

Implementació digital d'un PID

Teresa Escobet Canal

27 de març de 2013

Índex

1	Consideracions preliminars	1
1.1	Mostreig	1
1.2	Àliasing	2
1.3	Prefiltre	3
2	Discretització d'un PID	3
2.1	Algoritme PID	3
2.1.1	Filtrat de la derivada	4
2.1.2	Ponderació del set point	4
2.1.3	Windup	4
2.2	Discretització d'un controlador PID	5
2.2.1	Acció Proporcional	5
2.2.2	Acció Integral	5
2.2.3	Acció Derivada	6
2.2.4	PID discret	6
2.3	Sintonia de paràmetres d'un PID	7

La majoria dels controladors avui en dia s'implementen en un computador. En aquesta fitxa es discuteixen els aspectes pràctics relacionats amb la seva implementació <http://http://www.cds.caltech.edu/~murray/courses/cds101/fa02/caltech/astrom-ch6.pdf>.

1 Consideracions preliminars

1.1 Mostreig

Quan s'implementa un controlador en un computador, la lectura de les entrades analògiques i l'enviament de la sortida es fan a un cert període de mostreig. Això

és un inconvenient comparat amb les implementacions analògiques, ja que el mostreig introdueix retards en els llaços de control. Quan un computador s'utilitza per implementar una llei de control, la seqüència ideal és la següent:

1. Esperar la interrupció del rellotge
2. Llegir l'entrada analògica
3. Calcular el senyal de control
4. Enviar la sortida analògica
5. Actualitzar les variables del controlador
6. Retorn a 1

Si el senyal analògic es llegeix amb un període h , el retard és en promig de $h/2$. En general el temps de càlcul és molt petit comparat amb el període de mostreig, per tant el retard total és d'aproximadament $h/2$. De tots formes, la majoria dels sistemes de control no organitzen el càlcul de la forma indicada i per tant el retard introduït, si el programa no s'organitza correctament, potser de varies mostres.

1.2 Àliasing

El mecanisme de mostreig de senyals introdueix alguns fenòmens inesperats, que cal tenir en compte per a la correcta implimentació digital de controladors PID i altres, en general. Per explicar aquest concepte, considerem el senyal

$$s(t) = \cos(n\omega_s t \pm \omega t) \quad (1)$$

i

$$s_a(t) = \cos(\omega_s t) \quad (2)$$

a on $\omega_s = 2\pi h$ [rad/seg] és la freqüència de mostreig. Fent la discretització de la funció cosinus, s'obté que s a cada instant de mostreig $kh, k = 0, 1, 2, \dots$ té la propietat

$$s(kh) = \cos(nkh\omega_s \pm \omega kh) = \cos(nkh\omega_s) = s_a(\omega kh) \quad (3)$$

Les senyals s i s_a tenen els mateixos valors en els instants de mostreig. Això implica que no es pot saber de quina senyal es tracta si sols es coneixen els seus valors als instants indicats. La senyal s_a és de totes formes una àlies de la senyal s . Aquest fenomen s'il·lustra en la figura. Una conseqüència de l'efecte aliasing és que una pertorbació d'alta freqüència després de ser mostrejada pot aparèixer com a una senyal de baixa freqüència.

1.3 Prefiltre

Com s'ha vist, les altes freqüències, que en el cas de controladors analògics normalment s'eliminen amb un prefiltre passa baixos, a causa de l'aliasing apareixen com a senyals de baixa freqüència en l'ampla de banda del sistema de control mostrejat. Per evitar aquests problemes, cal emprar prefiltres analògics que eliminin les senyals d'alta freqüència. Per aquest motiu al prefiltre se l'anomena filtre anti-aliasing. S'acostumen emprar en les aplicacions normals filtres de tipus Butterworth de $2n$ ordre, en aplicacions crítiques s'utilitzen filtres de $3r$ ordre. Com seleccionar l'ampla de banda del prefiltre es mostra en el següent exemple.

