International Conference on Emerging Technology and Factory Automation (ETFA 2014). Barcelona (Espanya). 16-09-2014.
- Roberto Sanchis Llopis, Ignacio Peñarrocha Alós, Julio Ariel Romero Pérez. Enfoque unificado del diseño de PID mediante el lugar de las raíces y en frecuencia. XXXV Jornadas de Automática. Valencia (Espanya). 03-09-2014.
- Julio Ariel Romero Pérez, Roberto Sanchis Llopis, Ignacio Peñarrocha Alós. Ajuste de controladores PID basados en eventos por cuantificación y cruce de niveles. XXXV Jornadas de Automática. Valencia (Espanya). 03-09-2014.
- Roberto Sanchis Llopis, Silvia Estupiña Ariño. Herramientas de hardware y software libre para la identificación experimental, el diseño y la implementación de controladores PID. XXXIV Jornadas de Automática. Terrassa (Espanya). 04-09-2013.