

PID控制器

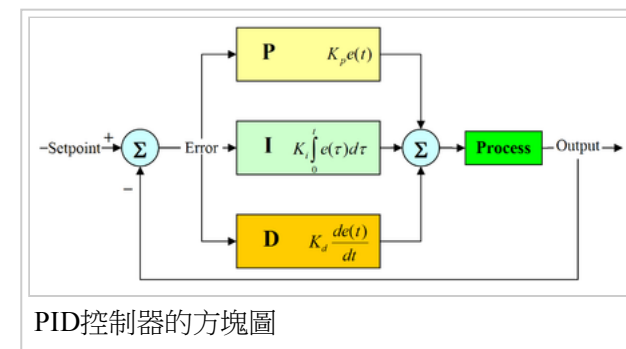
維基百科，自由的百科全書

PID控制器（比例-積分-微分控制器），由比例單元**P**、積分單元**I**和微分單元**D**組成^[1]。透過**K_p**，**K_i**和**K_d**三個參數的設定。**PID**控制器主要適用於基本上線性，且動態特性不隨時間變化的系統。

PID控制器是一個在工業控制應用中常見的反饋迴路部件。這個控制器把收集到的數據和一個參考值進行比較，然後把這個差別用於計算新的輸入值，這個新的輸入值的目的是可以讓系統的數據達到或者保持在參考值。**PID**控制器可以根據歷史數據和差別的出現率來調整輸入值，使系統更加準確而穩定。

PID控制器的比例單元**P**、積分單元**I**和微分單元**D**分別對應目前誤差、過去累計誤差及未來誤差。若是不知道受控系統的特性，一般認為**PID**控制器是最適用的控制器^[2]。藉由調整**PID**控制器的三個參數，可以調整控制系統，設法滿足設計需求。控制器的響應可以用控制器對誤差的反應快慢、控制器過衝的程度及系統震盪的程度來表示。不過使用**PID**控制器不一定保證可達到系統的最佳控制，也不保證系統穩定性。

有些應用只需要**PID**控制器的部份單元，可以將不需要單元的參數設為零即可。因此**PID**控制器可以變成**PI**控制器、**PD**控制器、**P**控制器或**I**控制器。其中又以**PI**控制器比較常用，因為**D**控制器對回授雜訊十分敏感，而若沒有**I**控制器的話，系統不會回到參考值，會存在一個誤差量。



目錄

- 1 反饋迴路基礎
- 2 歷史及應用
- 3 理論
 - 3.1 比例控制項
 - 3.1.1 穩態誤差
 - 3.2 積分控制項
 - 3.3 微分控制項
- 4 參數調試
 - 4.1 穩定性
 - 4.2 最佳性能
 - 4.3 各方法的簡介
 - 4.4 人工調整

- 4.5 齊格勒－尼科爾斯方法
- 4.6 PID調試軟體
- 5 PID控制的限制
 - 5.1 線性
 - 5.2 雜訊對微分器的影響
- 6 PID演算法的修改
 - 6.1 積分飽和
 - 6.2 PI控制器
 - 6.3 不動作區
 - 6.4 設定值的步階變化
 - 6.5 無衝擊運轉
- 7 串級PID控制器
- 8 其他PID的形式及其表示法
 - 8.1 理想的PID及標準形PID
 - 8.2 倒數增益
 - 8.3 只針對過程變數進行微分控制
 - 8.4 只針對過程變數進行比例控制
 - 8.5 PID控制器的拉氏轉換
 - 8.6 PID的極零點對消
 - 8.7 串級型或交互型
 - 8.8 離散化的控制器
- 9 偽代碼
- 10 參見
- 11 注釋
- 12 參考文獻
- 13 外部連結

反饋迴路基礎

PID迴路是要自動實現一個操作人員用量具和控制旋鈕進行的工作，這個操作人員會用量具測系統輸出的結果，然後用控制旋鈕來調整這個系統的輸入，直到系統的輸出在量具上顯示穩定的需求的結果，在舊的控制文檔里，這個過程叫做「復位」行為，量具被稱為「測量」，需要的結果被稱為「設定值」而設定值和測量之間的差別被稱為「誤差」。

一個控制迴路包括三個部分：

1. 系統的傳感器得到的測量結果

2. 控制器作出決定
3. 通過一個輸出設備來作出反應

控制器從傳感器得到測量結果，然後用需求結果減去測量結果來得到誤差。然後用誤差來計算出一個對系統的糾正值來作為輸入結果，這樣系統就可以從它的輸出結果中消除誤差。

在一個PID迴路中，這個糾正值有三種算法，消除目前的誤差，平均過去的誤差，和透過誤差的改變來預測將來的誤差。

比如說，假如利用水箱在為植物提供水，水箱的水需要保持在一定的高度。可以用傳感器來檢查水箱裡水的高度，這樣就得到了測量結果。控制器會有一個固定的用戶輸入值來表示水箱需要的水面高度，假設這個值是保持65%的水量。控制器的輸出設備會連在由馬達控制的水閥門上。打開閥門就會給水箱注水，關上閥門就會讓水箱裡的水量下降。這個閥門的控制信號就是控制變量。

PID控制器可以用來控制任何可被測量及可被控制變量。比如，它可以用來控制溫度、壓力、流量、化學成分、速度等等。汽車上的巡航定速功能就是一個例子。

一些控制系統把數個PID控制器串聯起來，或是連成網絡。這樣的話，一個主控制器可能會為其他控制輸出結果。一個常見的例子是馬達的控制。控制系統會需要馬達有一個受控的速度，最後停在一個確定的位置。可由一個子控制器用來管理速度，但是這個子控制器的速度是由控制馬達位置的主控制器來管理的。

