



datacompboy 6 мая 2012 в 08:40

Реклама

# Корректная реализация разностной схемы

Алгоритмы, Математика

Tutorial

[Все потоки](#) [Разработка](#) [Администрирование](#) [Дизайн](#) [Менеджмент](#) [Маркетинг](#) [Научпоп](#)

Войти

Регистрация

применяемый наиболее широко. Для своей работы требует настройки 3х коэффициентов, чтобы подобрать процесс регулирования согласно требованиям. Обладая простым физическим смыслом, запись, применяется широко и часто в регуляторах температуры, регуляторах расхода и т.д. поддерживать некий параметр на заданном уровне, с возможными переходами между уровнями. Разумеется, существуют более сложные регуляторы, позволяющие более точно и быстро выводить на заданные параметры, а так же учитывающие нелинейность или гистерезис процесса, но они обладают большей вычислительной сложностью и сложнее в настройке.

Несмотря на свою простоту как физического смысла, так и математической записи:

$$u(t) = P + I + D = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de}{dt}$$

при программной реализации ПИД регулятора очень часто допускают ошибки, которые вносят в работу приборов автоматики.

Причем проверить качество реализации ПИД регулятора крайне легко.

Рассмотрим простейший пример: терморегулятор. Для проверки его качества лучше всего использовать малоинерционный, маломощный объект. Классика жанра: обычная лампочка на 100Вт с терморепаром (ХА). И первое, на чем следует проверять ПИД регулятор — деградация ПИД до просто П-регулятора. То есть коэффициенты интегральной и дифференциальной ставим в ноль, а пропорциональную ставим в максимум.

Включаем регулятор, проверяем: текущая температура 22 градуса, уставка 16 градусов. Лампочка не горит. Начинаем увеличивать уставку: 16.1, 16.3, 16.7, 18... 19... лампочка загорелась. Как?! Откуда?! Останавливаемся — выключилась. Итак, мы встретились с **первой классической ошибкой** реализации ПИД регулятора.

Небольшое математическое отступление: вспомним еще раз интегральную запись указанную выше. Мы реализуем её программно, а значит — дискретно. То есть с завидной регулярностью производим измерение входной величины, сравниваем её с уставкой, вычисляем воздействие, выдаём, повторить. А значит, надо из интегральной формы перейти в конечно-разностную схему. При переходе обычно используется переход «в лоб»:

$$U(n) = K_p E(n) + K_i^{discr} \sum_{k=0}^n E(k) + K_d^{discr} (E(n) - E(n-1))$$

где  $E(n) = X(n) - X_0(n)$  — то есть величина рассогласования между текущим и заданным значением регулируемого параметра.

Использование прямой формулы потребует во-1х считать и хранить интеграл рассогласований за большой период, во-2х требует работы с плавающей точкой высокой точности (так как интегральный коэффициент  $K_i$  всегда  $< 1$ ), либо операции деления (представляя коэффициент в форме  $1/K_i$ ) большой разрядности. Всё это требует вычислительных ресурсов, коих в embedded как правило сильно ограничено... Поэтому, вместо реализации прямой схемы, реализуют рекуррентную формулу:

$$U(n) = U(n-1) + K_p (E(n) - E(n-1)) + K_i^{discr} E(n) + K_d^{discr} (E(n) - 2E(n-1) + E(n-2))$$

использование рекуррентной формулы позволяет сократить объём вычислений и разрядность промежуточных значений.

Итак, вернёмся к нашему регулятору. Итак, есть регулируемый объект: лампочка. Для управления подаваемой на неё мощностью, используют простой подход: сеть питания (220В 50Гц) подаётся через симистор на нагрузку. Симистор в момент перехода полуволны через ноль выключается, и остаётся выключен до тех пор, пока не будет подан сигнал на управляющий электрод. Таким образом, чем раньше после начала полуволны мы подадим управляющий сигнал, тем больше энергии от этой полуволны достигнет управляемого объекта. Правильно рассчитав время для линейности площади части полуволны с момента времени  $X$  до конца полуволны, мы получаем возможность выдавать мощность от 0 до 100% с точностью, с которой рассчитали таблицу линеаризации.