Exemple 1. Selecció de l'ampla de banda del prefiltre Considerem que es volen atenuar senyals amb un factor de 16 a la meitat del període de mostreig. Si l'ampla de banda del prefiltre és ω_b i la freqüència de mostreig és ω_s tenim que

$$\left(\frac{\omega_s}{2\omega_b}\right)^2 = 16 \quad (4)$$

Per tant,

$$\omega = \frac{1}{8}\omega_s \quad (5)$$

Cal tenir present que la dinàmica del prefiltre acostuma a ser significant i cal tenir-la en compte en el disseny del sistema de control.

2 Discretització d'un PID

Per implementar una llei de control en temps continu en un computador digital cal aproximar les accions derivades i integrals que apareixen a la llei de control. Hi ha diferents formes de realitzar aquesta aproximació. Comentar que la discretització d'un PID continu donarà bons resultats sempre que el període de mostreig no sigui crític, en cas contrari caldrà emprar eines de disseny digital.

2.1 Algoritme PID

L'algoritme PID es descriu com:

$$u(t) = K_P \Delta e(t) + K_I \int_0^t e(p) dp + K_D \frac{de(t)}{dt} \quad (6)$$

a on y és la variable mesurada, y_{sp} la variable referència o *set-point*, u el senyal de control i e l'error de control $e = y_{sp} - y$. A mesura que ha avançat la tecnologia s'han desenvolupat nous algoritmes PID, sempre amb l'objectiu de millorar les prestacions del llaç de control. De les modificacions del control PID estàndard en destaquem algunes per la seva importància.

2.1.1 Filtrat de la derivada

La diferenciació és molt sensible al soroll. A la pràctica cal limitar el guany de l'acció derivada a freqüències altes. La implementació del terme derivada es fa

$$D = \frac{sK_P T_d}{1 + sT_d/N} Y \quad (7)$$

a on T_D és el temps d'acció derivatiu i T_D/N ens fixa la constant de temps del sistema de 1r ordre i limita a $K_P N$ el guany de la derivada. Els valors típics de N són de 8 a 20. La relació que hi ha entre el guany derivatiu i el temps d'acció derivatiu és: $K_D = K_P T_D$.

2.1.2 Ponderació del set point

Quan s'utilitza la representació típica del PID 6 succeeix que degut a l'acció proporcional un canvi graó a la referència comporta un impuls a l'acció de control. Aquest fet en algunes aplicacions no és desitjable. Per evitar-ho s'acostuma a poderar la referència tant en l'acció proporcional com en la derivada. En aquest cas l'algoritme PID queda de la forma

$$u(t) = K_P \left(by_{sp}(t) - y(t) + \frac{1}{T_I} \int_0^t e(p) dp + T_D \left(c \frac{dy_{sp}(t)}{dt} - \frac{dy(t)}{dt} \right) \right) \quad (8)$$

a on b i c són paràmetres addicionals. El terme integral ha d'integrar l'error de realimentació per assegurar que en règim permanent s'assoleixin els valors desitjats.

Ambdós controladors 6 i 8 tenen el mateix comportament quan hi ha pertorbacions de càrrega, independentment dels valors de b , mentre que quan hi ha canvis de consigna el sobre excés disminueix per $b = 0$. El paràmetre c s'acostuma a posar a zero per eliminar les transicions degudes a canvis sobtats de la referència.

2.1.3 Windup

Si bé hi ha fenòmens en el control de sistemes que es poden explicar mitjançant la teoria de control lineal, hi ha alguns comportaments no lineals que cal tenir en compte. Windup és un d'aquest fenòmens que s'origina degut a la interacció de l'acció integral amb les saturacions. Tots els actuadors tenen limitacions: el motor té una velocitat limitada, la vàlvula no pot funcionar fora del seu rang de treball, etc. Ja que el sistema de control pot funcionar en un ampli rang d'operació, pot ser que la variable de control porti l'actuador als seus límits. Quan això succeeix, el llaç de control es trenca i el sistema passa a treballar en llaç obert, ja que l'actuador es manté en els seus límits independentment de la variable de procés. Si s'utilitza un controlador amb acció integral, l'error es continua integrant i, en conseqüència, el terme integral pot esdevenir molt gran. En aquest cas, caldrà un error de signa oposat durant un llarg període de temps abans de retornar a la normalitat. La conseqüència és que un controlador amb acció integral pot originar transitoris grans quan l'actuador entra en saturació.

