

## РЕАЛИЗАЦИЯ ПИ-РЕГУЛЯТОРА ПОВЫШЕННОЙ ТОЧНОСТИ ДЛЯ TMS320X24XX

В данной статье рассмотрен вопрос реализации оптимизированного по точности и быстродействию ПИ-регулятора координат электропривода для 16-разрядных микроконтроллеров TMS320x24xx фирмы Texas Instruments семейства “Motor Control”.

Микроконтроллеры TMS320x24xx относятся к классу “Motor Control”, обладают большим количеством периферийных устройств и содержат мощное вычислительное ядро, способное выполнять операцию умножения с накоплением за один процессорный такт (DSP-ядро). При этом операция умножения производится между двумя 16-разрядными числами, а 32-разрядный результат предыдущего умножения добавляется в 32-разрядный аккумулятор с преобразованием формата. Такая схема позволяет описывать любые передаточные функции в разностном виде и, приводя их к набору коэффициентов и выборок/результатов, попарно перемножать, суммируя их в аккумуляторе для получения следующего результата. Так как операция умножения всегда проводится над 16-разрядными числами, то и рассчитанный 32-разрядный ответ требуется привести к 16-разрядному в том случае, если он используется в расчете на следующей итерации. Отсюда видно, что при малых значениях коэффициентов и малых входных сигналах уменьшение точности за счет отброса лишних разрядов, не вмещающихся в 16-разрядный формат, приводит к потере связи между входным и выходным сигналами. В случае с ПИ-регулятором оказывается, что при малом входном рассогласовании интегральная составляющая отключается и регулятор становится пропорциональным, поэтому важно при реализации заложить необходимую точность без существенного увеличения времени расчета.

### Схемы классического ПИ-регулятора в цифровой системе

Передаточная функция непрерывного ПИ-регулятора, реализуемого на аналоговой технике (например, на базе операционных усилителей), имеет вид:

$$W(p) = K_n + \frac{1}{T_u p}, \quad (1)$$

где  $K_n$  – коэффициент передачи пропорциональной части;  $T_u$  – постоянная интегрирования. Если обозначить входной сигнал, который обычно является величиной рассогласования между заданным значением регулируемой величины и ее фактическим значением, снимаемым с датчика обратной связи, как

$$e(t) = x_{зад}(t) - x_{oc}(t), \quad (2)$$

а выходное управляющее воздействие, подаваемое на вход объекта управления, как  $u(t)$ , то передаточной функции ПИ-регулятора будет соответствовать дифференциальное уравнение:

$$u(t) = K_n \cdot e(t) + \frac{1}{T_u} \cdot \int_0^t e(t) dt. \quad (3)$$

Продифференцируем исходное уравнение непрерывного ПИ-регулятора, чтобы избавиться от интеграла:

$$\frac{du(t)}{dt} = K_n \cdot \frac{de(t)}{dt} + \frac{1}{T_u} \cdot e(t). \quad (4)$$

Для перехода от непрерывного уравнения к разностному заменим непрерывные переменные дискретными, дифференциалы этих переменных – разностями, а приращение времени  $dt$  – величиной интервала квантования по времени  $T$ . Получим:

$$\frac{u_k - u_{k-1}}{T} = K_n \cdot \frac{e_k - e_{k-1}}{T} + \frac{1}{T_u} \cdot e_k. \quad (5)$$

После очевидных алгебраических преобразований разностное уравнение ПИ-регулятора примет вид:

$$u_k = u_{k-1} + \left( K_n + \frac{T}{T_u} \right) \cdot e_k - K_n \cdot e_{k-1}. \quad (6)$$

Обозначим коэффициенты:  $K_0 = K_n + \frac{T}{T_u}$  и  $K_1 = K_n$ .

С учетом полученных коэффициентов уравнение будет выглядеть следующим образом:

$$u_k = u_{k-1} + K_0 \cdot e_k - K_1 \cdot e_{k-1}. \quad (7)$$

На рис. 1 показана структурная схема такого регулятора. Звено  $Z^{-1}$  – запаздывание на такт.

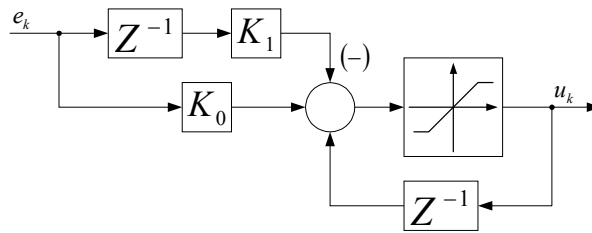


Рис. 1.

Подобная схема удобна для реализации, но имеет один существенный недостаток, не позволяющий использовать ее в большинстве систем управления. Для этой схемы невозможна отдельная настройка коэффициентов регулятора, так как интегральная составляющая выражена неявно, и значение ее постоянной времени влияет на соответствующий коэффициент разностного уравнения совместно с коэффициентом пропорциональной составляющей.