連合和串聯控制在化學程序控制系統中相當常見。

歷史及應用

PID控制器可以追溯到1890年代的調速器設計^{[2][3]}。PID控制器是在船舶自動操作系統中漸漸發展。1911年Elmer Sperry開發的控制器是最早期PID型控制器的其中之一^[4]，而第一個發表PID控制器理論分析論文的是俄裔美國工程師尼古拉斯·米諾爾斯基（Minorsky 1922）。米諾爾斯基當時在設計美國海軍的自動操作系統，他的設計是基於對舵手的觀察，控制船舶不只是依目前的誤差，也考慮過去的誤差以及誤差的變化趨勢^[5]，後來米諾爾斯基也用數學的方式加以推導^[6]。他的目的是在於穩定性，而不是泛用的控制，因此大幅的簡化了問題。比例控制可以在小的擾動下有穩定性，但無法消除穩態誤差，因此加入了積分項，後來也加入了微分項。

當時在新墨西哥號戰艦上進行測試，利用控制器控制舵的角速度，利用PI控制器可以角度誤差維持在 $\pm 2^\circ$ 以內，若加上D控制，角度誤差維持在 $\pm 1/6^\circ$ ，比最好的舵手還要好^[7]。

不過因為海軍人員的抗拒，海軍那時候未使用這套系統，在1930年代也有其他人作出類似的研究。

在自動控制發展的早期，用機械設備來實現PID控制，是由槓桿、彈簧、阻尼及質量組成，多半會用壓縮氣體驅動。氣動控制器還一度是工業上的標準。



PID控制理論是由觀察舵手的動作而來

電子的類比控制器可以用電晶體、真空管、電容器及電阻器組成。許多複雜的電子系統中常會包括PID控制，例如磁碟的讀寫頭定位、電源供應器的電源條件、甚至是現代地震儀的運動偵測線路。現代電子控制器已大幅的被這些利用單晶片或FPGA來實現的數位控制器所取代。

現代工業使用的PID控制器多半會用PLC或有安裝面板的數位控制器來實現。軟體實現的好處是相對低廉，配合PID實現方式調整的靈敏度很大。在工業鍋爐、塑膠射出機械、燙金機及包裝行業中都會用到PID控制。

變化的電壓輸出可以用PWM來實現，也就是固定週期，依要輸出的量去調整週期中輸出高電位的時間。對於數位系統，其時間比例有可能是離散的，例如週期是二秒，高電位時間設定單位為0.1秒，表示可以分為20格，精度5%，因此存在一量化誤差，但只要時間解析度夠高，就會有不錯的效果。

理論

PID是以它的三種糾正算法而命名。受控變數是三種算法（比例、積分、微分）相加後的結果，即為其輸出，其輸入為誤差值（設定值減去測量值後的結果）或是由誤差值衍生的信號。若定義***u*(*t*)**為控制輸出，PID演算法可以用下式表示：

$$\mathbf{u}(t) = \mathbf{M}\mathbf{V}(t) = K_p \mathbf{e}(t) + K_i \int_0^t \mathbf{e}(\tau) d\tau + K_d \frac{d}{dt} \mathbf{e}(t)$$

其中

K_p：比例增益，是調適參數

K_i：積分增益，也是調適參數

K_d：微分增益，也是調適參數

e：誤差=設定值（SP）- 回授值（PV）

t：目前時間

τ：積分變數，數值從0到目前時間***t***

用更專業的話來講，PID控制器可以視為是頻域系統的濾波器。在計算控制器最終是否會達到穩定結果時，此性質很有用。如果數值挑選不當，控制系統的輸入值會反覆振盪，這導致系統可能永遠無法達到預設值。

PID控制器的一般轉移函數是：

$$H(s) = \frac{K_d s^2 + K_p s + K_i}{s + C},$$

其中***C***是一個取決於系統帶寬的常數。

比例控制項

比例控制考慮當前誤差，誤差值 and 一個正值的常數 K_p （表示比例）相乘。 K_p 只是在控制器的輸出和系統的誤差成比例的時候成立。比如說，一個電熱器的控制器的比例尺範圍是 10°C ，它的預定值是 20°C 。那麼它在 10°C 的時候會輸出100%，在 15°C 的時候會輸出50%，在 19°C 的時候輸出10%，注意在誤差是0的時候，控制器的輸出也是0。

比例控制的輸出如下：

$$P_{\text{out}} = K_p e(t)$$

若比例增益大，在相同誤差量下，會有較大的輸出，但若比例增益太大，會使系統不穩定。相反的，若比例增益小，若在相同誤差量下，其輸出較小，因此控制器會較不敏感的。若比例增益太小，當有干擾出現時，其控制信號可能不夠大，無法修正干擾的影響。

穩態誤差

比例控制在誤差為0時，其輸出也會為0。若要讓受控輸出為非零的數值，就需要有一個穩態誤差或偏移量^[a]。穩態誤差和比例增益成正比，和受控系統本身的增益成反比。若加入一偏置，或是加入積分控制，可以消除穩態誤差。

