PID控制器基礎原理及應用

目录

[一、前言 2](#_Toc527567436)

[二、什麼是控制器？為什麼要用控制器？ 2](#_Toc527567437)

[三、什麼是PID? 2](#_Toc527567438)

[四、應用及代碼實現 4](#_Toc527567439)

[五、參考資料 4](#_Toc527567440)

一、前言

此文會從零解釋PID，包括一些專業術語的解釋。主要希望能讓普通的學生也能明白PID的基礎原理以及能多了解一些控制理論的基礎知識。我認為有著普通的高中數學基礎就應該能看明白，我相信只要努力你便能完完全全讀懂文章，畢竟不思考便能消化知識我是不相信的。

PID這個字眼對於經常上網查自動控制資料的學生絕對是不陌生的，但總感覺像是一個謎一樣完全不理解有什麼用，就算知道有什麼用，也不知道怎麼用。首先，網上可能有一些叫PID為算法，有的叫控制器，我個人比較偏向於叫控制器，畢竟我們即將要講述的PID是應用在控制方面上的。

作者我本人實際上理解的程度也不是很深，若有說的不對的請諒解，若想技術交流或者有問題可電郵jameshoi@foxmail.com。

二、什麼是控制器？為什麼要用控制器？

控制器顧名思義就是控制東西的器具。有參加過機器人比賽的同學就會知道，比賽的任務總是想讓你在手動或者自動的情況下控制機器人。舉個簡單的例子，你有一個四驅小車，你需要控制他向右行走特定的距離，也就是到達目標位置，實際上你可以通過編碼器來知道小車行走了多長，然後寫一句if讓小車到達指定位置後停下，但當摩擦力不夠大的情況下，你的小車便會超過你的指定位置，這就是為什麼我們需要一個控制器來完成操作，控制器可以以某些方式的關係讓小車輸出的速度一直在不斷改變，從而能讓小車在目標位置停下。PID控制器是一種不需要運動模型的控制器，其原理簡單，易於實現，所以它應用廣泛。

什麼是運動模型？通常上位機算法處理完之後會給予下位機一個以m/s的速度量，但實際上例如普通的直流馬達，是由下位機給予其占空比來作為輸出，所以運動模型就是用來轉換算法所希望的速度和占空比的。

三、什麼是PID?

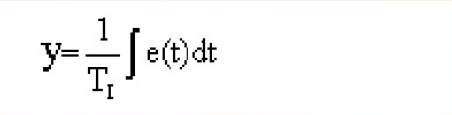
PID控制器的P代表比例，I代表積分，D代表微分。不明白這三個是什麼意思完全不要緊，明白其中代碼如何實現才是關鍵。雖說本文所說的是PID控制器，但接下來也會講述PI，PD，P控制器。控制系統一般分為開環控制和閉環控制，沒有回饋的控制系統就叫開環控制系統，有回饋的控制系統就叫閉環控制系統。（想了解更多自行百度或看[這個](https://zhidao.baidu.com/question/1539674505842188667.html)）PID控制器是閉環控制，根據反饋回來的數據進行調整。但是電腦自己還不會判斷這個值是好或者壞，所以你需要告訴它你的標準，這個標準稱為目標值。那麼評定好壞便可以用目標值減去目前傳感器所接受到的值，當這個值越大，說明離目標越遠；當這個值變成負數，說明已經超過目標值了。這個值叫做調節器的偏差

無論是什麼濾波或者控制器都需要調參數（小人不才，至少了解到的都是這樣，除了圖像識別的直方图均衡化），PID控制器需要調的三個參數為KP（比例系數），KI（積分系數），和KD（微分系數）。

比例：

（1）

積分：

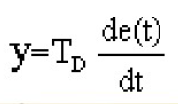


（2）

（KI=1/TI）

TI叫做積分時間常數，它表示積分速度的大小，TI越大，積分速度越慢，積分作用越弱

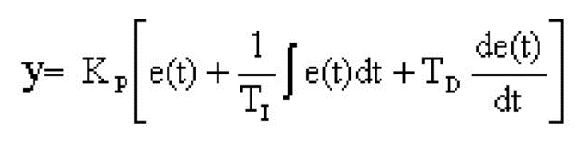
微分：



（3）

（KD=TD）

理想的PID方程：



（4）

我們到現在為止所說的都是理想狀態下，由於單片機頻率不能為無限大，所以真實狀態下調整的速度并沒有理想狀態下那麼快，而且不能實現理想狀態的積分和微分。

所以實際上編程積分的實現便是：



（這句話會加在一個循環裡面）

pidcontroller是我寫的一個結構體，它的Error便是我們之前所說的偏差，Error\_Integral（偏差的”積分”）便是將每一次的偏差相加。

編程微分的實現：



（這句話會加在一個循環裡面）

pidcontroller.Error-pidcontroller.LastError便是偏差的”微分”，因為是一個循環，當記錄了上一次的error值之後，在下一個循環你便能使用上一次的error值（LastError）

雖說PID的I和D有點難理解，但假如你只是需要讓一部小車到達指定位置，只用P已經能實現了。PID實際上可以分為P控制器，PI控制器，PD控制器，PID控制器。用哪一種控制器還是需要看應用的場景。

積分項一般在系統有持續的外力時才會添加，而且當到達目標后還需要一直持續，例如飛行器。飛行器在上升指定高度的後，還需要持續定在指定位置，所以可以使用PI控制器。再例如小車，雖然小車行走時有摩擦力作用，但當到達目標時不需要持續。所以用P控制即可。

微分項一般在系統有震蕩時增加，例如飛行器需要平穩在某個高度，由於風的關係，可能會導致飛行器晃動，添加微分項後能減少震動，不過若微分項添加的較大會導致飛行器晃動的更為厲害，所以測試時注意安全。

所有精華都在我的代碼中，我真的沒有時間打注釋，實在看不明白找我。

（實際上還有好多沒寫，打文檔真的很浪費時間……）（明明我上次和別人講了大概十五分鐘就講完了……）（有問題電郵我）

四、應用及代碼實現

（百度雲盤）<https://pan.baidu.com/s/1u-LTp2QTx0wt-OicUAc7Qg>

五、參考資料

PID算法解析<https://wenku.baidu.com/view/652c5fbcfd0a79563c1e72f2.html>

PID算法解析（百度雲盤）<https://pan.baidu.com/s/1ryW4ZTqUhQKomz-F1ofoPw>