Итак, мы можем выдавать мощность от 0 до 100%. В реальных объектах, часто нельзя выдавать 100% мощности — например, это чревато перегоранием нагревательного элемента. Поэтому, все приборы имеют настройку минимальной и максимальной выдаваемой мощности на объект.

Итак, после вычисления  $U(n)$  по вышестоящей формуле, добавляется еще ограничение результата:

```
if Un < Umin then Un := Umin; if Un>Umax then Un := Umax;
```

После чего, вычисленное  $U_n$  и есть требуемая выходная мощность на текущий момент. Та-дам! Именно вот эта реализация и создаёт ошибку, о которой написано выше.

Причина банальна: в момент перехода от дискретной к конечно-разностной схеме, мы «выносим за скобки» операцию вычисления интеграла, и на каждом шаге мы прибавляем производную к накопленной сумме  $U(n-1)$ . Наложив же на неё ограничение, мы весь вычисленный интеграл фактически обнуляем. (Ну не сколько обнуляем, сколько приводим к диапазону 0-100, что в данном случае несущественно). Таким образом, мы дифференцируем ПИД регулятор, и остаётся дифференциально-ускорительный. Что в реальности выглядит как просто дифференциальный регулятор — мощность при этом подаётся пропорционально изменению уставки или регулируемой величины, а не пропорционально разности между уставкой и регулируемой величиной.

**Вывод №1:** вычисление  $U(n)$  нельзя ограничивать. Для ограничения мощности подаваемой на выходное устройство следует заводить отдельную переменную.

Теперь, когда мы завели  $U_{rn}$ , для ограниченной мощности, перезагружаем, продолжаем тестировать.

Включаем регулятор, проверяем: текущая температура 22 градуса, уставка 16 градусов. Лампочка не горит.

Начинаем прибавлять уставку: 16.1, 16.4, 17, 18, 20, 22, 24 (опа! засветилось! ура!), 28, 30, 40, 60... Красота! Работает!

Наблюдаем за процессом — вышло примерно на 60, болтается чуток туда-сюда, но держит. Вроде, всё красиво. Выдыхаем, проверяем управление с ПК: задаём 600 градусов. И... Лампочка выключается. Как так? Уставка 600, текущая 60, а лампочка не горит?

Пока ждём и медленно осознаём, что мы явно напоролись на какой-то «**Классический Косяк №2**»<sup>™</sup> лампочка медленно разгорается, выходит на 100% мощности, и так и остаётся — 600-то градусов она выдать ну никак не может.

Возвращаемся снова к нашей разностной схеме.  $U(n) = U(n-1) + K_p * (dE + \dots)$ . К текущему расчетному значению воздействия прибавляется разность невязки помноженная на коэффициент пропорциональности. У нас была уставка 60, температура 60, то есть невязка нулевая. Выходная мощность так же была нулевая. И тут разом, скачком, уставку увеличили до 600 градусов. невязка резко стала 540 градусов, помножили еще и на коэффициент пропорциональности... и вылетели за

разрядность хранения  $U(n)$ . Не смейтесь, использование математики с фиксированной точки, вместо плавающей точки. При разнице в 540 градусов и работе через  $1/16$ , при коэффициенте пропорциональности 20, получаем...  $540 \cdot 20 \cdot 16 = 172800$ , а если у нас 16тиразрядный  $U(n)$ , да еще и знаковый, то фактически, в результате вычисления мы получили  $A300h = -8960$ . Опачки. Вместо большого плюса — ощутимый такой минус.

**Вывод №2:** вычисления должны проводиться с корректной поддержкой переполнения. Переполнилось? Ограничь предельным числом, уж никак не заворачивать.

Итак, нарастили разрядность  $U(n)$ , перетранслировали, зашили, запускаем. Лампочка еще не совсем остыла, там 80 градусов, уставка всё те же 600. Лампочка зажигается... и тухнет. Зажигается и тухнет. Как так? Уставка 600, лампочка 80 — и поддерживает вполне себе свои 80! Как так-то?! Явно у нас вылез **Жучок №3**.