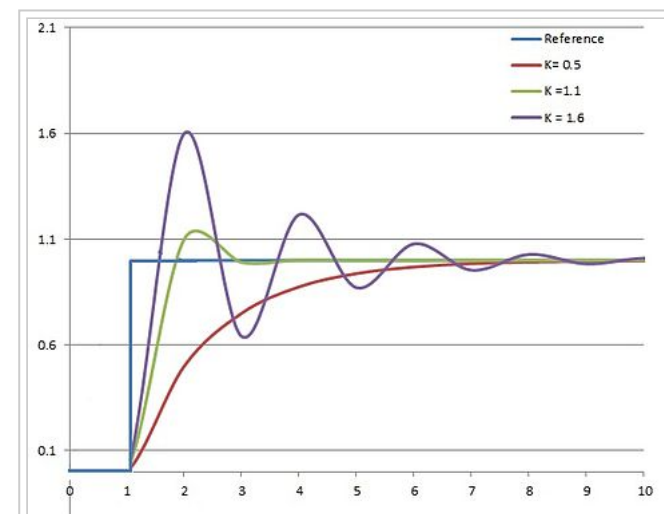
積分控制項

積分控制考慮過去誤差，將誤差值過去一段時間和（誤差和）乘以一個正值的常數 K_i 。 K_i 從過去的平均誤差值來找到系統的輸出結果和預定值的平均誤差。一個簡單的比例系統會震盪，會在預定值的附近來回變化，因為系統無法消除多餘的糾正。通過加上負的平均誤差值，平均系統誤差值就會漸漸減少。所以，最終這個PID迴路系統會在設定值穩定下來。

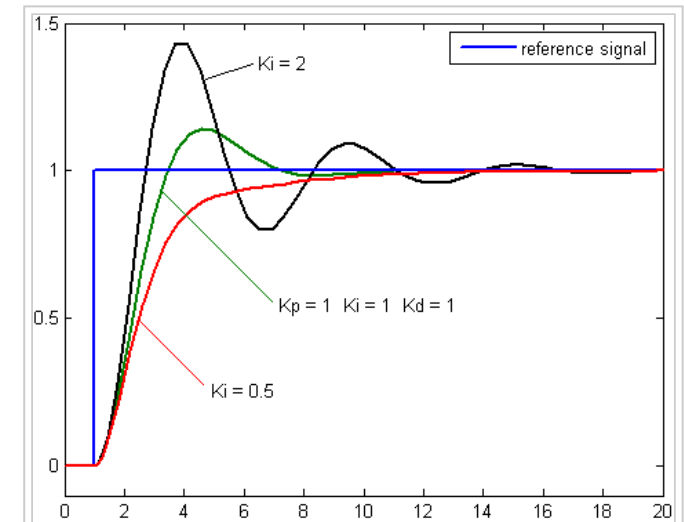
積分控制 \lceil 的輸出如下：

$$I_{\text{out}} = K_i \int_0^t e(\tau) d\tau$$

積分控制會加速系統趨近設定值的過程，並且消除純比例控制器會出現的穩態誤差。積分增益越大，趨近設定值的速度越快，不過因為積分控制會累計過去所有的誤差，可能會使回授值出現過衝的情形。



不同比例增益 K_p 下，受控變數對時間的變化（ K_i 和 K_d 維持定值）



不同積分增益 K_i 下，受控變數對時間的變化（ K_p 和 K_d 維持定值）

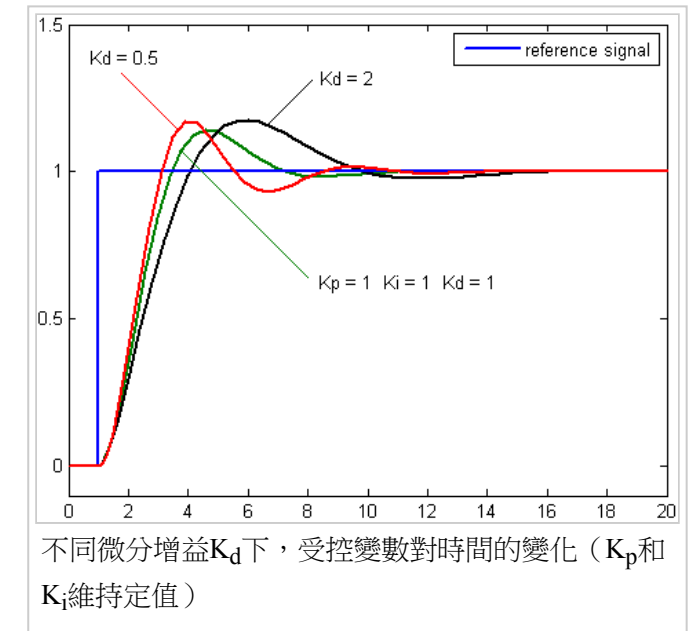
微分控制項

微分控制考慮將來誤差，計算誤差的一階導，並和一個正值的常數 K_d 相乘。這個導數的控制會對系統的改變作出反應。導數的結果越大，那麼控制系統就對輸出結果作出更快速的反應。這個 K_d 參數也是PID被稱為可預測的控制器的原因。 K_d 參數對減少控制器短期的改變很有幫助。一些實際中的速度緩慢的系統可以不需要 K_d 參數。

微分控制的輸出如下：

$$D_{\text{out}} = K_d \frac{d}{dt} e(t)$$

微分控制可以提昇整定時間及系統穩定性^{[8][9]}。不過因為純微分器不是因果系統，因此在PID系統實現時，一般會為微分控制加上一個低通濾波器以限制高頻增益及雜訊^[10]。實務上較少用到微分控制，估計PID控制器中只有約20%有用到微分控制^[10]。