Разделим интегральную и пропорциональную составляющую:

$$u_k = u_{k-1} + K_n \cdot e_k - K_n \cdot e_{k-1} + K_u \cdot e_k, \quad (8)$$

где

$$K_u = \frac{T}{T_u}. \quad (9)$$

Структурная схема такого ПИ-регулятора показана на рис. 2.

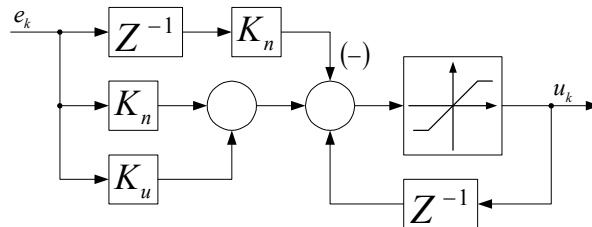


Рис. 2.

Для ПИ-регулятора можно синтезировать также другие структурные схемы. Они строятся исходя из раздельного формирования интегрального и пропорционального каналов. Примеры таких схем рассмотрены в [1] на странице 95, из которых следует проанализировать лишь вторую и четвертую.

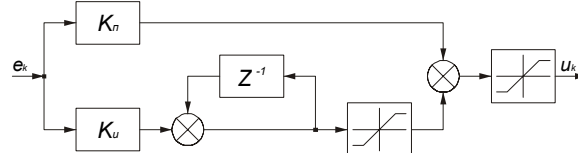


Рис. 3.

Такая схема наиболее точно соответствует аналоговой реализации ПИ-регулятора (аппаратное ограничение интегральной составляющей в рамках формата на схеме не показано). Выход из насыщения такой схемы оказывается затянутым по сравнению со схемой, полученной из уравнения передаточной функции, так как интегральная составляющая функционирует отдельно от выхода регулятора. В [2] и [3] предлагается использовать схему ПИ-регулятора с динамическим списыванием интегральной составляющей.

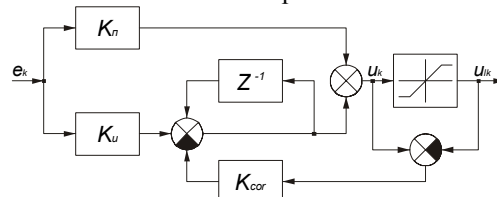


Рис. 4.

Такая схема, как и схема, синтезированная по непрерывной передаточной функции, обеспечивает быстрый выход регулятора из насыщения. При этом интегральная составляющая не зависит от пропорциональной, когда регулятор работает в линейном режиме. При выходе регулятора в насыщение, интегральная составляющая списывается в зависимости от значения пропорциональной составляющей. В [2] и [3] значение коэффициента коррекции  $K_{cor}$  рекомендуют рассчитывать из отношения интегрального коэффициента к пропорциональному, однако улучшения динамики для этого значения замечено не было, а наблюдался лишь затянутый выход из насыщения. При единичном значении этого коэффициента схема приближается к схеме рис. 2. Программная реализация схемы рис. 2 оказывается несколько проще реализации схемы рис. 4, поэтому в качестве основной выбираем схему рис. 2.

### Программная реализация ПИ-регулятора на TMS320x24xx

Для программной реализации ПИ-регулятора надо определить, в каком формате будут представлены все переменные. Сделать это требуется с учетом специфики программирования данного процессора, так как это кардинальным образом влияет на время расчета.

Коэффициент интегрирования может лежать в диапазоне от очень маленького значения до 0,5 (значения больше 0,5 уже не удовлетворяют основному правилу преобразования аналоговой системы к дискретной – теореме Котельникова), поэтому выберем формат 0.16 со знаком (знак хранится в старшем значащем разряде и одновременно в первом разряде после запятой).

После анализа синтеза регуляторов для различных систем управления было выявлено, что коэффициент пропорциональной части может по абсолютному значению достигать 100, поэтому представим его в формате 8.8.

Параметры ограничения на выходе регулятора необходимо варьировать в зависимости от решаемой задачи, поэтому определим переменные в формате 4.12, устанавливающие верхнюю и нижнюю границы ограничения. Выход регулятора настраивается на формат 4.12.

Сигнал ошибки вводится в систему в формате 4.12. При умножении коэффициента интегральной части на сигнал ошибки получается выходной формат 4.28, что совпадает с форматом выходной величины. Старшие 16 разрядов выводятся, а младшие 16 используются для точного интегрирования при малых рассогласованиях, ведь при малых значениях рассогласования на входе в старшем слове оказывается «ноль», и если не интегрировать младшее слово, то интегральная составляющая регулятора отключится.

При резких изменениях задания пропорциональная составляющая может существенно выйти за пределы формата 4.12, так как она получается в результате перемножения мгновенного изменения рассогласования в формате 4.12 на коэффициент пропорциональной составляющей в формате 8.8, поэтому в программе следует предусмотреть дополнительное ограничение пропорциональной составляющей на уровне максимума и минимума в формате 4.12.