И снова лирически-математическое отступление. Итак, есть наша разностная схема:  $U(n) = G(U(n-1), dE(n))$ . Еще раз: новое значение воздействия есть сумма прошлого воздействия и некоего воздействия, зависящего от разности невязки в текущий момент и предыдущий. А что такое предыдущий момент? А какой момент предыдущий у предыдущего? Ну-ка, вспомнили школу. Доказательство по индукции. Если можно построить доказательство для  $K+1$ , считая что доказательство для  $K$  верно, **И** доказать отдельно что верно для  $K=0$ , тогда доказательство истинно. Итак, а как мы считаем  $U(0)$ ?

Часто встречающееся решение: всё обнуляем, уставку считываем из флешпамяти уставки, ждём 1 цикл опроса, и считываем  $X(0)$ . Вот, ноль готов, теперь работаем. И... И не правильно. Почему? Потому что рекуррентная формула отталкивается от изменений в невязке. А проинициализировав нулём и загрузив текущие значения мы потеряли стартовые условия. Всё — вместо поддержания абсолютного значения температуры на уровне равном абсолютной уставке регулятор начинает держать температуру равную стартовой плюс разнице уставки. То есть было 80 градусов и уставка 200, включили прибор — он держит 80. Сменили уставку на 240 — он начал держать 120.

Правильная инициализация разностной схемы: обнулить вообще всё. То есть

$$X(0) = 0, X0(0) = 0. U(0) = 0. E(0) = X(0) - X0(0) = 0.$$

И на первом же цикле вычислений у нас как бы скачком появляются уставка и текущее значение:

$$X(1) = 80. X0(1) = 200. U(1) = U(0) + K_p * (E(1) - E(0)) = U(0) + K_p * (X(1) - X0(1) - E(0)) = 0 + 20 * (200 - 80 - 0) = 2400$$

Вот теперь схема работает правильно.

**Вывод №3:** корректно инициализируй стартовые условия.

Правильно ли? Ну-ка, ну-ка... Еще раз... Ставим уставку 20. Ждём охлаждения... Выключаем. Включаем. Итак, красота: текущая 20, уставка 20. Ставим скачком 600. Поехало греться. 100, 120... ставим уставку 20. Отключилось, пошло охлаждаться. Ждём чутка (120... 110... 100... 90... 80...) и ставим уставку 100. Поехало греться... 105 градусов, отключилось. Стоп. А почему оно держит 105? У нас же сейчас работает только пропорциональная составляющая. При правильной реализации из физического смысла процесса колебательный процесс не может держать уставку выше чем задано. Строго ниже. А держит на 5 градусов больше чем попросили. Это наблюдается **Прикол №4**.

Итак, вспоминаем что у нас было выше: Вывод№2:  $U(n)$  нельзя ограничивать. И Вывод№3: при переполнении ограничить всё-таки придётся. Да-да. Иначе «рабочая точка» смещается на ограниченный момент. Что же делать? Увеличить разрядность? Хорошо, если хватает вычислительной мощности. А надо ли? Собственно, что плохого, что у нас  $U(n) = 9999.99$ , а не  $29999.99$ ? В общем-то только то, что мы потеряли 20000. Но сейчас-то для работы нам так и так надо вваливать просто 100% мощности, правильно? Правильно. Значит, проблемы с ограничением в полку нет, до тех пор, пока мы не отходим от предела. Таким образом, в случае переполнения надо ставить флаг, и по достижении, например, половины диапазона (то есть как  $U(n)$  после 9999.9 опустилось ниже 5000.00), заново реинициализировать схему. То есть отбрасывать историю, сказать что  $n=0$  и см. выше Вывод №3. Пытливый ум уже сообразил, что в случае полной схемы, когда все три компоненты не равны нулю, обнуляя в процессе итеративный процесс, мы в том числе обнуляем накопленный интеграл интегральной составляющей. Однако, в связи с тем, что обнуляем мы значительно заранее, он успеет подкопиться за время довырабатывания остатка. Да и не совсем корректно копить интеграл на «больших» перегонах, так как цель интегральной составляющей — «выбрать» невязку, которую не может отработать пропорциональная составляющая отдельно.

**Вывод №4:** если по какой-то причине  $U(n)$  было ограничено, схему следует переинициализировать как только создалось впечатление, что схема вернулась в нормальное состояние.