參數調試

PID的參數調試是指透過調整控制參數（比例增益、積分增益/時間、微分增益/時間）讓系統達到最佳的控制效果。穩定性（不會有發散性的震盪）是首要條件，此外，不同系統有不同的行為，不同的應用其需求也不同，而且這些需求還可能會互相衝突。

PID只有三個參數，在原理上容易說明，但PID參數調試是一個困難的工作，因為要符合一些特別的準則，而且PID控制有其限制存在。歷史上有許多不同的PID參數調試方式，包括齊格勒－尼科爾斯方法等，其中也有一些已申請專利。

PID控制器的設計及調試在概念上很直覺，但若有多個（且互相衝突）的目標（例如高穩定性及快速的暫態時間）都要達到的話，在實際上很難完成。PID控制器的參數若仔細的調試，會有很好的效果，相反的，若調適不當，效果會很差。一般初始設計常需要不斷的電腦模擬，並且修改參數，一直達到理想的性能或是可接受的妥協為止。

有些系統有非線性的特性，若在無載下調試的參數可能無法在滿載下動作，可以利用增益規劃的方式進行修正（在不同的條件下選用不同的數值）。

穩定性

若PID控制器的參數未挑選妥當，其控制器輸出可能是不穩定的，也就是其輸出發散，過程中可能有震盪，也可能沒有震盪，且其輸出只受飽和或是機械損壞等原因所限制。不穩定一般是因為過大增益造成，特別是針對延遲時間很長的系統。

一般而言，PID控制器會要求響應的穩定，不論程序條件及設定值如何組合，都不能出現大幅振盪的情形，不過有時可以接受臨界穩定的情形。

最佳性能

PID控制器的最佳性能可能和針對過程變化或是設定值變化有關，也會隨應用而不同。

兩個基本的需求是調整能力（regulation，干擾拒絕，使系統維持在設定值）及命令追隨（設定值變化下，控制器輸出追隨設定值的反應速度）。有關命令追隨的一些準則包括有上昇時間及整定時間。有些應用可能因為安全考量，不允許輸出超過設定值，也有些應用要求在到達設定值過程中的能量可以最小化。

各方法的簡介

有許多種調試PID控制器參數的方法，最有效的方式多半是建立某種程序，再依不同參數下的動態特性來調試參數。相對而言人工調試其效率較差，若是系統的響應時間到數分鐘以上，更可以看出人工調試效率的不佳。

調試方法的選擇和是否可以暫時將控制迴路「離線」有關，也和系統的響應時間有關。離線是指一個和實際使用有些不同的條件（例如不加負載），而且控制器的輸出只需考慮適適，不需考慮實際應用。在線調試是在實際應用的條件，控制器的輸出需考慮實際的系統。若控制迴路可以離線，最好的調試方法是對系統給一個步階輸入，量測其輸出對時間的關係，再用其響應來決定參數。

| 選擇調試方式 | | |
|------------|--|--------------------------|
| 方法 | 優點 | 缺點 |
| 人工調試 | 不需要數學，可以在線調試 | 需要有經驗的工程師 |
| 齊格勒－尼科爾斯方法 | 被證實有效的方法，可以在線調試 | 會影響製程，需要試誤，得到的參數可能使響應太快 |
| 軟體工具 | 調適的一致性，可以在線調試或離線調試，可以配合計算機自動設計，包括閥及感測器的分析，可以在下載前進行模擬，可以支援非穩態（NSS）的調試 | 需要成本或是訓練 ^[1] |
| Cohen–Coon | 好的程序模型 | 需要一些數學，需離線調試，只對一階系統有良好效果 |

人工調整

若需在系統仍有負載的情形進行調試（線上調試），有一種作法是先將 K_i 及 K_d 設為零，增加 K_p 一直到迴路輸出震盪為止，之後再將 K_p 設定為「1/4 振幅衰減」（使系統第二次過衝量是第一次的1/4）增益的一半，然後增加 K_i 直到一定時間後的穩態誤差可被修正為止。不過若 K_i 可能會造成不穩定，最後若有需要，可以增加 K_d ，並確認在負載變動後迴路可以夠快的回到其設定值，不過若 K_d 太大會造成響應太快及過衝。一般而言快速反應的PID應該會有輕微的過衝，只是有些系統不允許過衝，因此需要將回授系統調整為過阻尼系統，而 K_p 比造成震盪 K_p 的一半還要小很多。

調整PID參數對系統的影響如下

| 調整方式 | (on) 上升時間 | 超調量 | 安定時間 | 穩態誤差 | 穩定性 ^[10] |
|---------|-----------|-----|-------|--------|---------------------|
| ↑ K_p | 減少↓ | 增加↑ | 小幅增加↗ | 減少↓ | 變差↓ |
| ↑ K_i | 小幅減少↘ | 增加↑ | 增加↑ | 大幅減少↓↓ | 變差↓ |
| ↑ K_d | 小幅減少↘ | 減少↓ | 減少↓ | 變動不大→ | 變好↑ |

齊格勒－尼科爾斯方法

齊格勒－尼科爾斯方法是另一種啟發式的調試方式，由John G. Ziegler和Nathaniel B. Nichols在1940年代導入，一開始也是將 K_i 及 K_d 設定為零，增加比例增益直到系統開始振盪為止，當時的增益稱為 K_u ，而振盪週期為 P_u ，即可用以下的方式計算增益：