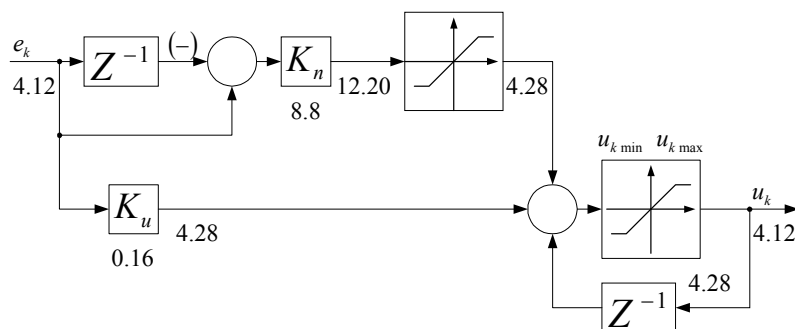


Рис. 5.

Структурная схема регулятора, с учетом вышесказанного, представлена на рис. 5. Напишем программу для этой схемы.

Сначала приведем фрагмент программы для расчета текущего рассогласования и сохранения предыдущего к умножению на пропорциональный коэффициент.

```
setc OVM
ldp #piCont_vars
lt piCont_ek ; загрузка e[k-1] в TREG
lacc piCont_xSet, 16
sub piCont_xFdBk, 16
sach piCont_ek ; сохранение новой ошибки
```

Далее программа раздельно умножает на пропорциональный коэффициент предыдущее и текущее рассогласование, вычитая их в аккумуляторе, попутно рассчитав интегральную составляющую, которая пока остается в PREG.

```
spm 0
mpy piCont_kp ; расчет kp*e[k-1]
ltp piCont_ek ; e[k]->TREG, PREG(kp*e[k-1])->ACC
mpy piCont_kp ; расчет kp*e[k] -> PREG
mpys piCont_ki ; расчет ki*e[k], ACC <- ACC - PREG
neg ; ACC = -ACC
```

После необходимо преобразовать изменение пропорциональной составляющей в аккумуляторе из формата 12.20 в 4.28, что осуществляется восьмикратным повторением команды нормирования числа. Если число не вписывается в формат 4.28, то флаг TC будет об этом сигнализировать.

```
bcnd no_p_saturation, EQ
rpt #7
norm * ; приведение к формату 4.28
bcnd no_p_saturation, NTC
bcnd positive_p_saturation, GT
bcnd negative_p_saturation, LT
```

Если ограничение не требуется, то можно провести суммирование остальных составляющих (интегральной и значения предыдущего выхода регулятора).

```
no_p_saturation:
```

```
арас      ; добавление интегральной составляющей
adds     piCont_ukLow
add      piCont_ukHigh, 16 ; сложение с u[k-1]
```

Далее необходимо произвести программное ограничение выхода регулятора в заданных пределах.

```
sub      piCont_ukMax, 16
bcnd    positive_saturation, GT
add      piCont_ukMax, 16
sub      piCont_ukMin, 16
bcnd    negative_saturation, LT
add      piCont_ukMin, 16
```

Завершаем программу сохранением нового выхода.

```
no_saturation:
sac1    piCont_ukLow
sach    piCont_ukHigh
ret
```

Если происходит программное ограничение, то обработчик его выглядит следующим образом.

```
positive_saturation:
lacc    piCont_ukMax, 16
b       no_saturation
negative_saturation:
lacc    piCont_ukMin, 16
b       no_saturation
```

В случае аппаратного ограничения обработчик обеспечивает расчет оставшихся составляющих.

```
positive_p_saturation:
lacc    #7FFFh, 15
sfl
b       no_p_saturation
negative_p_saturation:
lacc    #8000h, 15
sfl
b       no_p_saturation
```

Текст модуля ПИ-регулятора написан с использованием прямой адресации для того, чтобы проще прочитывался алгоритм его работы. В работе удобнее использовать косвенную адресацию, чтобы можно было создавать несколько структур данных под разные регуляторы с разными параметрами, а модулю при его вызове передавать только начальный адрес структуры. Таким образом, система может работать с любым количеством регулируемых величин.

Данный регулятор успешно используется в составе программного обеспечения преобразователей частоты серии «Универсал», в контуре регулирования скорости вентильно-индукторного многофазного привода специального применения и в ряде других задач.

#### Список использованной литературы

1. А.С. Анучин, В.Ф. Козаченко / Архитектура, Система Команд, Технология Проектирования и Отладки Специализированных Сигнальных Микроконтроллеров для Управления Двигателями / Лабораторный практикум / Москва / Изд-во МЭИ / 2001
2. Field Orientated Control of 3-Phase AC-Motors / Literature Number: BPRA073 / Texas Instruments Europe / February 1998
3. Implementation of a Speed Field Orientated Control of Three Phase AC Induction Motor using TMS320F240 / Literature Number: BPRA076 / Texas Instruments Europe / March 1998