*В следующем выпуске:* а так ли надо реализовывать разностную схему? Подробная реализация прямой дискретной схемы, обладающей простыми и понятными настраиваемыми коэффициентами, с прямым физическим смыслом, которая без проблем вычисляет управляющее воздействие с частотой 25Гц на процессоре ADuC847 (быстрый 8-битный контроллер, с ядром 8051), оставляя еще море процессорного времени для других процессов.

(Картинки с изображением формул взяты из статьи [ПИД-Регулятор](#) в Wikipedia)

**Теги:** PID, ПИД, регулятор, реализация, математика, кодирование алгоритмов

**Хабы:** Алгоритмы, Математика

↑ +43 ↓ 240 72k 27 Поделиться



163,5

Карма

1,2

Рейтинг

**Anton Fedorov @datacompboy**

Программист / сисадмин (BigQuery SRE, L5)

#### ПОХОЖИЕ ПУБЛИКАЦИИ

19 февраля 2021 в 12:21

**Взлом по любви. Как математик взломал алгоритм сайта знакомств и нашел идеальную девушку**

↑ 83 50,8k 120 180

19 августа 2020 в 19:39

**Настройка ПИД-регуляторов: так ли страшен черт, как его малюют? Часть 5. Closed-Loop PID Autotuner**

↑ 5 1,8k 25 1

12 августа 2020 в 11:17

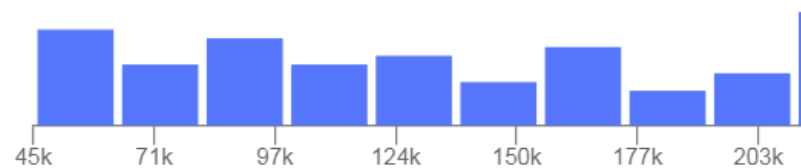
**Настройка ПИД-регуляторов: так ли страшен черт, как его малюют? Часть 4. Frequency Response Based PID Tuner**

↑ 2 1,8k 28 4

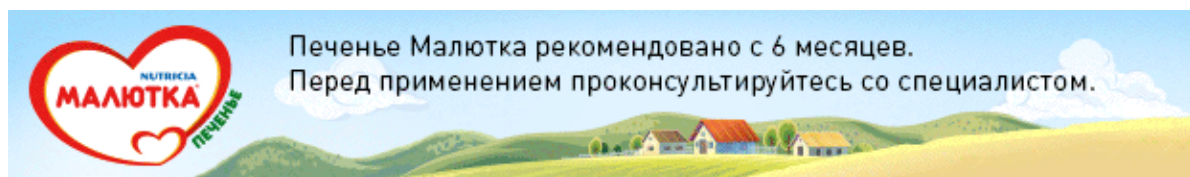
## СРЕДНЯЯ ЗАРПЛАТА В IT

**120 000** ₽/мес.

Средняя зарплата по всем IT-специализациям на основании 3 808 анкет, за 1-ое пол. 2021 года

[Узнать свою зарплату](#)

Реклама



## Комментарии 27

**harlov91**

6 мая 2012 в 09:06

**+8**

Отличное легкое чтение в воскресенье, под кофе.

**Den\_Fisher**

6 мая 2012 в 11:04

**0**

О, мне как раз скоро ТАУ сдавать, спасибо.

**datacompboy**

6 мая 2012 в 11:10

**+1**

Для сдачи не пойдёт, это именно относительно практики, а не теории.



**boolivar** 6 мая 2012 в 11:12 #

↑ +1 ↓

С'И'мистор, простите



**datacompboy** 6 мая 2012 в 11:19 #

↑ 0 ↓

уже исправлено, да. очепятка. my bad.



**DENICKA** 6 мая 2012 в 14:37 #

↑ +2 ↓

Спасибо! Для наглядности немного не хватает графиков.



**datacompboy** 6 мая 2012 в 15:27 #

↑ 0 ↓

Графики, они нужны для настройки. Для отлова вот этих классических ошибок реализации достаточно банального наблюдения глазами за лампочкой и текущим показанием прибора. Какие там графики, вы о чем :)



**DENICKA** 6 мая 2012 в 16:50 #

↑ +2 ↓

Помоему, обычные графики зависимости температуры от времени нагляднее чем словесное описание поведения лампочки.