齊格勒－尼科爾斯方法

| 控制器種類 | K_p | K_i | K_d |
|-------|-----------|--------------|------------|
| P | $0.50K_u$ | - | - |
| PI | $0.45K_u$ | $1.2K_p/P_u$ | - |
| PID | $0.60K_u$ | $2K_p/P_u$ | $K_pP_u/8$ |

PID調試軟體

大部份現代的工業設備不再用上述人工計算的方式調試，而是用PID調試及最佳化軟體來達到一致的效果。軟體會收集資料，建立模型，並提供最佳的調試結果，有些軟體甚至可以用參考命令的變化來進行調試。

數學的PID調試會將脈衝加入系統，再用受控系統的頻率響應來設計PID的參數。若是響應時間要數分鐘的系統，建議用數學PID調試，因為用試誤法可能要花上幾天才能找到可讓系統穩定的參數。最佳解不太容易找到，有些數位的迴路控制器有自我調試的程序，利用微小的參考命令來計算最佳的調試值。

也有其他調試的公式，是依不同的性能準則所產生。許多有專利的公式已嵌入在PID調試軟體及硬體模組中^[12]。

一些先進的PID調試軟體也可以有演算法可以在動態的情形下調整PID迴路，這類軟體會先將程序建模，給微擾量，再根據響應計算參數。

PID控制的限制

PID控制可以應用在許多控制問題，多半在大略調整參數後就有不錯的效果，不過有些應用下可能反而會有差的效果，而且一般無法提供最佳控制。PID控制的主要問題是在於其為回授控制，係數為定值，不知道受控系統的資訊，因此其整理性能常常是妥協下的結果。在沒有受控系統模型的條件下，PID控制最佳的控制器^[2]，但若配合系統模型，可以有進一步的提昇。

當PID控制器單獨使用時，若因應用需求，需調整PID迴路增益使控制系統不會過衝，其效果有可能很差。PID控制器的缺點還包括無法處理受控系統的非線性、需在反應時間及調整率之間妥協、無法針對參數的變動而反應（例如系統在暖機後特性會改變）、以及大擾動下的波形落後。

PID控制器最顯著的提昇是配合前饋控制，加入有關系統的資訊，只用PID控制器來控制誤差。另外，PID控制器也有一些小幅的改善方式，例如調整參數（增益規劃或是依性能進行適應性的調整）、提昇性能（提高取樣率、精度及準度，若有需要加入低波濾波器），或是用多個串接的PID控制器。

線性

PID控制器常見的問題是在於其線性且對稱的特性，若應用在一些非線性的系統，其效果可能會有變化。以暖通空調中常見的溫度控制，可能是採用主動加熱（用加熱器加熱），但冷卻是使用被動冷卻（不加熱，自然冷卻），其冷卻速度比加熱速度慢很多，輸出若有過衝，下降速度很慢，因此PID控制需調整為不會過衝的過阻尼，以減少或避免過衝，但這也延長了整定時間，使性能變差。

雜訊對微分器的影響

微分器的問題在於對量測或程序產生的高頻雜訊會有放大效果，因此會對輸出造成大幅的變動。因此真實的控制器不會有理想的微分器，只有一個有限頻寬的微分器或高通濾波器。一般為了移除高頻的雜訊，會在量測時加入低通濾波器，若低通濾波器和微分器對消，濾波效果也就受限了，因此低雜訊的量測設備相當重要。實務上可以使用中值濾波器，調昇濾波效率及實際上的性能^[13]。有時可以將微分器關閉，對控制性能的影響不大，此時稱為PI控制器。

PID演算法的修改

基本的PID演算法在一些控制應用的條件下有些不足，需進行小幅的修改。

積分飽和

積分飽和是理想PID演算法實現時常見的問題。若設定值有大的變動，其積分量會有大幅的變化，大到輸出值被上下限限制而飽和，因此系統會有過衝，而且即使誤差量符號改變，積分量變小，但輸出值仍被上下限限制，維持在上限（或下限），因此輸出看似沒有變化，系統仍會持續的過衝，一直要到輸出值落在上下限的範圍內，系統的回授量才會開始下降。此問題可以用以下方式處理：

- 在程序變數離開可控制範圍時，暫停積分。
- 讓積分值限制在一個較小的上下範圍內。

- 重新計算積分項，使控制器輸出維持上下限之間的範圍內^[14]。

PI控制器

PI控制器（比例-積分控制器）是不用微分單元的PID控制器。

控制器的輸出為

$$K_P \Delta + K_I \int \Delta dt$$

其中 Δ 為設定值SP和量測值PV的誤差：

$$\Delta = SP - PV.$$

PI控制器可以用Simulink或Xcos之類的軟體進行建模，方式是使用「flow chart」圖框，其中用以下的拉氏運算子：

$$C = \frac{G(1 + \tau s)}{\tau s}$$

其中

$$G = K_P = \text{比例增益}$$

$$G/\tau = K_I = \text{積分增益}$$

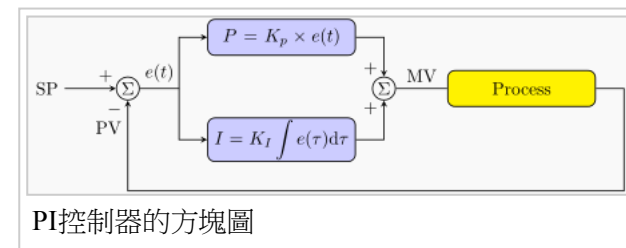
G 值的選擇需在減少過衝以及增加安定時間之間取捨。

微分單元對輸入中的高頻信號格外敏感，PI控制器因為沒有微分單元，在訊號雜訊大時，在穩態時會更加穩定。但對狀態快速變化的反應較慢，因此相較於調適到最佳值的PID控制器，PI控制器會較慢到達設定值，受干擾後也比較慢恢復到正常值。

