

# Разработка роботов

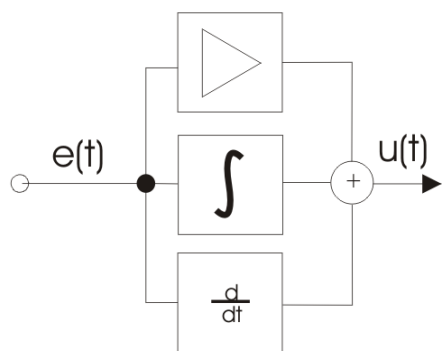
## Дискретный ПИД регулятор

Автор: Dmitry

Опубликовано: 07 Сентябрь 2011

Метки: Software

Рубрики: Публикации



При разработке автоматических систем, таких как система следования по заданной траектории мобильного робота или система поддержания постоянной температуры, возникает необходимость поддержания какого-либо параметра в соответствии с заданным значением. Само заданное значение может достаточно быстро меняться во времени. Теория автоматического управления для этой цели предлагает использовать регуляторы. Наиболее распространён ПИД (пропорциональный интегральный дифференциальный) регулятор. В данной статье рассмотрим на простом примере, что он собой представляет и как его реализовать на C++.

Сначала вкратце опишу, в чём заключается сама проблема регулирования. Допустим, у нас имеется мобильный робот, которому нужно повернуться на 30 градусов вокруг центра масс. Поворот осуществляется подачей на его привода требуемой угловой скорости вращения вокруг центра масс. Чтобы повернуться именно на 30 градусов требуемую угловую скорость вращения нужно подавать по определённому закону. Пока разность между текущим углом и требуемым велика, угловая скорость должна быть большой. При уменьшении разности скорость должна уменьшаться. И когда разность равна 0, скорость тоже должна стать равной 0. Но такой закон нельзя рассчитать заранее, потому что привода не моментально отрабатывают требуемую скорость, робот может кратковременно упереться в препятствие. И наконец, робот обладает моментом инерции, он может просто пролететь требуемое положение, и тогда скорость нужно будет подавать в противоположном направлении.

Самое простое решение в этом случае – это подавать требуемую скорость как разность требуемого углового положения и текущего, умноженную на некоторую неотрицательную константу. Такой алгоритм называется *П регулятором*. Но у этого алгоритма есть существенный недостаток: если требуемое углового положение вращается с некоторой угловой скоростью, то робот никогда не догонит эту точку. Настанет такой момент, когда угловая скорость, рассчитанная П регулятором, станет равной угловой скорости вращения требуемого положения. Чтобы решить эту проблему, можно к скорости от П регулятора прибавить скорость, рассчитанную как интеграл по времени ошибки по углу умноженный на некоторую неотрицательную константу.

[Go top](#)

требуемая скорость будет тем больше, чем дольше робот не может догнать требуемое угловое положение. Такой алгоритм называется *ПИ регулятором*. Данный алгоритм тоже можно улучшить. Когда ошибка по углу станет равной нулю, робот из-за инерции может пролететь дальше, если до этого его скорость была достаточно большой. Неплохо бы было притормаживать более быстро, когда ошибка мала. Для этого к требуемой скорости от ПИ регулятора нужно добавить производную ошибки по времени, умноженную на положительную константу. Уравнение *ПИД регулятора*:

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

В нашем случае:

**u(t)** – требуемая угловая скорость;

**e(t)** – угловая ошибка;

**P, I, D** – пропорциональная, интегральная и дифференциальная составляющие соответственно;

**Kp, Ki, Kd** – коэффициенты ПИД регулятора.

Если вы заметили, то коэффициенты содержат постоянные времени: интегральную и дифференциальную. Они и рассчитываются исходя из модели регулируемого устройства, используя математические методы, также существуют эмпирические методы измерения постоянных времени. Обязательно нужно обратить внимание на присутствие периода дискретизации, поэтому процесс PID регулирования должен вызываться в режиме реального времени через жестко заданные интервалы времени. И если вы изменили период обновления сигнала ошибки, то необходимо скорректировать коэффициенты ПИД.

В таком виде ПИД регулятор невозможно реализовать на компьютере. Дифференцирование нужно заменить разностью, а интегрирование суммой:

$$u(t) = P(t) + I(t) + D(t)$$

$$P(t) = C_p * e(t)$$

$$I(t) = I(t - T) + C_i * e(t)$$

$$D(t) = C_d * (e(t) - e(t-T))$$

**Cp, Ci, Cd** – коэффициенты дискретного ПИД регулятора;

**T** – период дискретизации(период, с которым вызывается ПИД регулятор).

Данное уравнение называется *уравнением ПИД регулятора в конечных разностях*.

Чтобы регулятор не выдавал бесконечно больших значений требуемой скорости, его максимальное выдаваемое значение нужно ограничить. Также имеет смысл ограничивать минимальное выдаваемое значение.

Актуальным остаётся вопрос подбора коэффициентов ПИД регулятора. Если их подобрать неправильно, система может стать неустойчивой. Т.е. робот просто начнёт крутиться на 360 градусов и никогда не остановится.

[Go top](#)

Скачать реализацию ПИД регулятора на C++.

---