**DENICKA** 6 мая 2012 в 16:56 #

↑ 0 ↓

Вот, как здесь например предельно доступно рассмотрены основы PID:  
[www.eetimes.com/ContentEETimes/Documents/Embedded.com/2000/f-wescot.pdf](http://www.eetimes.com/ContentEETimes/Documents/Embedded.com/2000/f-wescot.pdf)



**datacompboy** 6 мая 2012 в 17:46

Вы читали статью? Статья совсем не ПИД регуляторе, и не о его свойствах. А о стандартных ошибках в его реализации, и способе проверки поведения регулятора «на глаз». Статья для тех, кто этот самый ПИД реализовывал, и не может понять чего он дурит.

Уже предполагается определённое понимание что это такое и как он работает.

**DENICKA** 7 мая 2012 в 19:29

Конечно. Как бы иначе я узнал про лампочку? :)

Я же не призываю вас описывать работу ПИД с нуля, а просто выразил своё мнение насчет наглядности.

Ваши ошибки из статьи были бы также прекрасно видны на графиках, как и настройки ПИДа.

**Tairesh** 6 мая 2012 в 14:45

> > *Простоту математической записи*  
и вот эта формула с интегралом и производной.

**datacompboy** 6 мая 2012 в 15:43

сравните с фильтром калмана:



**creat0r** 6 мая 2012 в 15:09



+4



Привет, Антон :)

Прикол №5, когда мы пытаемся классический хорошо допиленный ПИД-регулятор применить к ОЧЕНЬ инерционному объекту. Скажем, бочка 20 тонн с жижкой, которая экзотермична (сама себя подогревает, причём нелинейно). Для охлаждения жижку циркулируем через теплообменник с холодной водой. Датчик температуры стоит где-то в бочке. Регулируем проток жижки через теплообменник задвижкой с позиционером, циркуляционный насос фигачит прямым пуском... Время реакции системы на воздействие что-то вроде 7-10 минут.

Так вот, подобрать «отзывчивость» регулятора очень сложно. Слишком медленный регулятор очень часто не успевает за ростом температуры жижки. Вроде и задвижка уже 100%, а всё, процесс раскошегарился и сварил сам себя. Слишком быстрый регулярно переохлаждает жижку, а если нет, то колебания температуры всё равно слишком велики.

Решение — модифицировать регулятор, добавив «адаптацию», а именно — в момент прекращения роста (или падения) температуры замораживать выход регулятора, несмотря на то, что рассогласование всё ещё положительно (отрицательно). Размораживать, соответственно, при переходе через уставку (т.е. через нулевое рассогласование) или при возобновлении роста (падения). Шесть таких регуляторов реализованы на контроллере Simatic S7-300.

Результат — регулировка температуры 20 тонн жижки с точностью до 0.2 градусов. Классический ПИД при этом едва в 1.5 укладывался с регулярным подкручиванием коэффициентов.

Вывод №5 — не бояться отходить от канонов :)



**datacompboy** 6 мая 2012 в 15:38 # 📌 ↻ ↱

↑ +1 ↓

Да, ПИД регулятор не является ни идеальным регулятором, ни лучшим даже в тех случаях, где его применение допустимо. У него только один плюс — простота физического смысла — есть пропорциональная составляющая, когда мы просто греем на разность температур. Это съедает основную ошибку, но не может бороться с постоянными потерями — поэтому просто пропорциональный регулятор и держит всегда чуть ниже чем задали. Для избавления от вот этого «чуть ниже» вводится интегральная составляющая — которая аккумулирует ошибку, и даёт добавку к мощности, компенсирующую утечки. Собственно, в случае линейного самоподогрева — интегральная составляющая \_может\_ компенсировать ошибку. Если, конечно, реализация правильная — интеграл считается знаково :) — но этот тип ошибок я оставил за скобками, и так вон какая портянка. И диф. составляющая, повышающая «отзывчивость» на изменение уставки.

Хороший регулятор должен предсказывать поведение объекта, и рассчитывать управляющее воздействие именно с учетом предсказания.

Если же объект во-1х нелинейный, во-2х известный — ни интегральная ни дифференциальная составляющие не являются оптимальными, да еще и вредны в общем-то. Поэтому добавление к регулятору дополнительной «адаптации» под объект — и есть самое правильное поведение.