不動作區

許多PID迴路是控制機械元件（例如閥）。機械保養是一筆可觀的費用，磨損會使得機械在有輸入信號時出現靜摩擦或是不動作區，都會導致控制性能的下降。機械損耗的速度主要和設備多常改變其狀態有關。若磨損是主要考量的話，PID迴路可以有輸出的不動作區以減少輸出狀態的改變。若變化小，仍在不動作區內，讓控制器的輸出維持上一次的值。變化要大到超過不動作區，實際的狀態才會隨之變化。

設定值的步階變化



若系統的設定值有步階變化，比例單元和微分單元也會有對應的變化，特別是微分單元對於步階變化的輸出特別的大，因此有些PID演算法會配合以下的修改來處理設定值的變化。

設定值斜坡變化

此修改方式下，設定值會用線性或是一階濾波的方式，由原始值變到新的值，避免因為步階變化產生的不連續。

只對程序變數（回授量）微分

此修改下，PID控制器只針對量測的程序變數（PV）微分，不對誤差微分。程序變數是實際的物理量，較不易有瞬間的變化，而誤差可能因為設定值的步階變化而有瞬間變化。這也是一種簡單的設定值加權法。

設定值加權

設定值加權分別調整在比例單元及微分單元中的誤差量，誤差量的設定值乘以一個0到1之間的加權，積分單元的誤差量需使用真實的設定值，以避免穩態誤差。這兩個參數不影響對負載變化及量測雜訊的響應，可以提昇對設定點變化的響應。

無衝擊運轉

有時PID控制器會規劃為無衝擊（bumpless）的特性，在參數變化時重新計算適當的積分累計值，使輸出不會因參數變化有不連續的改變^[15]。

串級PID控制器

二個PID控制器可以組合在一起，得到較佳的效果，這稱為串級PID控制。以兩個PID控制器組成的串級PID控制為例，其中一個PID控制器在外迴路，控制像液面高度或是速度等主要的物理量，另一個PID控制器是內迴路，以外迴路PID控制器的輸出做為其目標值，一般是控制較快速變化的參數，例如流量或加速度等。若利用串級PID控制，可以增加控制器的工作頻率，並降低其時間常數。

例如一個溫控的循環水浴設備有二個串級的PID控制器，分別有各自的熱電偶溫度感測器。外迴路的控制器控制水溫，其感測器距加熱器很遠，直接量測整體水溫，其誤差量是理想水溫及整體水溫的差值。外迴路PID控制器的輸出即為內迴路控制器的目標值，內迴路控制器控制加熱器，其感測器是在加熱器上，其誤差量是加熱器的理想溫度及量測到溫度的差值，其輸出會使加熱器維持在設定值附近。

內外迴路控制器的參數可能會差很多，外迴路的PID控制器有較長的時間常數，對應所有的水加熱或是冷卻需要的時間。內迴路的PID控制器反應會比較快。每個控制器可以調整到符合其真正控制的系統，例如水槽中所有的水，或是加熱器本身。

其他PID的形式及其表示法

理想的PID及標準形PID

工業上常看到PID控制器，而許多工業相關資料中看到的都是「標準形」的PID，其中比例極益 K_p 也作用在 I_{out} 及 D_{out} 兩項，和上述「理論」段落看到的形式不同。「標準形」的PID為：

$$MV(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{d}{dt} e(t) \right)$$

其中

T_i 為積分時間

T_d 為微分時間

在標準形中，每一個參數有其明確的物理意義，輸出是根據現在誤差、過去誤差及未來誤差而決定，加上微分項可以預測若控制系統不改變的話， T_d 時間後的誤差，而積分項是用過去所有誤差的和來調整輸出，希望在 T_i 時間後可以完全消除誤差，而輸出的值會再乘以單一的增益 K_p 。

在理想的平行式PID中，其方程式如下：

$$MV(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t)$$

其中的增益和標準形PID係數的關係是： $K_i = \frac{K_p}{T_i}$ 及 $K_d = K_p T_d$ 。平行式PID中的參數都視為單純的增益，最泛用，靈活性也最高，但較沒有物理意義，因此只用在PID的理論處理中，標準形PID雖在數學上比較複雜，在工業中較常使用。

倒數增益

許多情形下，PID控制器處理的變數是無因次的量，是某個最大值的比例，介於0到100%之間，而轉換為實際物理量（如泵浦速率或是水加熱的功率）是在PID控制器外，而這些控制變數是有因次的物理量（例如溫度）。此時 K_p 增益多半不會表示為「每變化一度的輸出」，而會以溫度的形式 $1/K_p$ 表示，代表「100%輸出下的溫度（變化）」，代表輸出由0變到1（0%變為100%）下的溫度變化。

只針對過程變數進行微分控制

在大部份的商業控制系統中，是用過程變數取代誤差作為微分項的輸入，其原因是當目標值有不連續變化時，微分控制會產生很大的突波，若目標值不變，改變過程變數的效果和改變誤差相同，因此有些PID控制器會用過程變數作為微分項的輸入，不會影響控制器控制過程變數，抗雜訊的能力。

$$MV(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau - T_d \frac{d}{dt} PV(t) \right)$$