S'utilitzen diferents mètodes per evitar el windup:

- Limitació de la variació del set point.
- Saturació de l'acció integral
- Algoritmes incrementals
- Estimació de l'acció integral i seguiment (back calculation and tracking)

La implementació d'aquests mètodes es descriu a 2.2.4

2.2 Discretització d'un controlador PID

2.2.1 Acció Proporcional

El terme proporcional és

$$P(t) = K_P (by_{sp}(t) - y(t)) \quad (9)$$

Aquest terme s'implementa senzillament substituint la variable continua per la variable mostrejada

$$P(t_k) = K_P (by_{sp}(t_k) - y(t_k)) \quad (10)$$

a on t_k indica els instant de mostreig, es a dir el temps en que el computador llegeix l'entrada analògica.

2.2.2 Acció Integral

El terme integral és

$$I(t) = \frac{K_P}{T_I} \int_0^t e(p) dp \quad (11)$$

o equivalentment

$$\frac{dI(t)}{dt} = \frac{K_P}{T_I} e(t) \quad (12)$$

La derivada es pot aproximar per la diferència cap endavant (forward difference)

$$\frac{I(t_{k+1}) - I(t_k)}{h} = \frac{K_P}{T_I} e(t_k) \quad (13)$$

Obtenint-se l'equació recursiva del terme integral

$$I(t_{k+1}) = I(t_k) + \frac{K_P}{T_I} e(t_k) \quad (14)$$

2.2.3 Acció Derivada

L'acció derivada,7, es pot expressar en el cas de $c = 0$ com

$$\frac{K_D}{N} \frac{dD}{dt} + D = -K_P T_D \frac{dy}{dt} \quad (15)$$

Aquesta equació es pot aproximar per la diferenciació enrere

$$\frac{T_D}{N} \frac{D(t_k) - D(t_{k-1})}{h} + D(t_k) = -K_P T_D \frac{y(t_k) - y(t_{k-1})}{h} \quad (16)$$

I es pot escriure com

$$D(t_k) = \frac{T_D}{T_D + Nh} D(t_{k-1}) - \frac{K_P T_D N}{T_D + Nh} (y(t_k) - y(t_{k-1})) \quad (17)$$

L'avantatge d'utilitzar una diferencial cap enrere és que el paràmetre $T_D/(T_D + Nh)$ es mou en el rang de 0 a 1 per tots els valors dels paràmetres, fet que garanteix que l'equació en diferències sigui estable.

2.2.4 PID discret

Un controlador PID discret es pot aproximar per

$$P(t_k) = K_P (by_{sp}(t_k) - y(t_k)) \quad (18)$$

$$e(t_k) = y_{sp}(t_k) - y(t_k) \quad (19)$$

$$D(t_k) = \frac{T_D}{T_D + Nh} (D(t_{k-1}) - K_P N (y(t_k) - y(t_{k-1}))) \quad (20)$$

$$u(t_k) = P(t_k) + I(t_k) + D(t_k) \quad (21)$$

$$I(t_{k+1}) = I(t_k) + \frac{K_P h}{T_I} e(t_k) \quad (22)$$

L'*algoritme incremental* es realitzaria incorporant en l'algoritme anterior l'equació incremental

$$\Delta u(t_k) = u(t_k) - u(t_{k-1}) = \Delta P(t_k) + \Delta I(t_k) + \Delta D(t_k) \quad (23)$$

Una forma de reduir el windup del integrador és incorporant la saturació a l'acció integral. L'algoritme de l'acció integral es modifica segons:

$$I(t_{k+1}) = I(t_k) + \frac{K_P h}{T_I} e(t_k); \quad (24)$$

$$\text{if } (u(t_k + 1) > u_{max}) I(t_{k+1}) = I(t_k); \quad (25)$$

$$\text{elseif } (u(t_k + 1) < u_{min}) I(t_{k+1}) = I(t_k); \quad (26)$$

2.3 Sintonia de paràmetres d'un PID

La sintonia consisteix en ajustar els paràmetres del PID. Hi ha dos mètodes de sintonia:

- Mètodes empírics: permeten calcular els paràmetres del regulador sense necessitat de disposar d'un model matemàtic de la planta.
- Mètodes analítics: cal disposar d'un model matemàtic de la planta a controlar. S'utilitzen tècniques temporals i/o freqüencials per calcular els paràmetres del regulador.