Еще частая адаптация — это использование нелинейных коэффициентов ПИД регулятора: когда на разные рабочие участки свой набор коэффициентов + свой набор коэффициентов для положительного и для отрицательного случая рассогласования (нагрев и охлаждение почти всегда различаются).

А вообще, регуляторы на базе фильтра калмана, или вообще синтезированные под конкретный объект — самые оптимальные регуляторы.



**datacompboy** 6 мая 2012 в 15:39 # 📌 ↻ ↱

↑ +1 ↓

Кстати, если кому интересно будет — неплохо [www.bookasutp.ru/Chapter5\\_3\\_4.aspx](http://www.bookasutp.ru/Chapter5_3_4.aspx) вот тут рассматриваются



**mayorovp** 6 мая 2012 в 22:35 # 📌 ↻ ↱

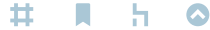
↑ +1 ↓

Вывод №5 — в сложных случаях надо, наоборот, вспоминать каноны (ТАУ) и не надеяться на ПИД-регуляторы.

Конкретно в данном случае в необычной повелении системы «виновата» не инерционность (которая, вообще-то, благо), а задержка: охлажденная жидкость из теплообменника должна успеть вернуться в бочку и перемешаться там, а пока этого не произошло, температура и не начнет уменьшаться. Иными словами, в системе имеется звено  $e^{-ts}$ , такое звено дает 2-3 «лишние» степени системе.



**creat0r** 6 мая 2012 в 22:45



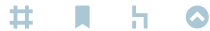
Ну как раз каноны ТАУ в данном случае не помогут. Система слишком сложна для построения каких-то моделей и передаточных функций. Какие уж там звенья  $n$ -го порядка...

Тут вплоть до того, что между выходом регулятора и позиционером небольшая функция написана, пытающаяся переменной длины импульсами «вверх» и «вниз» подогнать положение задвижки под выход регулятора на основе показаний не особо точного датчика положения этой самой задвижки. Помимо этого, функция ещё и реализована с учётом сохранения механического ресурса позиционера (минимизация движений).

Я уж молчу про то, что в бочку постоянно поступает, так сказать, исходное сырьё неизвестной температуры и консистенции :)



**HomoLuden** 7 мая 2012 в 13:23



> Ну как раз каноны ТАУ в данном случае не помогут.

Ну как раз каноны НЕЛИНЕЙНОЙ ТАУ здесь как нельзя кстати. Существуют отлаженные методы построения передаточной функции обратной связи с целью получения "Желаемой АЧХ". Все упирается в идентификацию объекта (в построение его АЧХ или передаточной функции). Если есть возможность экспериментально снять переходный процесс для импульсного или ступенчатого воздействия, то это уже позволит построить хоть какую-то приближенную модель. Дальше — дело техники.



**creat0r** 7 мая 2012 в 13:47



В том-то и дело, что, во-первых, реакция на «единичное воздействие» у системы всегда разная и зависит от доброго десятка непредсказуемых и неизмеряемых факторов. Во-вторых, никто не разрешит выдавать на этот объект всякие там воздействия. Стоимость жижки в бочке эдак примерно под 100К евро :)

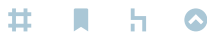
Поэтому в данном случае — только адаптивный регулятор, реализованный на основе существующего в библиотеке функций симатика ПИД-регулятора с минимальными изменениями (дабы не заниматься потом тестированием с ночёвкой на объекте :).

Да и вообще, к ожидаемому недовольству апологетов классической ТАУ, могу сказать, что при всём моём немаленьком опыте автоматизации разнообразных производств, мне ни разу не приходилось видеть, чтобы регулятор считали по всем правилам, с вычислением передаточных функций, с единичными ступенчатыми воздействиями и прочими изучаемыми в универе штуками. Всё строго на глаз, с потолка, научным тыком. И главное — работает :)

Один раз я только слышал о расчёте регулятора для какой-то сложной и быстрой колонны синтеза, где подбирать коэффициенты никак бы не получилось, система очень быстро уходила вразнос. Правда, расчётный регулятор в результате заработал едва ли не хуже настроенного по интуиции. Как решили проблему — не знаю. Скорее всего подобрали таки подходящий режим, утилизировав сотню-другую тонн исходного сырья :)



**HomoLuden** 7 мая 2012 в 17:34



> реакция на «единичное воздействие» у системы всегда разная... никто не разрешит выдавать на этот объект всякие там воздействия.