只針對過程變數進行比例控制

大部份的商業控制系統也提供選擇，讓過程變數也作為比例控制的輸入，因此誤差是作為積分控制的輸入，這也不會不會影響控制器控制過程變數，抗雜訊的能力。

上述的修改可以避免目標值有不連續變化時，輸出值有對應不連續的變化，若目標值有步階變化，這項調整就相當重要。

$$MV(t) = K_p \left(-PV(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau - T_d \frac{d}{dt} PV(t) \right)$$

PID控制器的拉氏轉換

有關會將PID控制器進行拉氏轉換：

$$G(s) = K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s}$$

PID控制器的拉氏轉換也代表著控制器的傳遞函數，因此可以確認整體系統的傳遞函數。

PID的極零點對消

PID控制器可以寫成以下的形式

$$G(s) = K_d \frac{s^2 + \frac{K_p}{K_d} s + \frac{K_i}{K_d}}{s}$$

若受控設備的傳遞函數如下：

$$H(s) = \frac{1}{s^2 + 2\zeta\omega_0 s + \omega_0^2}$$

又令

$$\frac{K_i}{K_d} = \omega_0^2$$

$$\frac{K_p}{K_d} = 2\zeta\omega_0$$

則

$$G(s)H(s) = \frac{K_d}{s}$$

因此若受控設備有不穩定的極點，看似可以用此方式消除，不過實際上有些差異，由干擾到輸出的閉迴路傳遞函數中仍有不穩定的極點，因此仍可能會發散。

串級型或交互型

另一種PID控制器的表示法為串級型（series）或稱為交互型（interacting）

$$G(s) = K_c \frac{(\tau_i s + 1)}{\tau_i s} (\tau_d s + 1)$$

其中參數和標準型的參數以下的關係

$$K_p = K_c \cdot \alpha, T_i = \tau_i \cdot \alpha$$

$$T_d = \frac{\tau_d}{\alpha}$$

而

$$\alpha = 1 + \frac{\tau_d}{\tau_i}.$$

上述作法可表示為二個串級的PD控制器及PI控制器，在早期類比電路的時代較容易實現，雖然控制器已經數位化，不過仍有些維持此形式。

離散化的控制器

若要在微處理機（MCU）或是FPGA中實現PID控制或是分析其性能，就需要將控制器離散化^[16]。一階微分可以用後向有限差分表示，積分項也離散化，若取樣時間為 Δt ，積分項可以用下式近似

$$\int_0^{t_k} e(\tau) d\tau = \sum_{i=1}^k e(t_i) \Delta t$$

微分項可近似為

$$\frac{de(t_k)}{dt} = \frac{e(t_k) - e(t_{k-1})}{\Delta t}$$

因此PID控制器的離散化可以將 $\mathbf{u}(t)$ 微分，再用一階導數及二階導數的定義求得 $\mathbf{u}(t_k)$ ，可以得

$$\mathbf{u}(t_k) = \mathbf{u}(t_{k-1}) + K_p \left[\left(1 + \frac{\Delta t}{T_i} + \frac{T_d}{\Delta t} \right) e(t_k) + \left(-1 - \frac{2T_d}{\Delta t} \right) e(t_{k-1}) + \frac{T_d}{\Delta t} e(t_{k-2}) \right]$$

其中 $T_i = K_p/K_i, T_d = K_d/K_p$

偽代碼

以下是一段實現PID演算法的偽代碼：^[17]

```
previous_error = 0
integral = 0
start:
    error = setpoint - measured_value
    integral = integral + error*dt
    derivative = (error - previous_error)/dt
    output = Kp*error + Ki*integral + Kd*derivative
    previous_error = error
    wait(dt)
    goto start
```

此例中有兩個變數在迴圈前需初始化為0，然後開始迴圈。目前的誤差（**error**）是用目前目標值（**setpoint**）減去系統反饋值（**measured_value**）而得，然後再進行積分和微分運算，比例項、積分項及微分項乘以各自參數後得到輸出（**output**）。在實際系統中，這會透過數位類比轉換器轉換為類比訊號，作為受控系統的控制量。目前的誤差量及積分會儲存，以便下次計算微分及積分時使用，程式會等待**dt**秒後開始，迴圈繼續進行，透過類比數位轉換器讀取新的系統反饋值及目標值，再計算誤差量及輸出^[17]。