Per sintonitzar els paràmetres d'una planta cal seguir les següents etapes:

- Identificació de la planta: estimació de les característiques dinàmiques de la planta o procés a controlar.
- Criteri d'optimització: definir els objectius d'ajust (reposta transitòria, característiques en règim permanent,..) que es volen aconseguir com a resultat d'aplicar un controlador.
- Ajust dels paràmetres: a partir dels resultats anteriors, del controlador que es vol utilitzar, s'obtenen els paràmetres del regulador.

En el cas d'estudi utilitzarem la tècnica del relé. Aquesta tècnica consisteix en controlar la planta mitjançant un relé, segons la representació de la figura 1. S'obté la resposta de la figura 2.

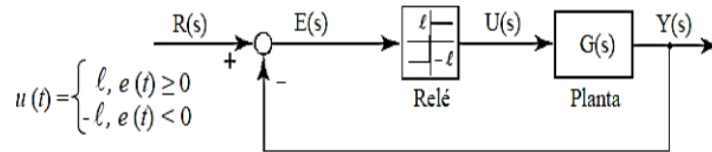


Figura 1: Diagrama de blocs utilitzant el mètode del relé

Observant el comportament del sistema al ser controlat per un relé, es calcula el guany d'oscil·lació aplicant l'equació 27.

$$K_u = \frac{4d}{\pi a} \quad (27)$$

El mètode de Ziegler-Nichols en enllaç tancat ens proporciona els valors del PID en funció del guany calculat K_u i el període d'oscil·lació T_u . El valor dels paràmetres s'obté a partir dels valors de la taula de la figura 3.

Quan hi ha soroll a la mesura s'utilitza un relé amb histéresis. El mètode de Ziegler-Nichols és una tècnica molt utilitzada en sistemes estables i quan s'utilitza per ajustar sistemes integradors dona molta oscil·lació.

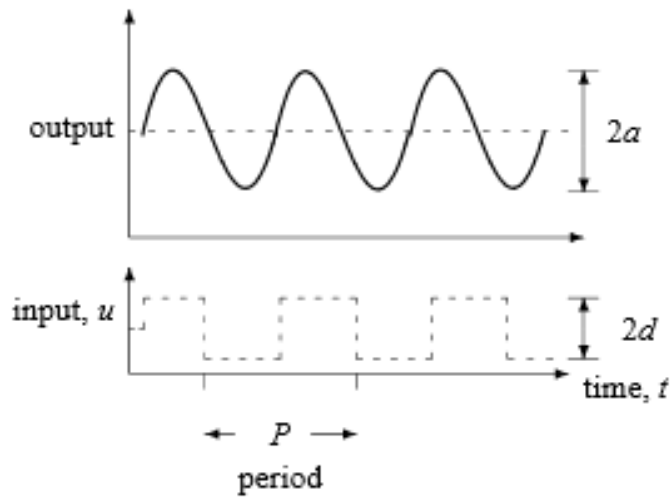


Figura 2: Senyal d'entrada al procés i sortida

Controlador	K_c	T_i	T_d
P	$0.5K_u$		
PI	$0.4K_u$	$0.8T_u$	
PID	$0.8K_u$	$0.5T_u$	$0.12T_u$

Figura 3: Ajust de paràmetres Ziegler-Nichols

En el cas de sistemes integradors, com és el cas d'estudi, hi ha la tècnica de Pec-harromán (2000) que proposa ajustar un PI del tipus de la forma que es mostra a les equacions 18 amb els següents valors:

- $b = 0.5$
- $K_P = 0.049 * K_u$
- $T_I = 2.826 * T_u$

Recordeu que l'equivalència entre el temps d'acció integral T_I i el guany de l'acció integral K_I és de: $K_I = K_P / T_I$.

Les tècniques de sintonia les podeu trobar al Handbook of PI and PID Controller Tuning Rules de Aidan O'Dwyer, 2006.

Índex alfabètic

tableofcontents, 1