Тогда понятно.

А четкое следование пособиям ТАУ и не требуется. Достаточно синтезировать в общем виде передаточную функцию (это достаточно просто), а потом подвинчивать коэффициенты.

Я вот тоже как-то ПИД настраивал с нуля в мелком автомате по нанесению покрытия (никелевого или еще какого-то — не помню). Так у меня очень много времени ушло на «метод тыка». Поэтому, кажется, нужен компромисс. А синтезировать регулятор в виде модели пространства состояний для меня проще и понятнее, а следовательно результативнее и предсказуемее.



**Ariman** 11 мая 2012 в 13:14



Полезная статья, спасибо.

Интересно было бы узнать, насколько реализации, предлагаемые в DSP-библиотеках (от тех же ST Microelectronics для своих STM32Fxx) избавлены от указанных ошибок. Хочется верить что полностью — все же библиотека эта в своем роде стандарт, притом оптимизирована, и предпочтительнее использовать ее, нежели писать свой велосипед...

**datacompboy** 11 мая 2012 в 21:09

подозреваю, что хорошие DSP используют всё-таки прямую схему, а не разностную.

**Ariman** 11 мая 2012 в 21:16

```
uint16_t DoFullPID(uint16_t In, uint16_t Ref, uint16_t *Coeff)
{
    uint16_t Kp, Ki, Kd, Output, Error;

    Error = Ref - In;
    Kp = Coeff[0];
    Ki = Coeff[1];
    Kd = Coeff[2];

    IntTerm_C += Ki*Error;
    Output = Kp * Error;
    Output += IntTerm_C;
    Output += Kd * (Error - PrevError_C);

    PrevError_C = Error;

    return (Output);
}
```

Совсем в лоб.

**datacompboy** 12 мая 2012 в 08:03

Охoho... Можно влететь в шар с волосами из-за округлений при расчетах. Я во второй статье описываю приколюхи реализации прямой схемы.

**e\_asphyx** 24 января 2013 в 22:44

0



Спасибо, весьма интересно; сейчас как раз разбираюсь с PID управлением. Правда, мне по-прежнему не ясна базовая вещь: выходит PID — это абсолютное значение (тока через нагреватель, например) или же его изменение?

Отвлечёмся от ограничений, переполнения и прочих тонкостей, а также оставим для начала только пропорциональную составляющую. Пусть на выходе абсолютное значение. Итак, насколько я понимаю, в результате отклонение измеряемого параметра (например, температуры) должно пребывать вблизи нуля. Но тогда и ток будет вблизи нуля, а должен быть вблизи некоторого постоянного значения! В чём подвох?

**Int\_13h** 18 марта 2013 в 16:53

0



ПИД в первом приближении (благодаря пропорциональной части) — это усилитель ошибки. Когда ошибка 0, на выходе ПИД тоже 0.

Только полноправные пользователи могут оставлять комментарии. [Войдите](#), пожалуйста.

## САМОЕ ЧИТАЕМОЕ

Сутки

Неделя

Месяц

## Судьба предателя, угнавшего новейший МиГ-25 в Японию

↑ +93

👁 726k

🔖 121

💬 2450

Смотрим любое кино мгновенно

↑ +39    👁 21,8k    📖 209    💬 128

О фейковых криптовалютах (Ethereum, Tron, Ripple и пр)

↑ +219    👁 151k    📖 271    💬 448

Как освоить Vim?

↑ +42    👁 19,3k    📖 172    💬 172

Куда же плывут облака? Подбиваем итоги опроса и даём аналитику

Мегапост

Ваш аккаунт	Разделы	Информация	Услуги
Войти	Публикации	Устройство сайта	Реклама
Регистрация	Новости	Для авторов	Тарифы
	Хабы	Для компаний	Контент
	Компании	Документы	Семинары
	Пользователи	Соглашение	Мегaproекты
	Песочница	Конфиденциальность	Мерч