參見

- 控制理論
- 反饋
- 不穩定
- 振盪

注釋

- a. ^ 唯一的例外是目標值恰好是比例增益等於0時的受控輸出。

參考文獻

- ^ Li, Y., Ang, K.H., and Chong, G.C.Y. (2006) **Patents, software and hardware for PID control: an overview and analysis of the current art**. *IEEE Control Systems Magazine*, 26 (1). pp. 42-54. ISSN 0272-1708 (doi:10.1109/MCS.2006.1580153) (<http://eprints.gla.ac.uk/3816/1/IEEE2pdf.pdf>),
 - ^ **2.0 2.1 2.2** Bennett, Stuart. A history of control engineering, 1930-1955. IET. 1993: p. 48. ISBN 978-0-86341-299-8.
 - ^ Bennett, Stuart. Nicholas Minorsky and the automatic steering of ships (PDF). *IEEE Control Systems Magazine*. November 1984, **4** (4): 10–15. doi:10.1109/MCS.1984.1104827. ISSN 0272-1708.
 - ^ A Brief Building Automation History. [2011-04-04].
 - ^ (Bennett 1993 , p. 67 (http://books.google.com/books?id=VD_b81J3yFoC&pg=PA67))
 - ^ Bennett, Stuart. A brief history of automatic control (PDF). *IEEE Control Systems Magazine* (IEEE). 1996, **16** (3): 17–25.
 - ^ Bennett, Stuart. A history of control engineering, 1800-1930. IET. June 1986: 142–148. ISBN 978-0-86341-047-5.
 - ^ Introduction: PID Controller Design. University of Michigan.
 - ^ Tim Wescott. PID without a PhD (PDF). *EE Times-India*. October 2000.
 - ^ **10.0 10.1 10.2** Ang, K.H., Chong, G.C.Y., and Li, Y. (2005). PID control system analysis, design, and technology, *IEEE Trans Control Systems Tech*, 13(4), pp.559-576. <http://eprints.gla.ac.uk/3817/1/IEEE3.pdf>
 - ^ Li, Y., et al. (2004) CAutoCSD - Evolutionary search and optimisation enabled computer automated control system design, *Int J Automation and Computing*, vol. 1, No. 1, pp. 76-88. ISSN 1751-8520. http://userweb.eng.gla.ac.uk/yun.li/ga_demo/
 - ^ Y Li, KH Ang, GCY Chong, Patents, software, and hardware for PID control: An overview and analysis of the current art, *Control Systems, IEEE*, 26 (1), 42-54. <http://eprints.gla.ac.uk/3816/1/IEEE2pdf.pdf>
 - ^ Li, Y. and Ang, K.H. and Chong, G.C.Y. (2006) PID control system analysis and design - Problems, remedies, and future directions. *IEEE Control Systems Magazine*, 26 (1). pp. 32-41. ISSN 0272-1708 (http://eprints.gla.ac.uk/3815/1/IEEE_CS_PID_01580152.pdf)
 - ^ Cooper, Douglas. Integral (Reset) Windup, Jacketing Logic and the Velocity PI Form. [2014-02-18].
 - ^ Cooper, Douglas. PI Control of the Heat Exchanger. *Practical Process Control by Control Guru*. [2014-02-27].
 - ^ Discrete PI and PID Controller Design and Analysis for Digital Implementation. *Scribd.com*. [2011-04-04].
 - ^ **17.0 17.1** PID process control, a "Cruise Control" example. *CodeProject*. 2009 [4 November 2012].
- Minorsky, Nicolas. Directional stability of automatically steered bodies. *J. Amer. Soc. Naval Eng.* 1922, **34** (2): 280–309. doi:10.1111/j.1559-3584.1922.tb04958.x.

外部連結

- 改善PID微分和積分的方法及其它控制系統的計算機自動設計CAutoD (http://userweb.eng.gla.ac.uk/yun.li/ga_demo/)
- 學習PID和其他系統調試是如何工作的 (http://www.bin95.com/PID_Process_Control_Saint-Louis.htm)
- PID教程 (<http://www.expertune.com/r2.asp?f=Wikipedia&l=tutor.html>)
- PID控制器實驗室，PID調試的Java applets (<http://www.pidlab.com>)
- 一系列的PID調試的Java Applets (<http://ae.tut.fi/~juke/java/pidtuning/>)
- PID調試的問答 (<http://www.tcnj.edu/~rgraham/PID-tuning.html>)
- PID控制系統算法的資訊和教程 (<http://www.jashaw.com/pid>)
- 用Excel模擬基本的PID (<http://www.htservices.com/Applications/Process/PID2.htm>)
- 如果用電子部件製作一個PID控制器 (<http://asl.epfl.ch/research/projects/VtolIndoorFlying/rapports/rapportSemStauffer.pdf>)查看22頁
- 關於PID控制器的文章，教材 (<http://www.expertune.com/r2.asp?f=Wikipedia&l=articles.html>)
- 一個控制系統的一部分 (<http://www.industrial-electricity.com/open-and-closed-loop-feedback-systems-2-Parts-Typical-Control-System.html>)
- PID定速控制應用 (<http://4rdp.blogspot.com/2008/05/pid-speed-control.html>)
- PID馬達定速與定角控制公式比較 (<http://4rdp.blogspot.com/2008/12/apply-pid-to-control-lego-nxts-speed.html>)
- Ang, K.H., Chong, G.C.Y., and Li, Y. (2005) , **PID control system analysis, design, and technology**. *IEEE Transactions on Control Systems Technology*, 13 (4). pp. 559-576. ISSN 1063-6536 (<http://eprints.gla.ac.uk/3817/1/IEEE3.pdf>)
- Understanding Servo Tune(其中包括PID調整方法範例) (<http://www.ni.com/product-documentation/2923/en/>)
- LabView360技術文章 PID (<http://labview360.com/article/info.asp?TID=10149&FID=165>)

取自 "https://zh.wikipedia.org/w/index.php?title=PID控制器&oldid=41274809"

-
- 本頁面最後修訂於2016年8月27日 (週六) 07:31。
 - 本站的全部文字在創用CC 姓名標示-相同方式分享 3.0 協議之條款下提供，附加條款亦可能應用（請參閱使用條款）。
Wikipedia®和維基百科標誌是維基媒體基金會的註冊商標；維基™是維基媒體基金會的商標。
維基媒體基金會是在美國佛羅里達州登記的501(c)(3)免稅、非營利、慈善機構。