



# Теория и практика автоматического управления в робототехнике

Андрей Антонов (Волгоград)

В статье представлены базовые понятия теории автоматического управления и рассмотрены практические моменты компьютерной реализации управления движением мобильного робота.

## Автоматическое управление

Одной из основных задач, решаемых в робототехнике, является задача управления. Если техническая система работает без участия человека, то её называют «автоматическая система». Разработкой общих принципов реализации автоматического управления, методов анализа и синтеза систем автоматического управления занимается научная дисциплина под названием «Теория автоматического управления» (ТАУ). ТАУ является частью технической кибернетики – науки, изучающей процессы управления и обработки информации в технических системах. Зачастую в качестве синонима названию «Теория автоматического управления» используется «Теория автоматического регулирования». В зарубежной литературе применяется термин "Control Theory".

Сталкиваясь с задачами управления, мы всегда имеем дело с двумя объектами – управляемым, или объектом управления, и управляющим, который называют регулятором или контроллером. Воздействие регулятора на объект управления производится обыч-

но через исполнительные устройства, называемые приводами. Приводы усиливают и преобразуют сигнал управления. Например, электрический управляющий сигнал трансформируется в перемещение руки манипулятора в промышленном роботе. Для «осязания» регулятором объекта управления используются датчики – устройства, измеряющие те характеристики объекта, которыми необходимо управлять. Различные элементы обмениваются между собой информацией, используя каналы связи, в качестве которых может выступать электрический ток, воздух (пневматические системы), жидкость (гидравлические системы), компьютерные сети [1].

Для реализации управления технической системой используется принцип обратной связи. Основная идея обратной связи состоит в том, что текущее значение на выходе системы непрерывно сравнивается с требуемым эталонным значением (уставкой) и состояние входа системы корректируется таким образом, чтобы свести к нулю любое отклонение выхода от уставки.

Другими словами, если значение контролируемого физического параметра на выходе велико, то производятся корректировки параметра на входе, которые приводят к снижению выходной величины. Если же значение на выходе меньше уставки, то вход корректируется таким образом, чтобы увеличить значение на выходе системы. Таким образом, реализуется управление по ошибке, стремящееся свести рассогласование к нулю. Поскольку информация в системе с обратной связью передаётся по замкнутому контуру, то такие системы ещё называют замкнутыми системами.

Обобщённо идея управления состоит в использовании информации о состоянии объекта для выработки управляющего сигнала. Схема системы управления с обратной связью показана на рисунке 1.

Если же выход системы в расчёт не принимается, то такая система является разомкнутой. В ней регулятор не получает никакой информации о реальном состоянии объекта. В этом случае, чтобы иметь возможность построить программу управления, должно быть точно известно поведение объекта управления. Разомкнутые системы часто называют системами программного управления. Примером системы программного управления является электронное информационное табло. Структурная схема разомкнутой системы управления приведена на рисунке 2.

Типы задач, решаемых с использованием автоматических систем управления:

- стабилизация – это поддержание заданного режима работы (уставка равна константе);
- программное управление – это управление с использованием заданной программы. В этом случае задающий сигнал меняется, но является заранее неизвестным;
- слежение за неизвестным сигналом.

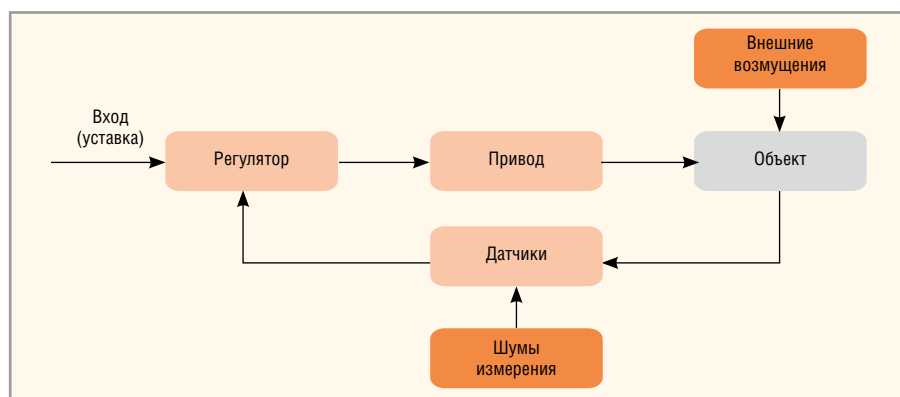


Рис. 1. Система автоматического управления с обратной связью

## РЕГУЛЯТОРЫ

Рассмотрим, каким образом могут быть реализованы регуляторы в системах автоматического управления.

Первым вариантом является реализация пропорционального управления, когда величина коррекции уставки пропорциональна величине ошибки. В результате небольшая ошибка будет приводить к небольшой корректировке, а большая ошибка приведёт к большому корректирующему действию. То есть выходной сигнал регулятора  $u_p(t)$  будет пропорционален ошибке слежения  $e(t)$ :

$$u_p(t) = k_p e(t), k_p > 0,$$

где  $k_p$  – усиление регулятора, положительная константа.

Фактически пропорциональный регулятор является усилителем. При использовании пропорционального, или, как его ещё называют, П-регулятора, выход системы будет всегда меньше значения уставки. Причиной является то, что П-регулятор имеет на выходе ненулевое значение только в случае ненулевого входа. Если ошибка слежения отсутствует, то пропорциональный регулятор не создаёт выходного сигнала. Практически на входе мы всегда будем иметь ненулевое значение в стационарном состоянии. Следствием этого будет сохранение некоторой статической ошибки. Эта ошибка может быть уменьшена за счёт увеличения усиления регулятора  $k_p$ , но при слишком больших значениях это может привести к автоколебаниям и потере устойчивости системы. Значение регулируемой величины в П-регуляторе никогда не стабилизируется на заданном значении, поэтому для устранения статической ошибки нам нужен другой способ. Конечно, можно изначально задать большее значение уставки, но существует вид регулятора, который позволяет устранить статические ошибки автоматически. Это приводит нас к интегральному управлению.

Интегральное управление – это стратегия, основанная на общей накопленной ошибке. П-регулятор основывается только на отслеживании мгновенной ошибки. Если ошибка слежения мала, то пропорциональный регулятор теряет свою эффективность (так как результирующие корректировки также будут малы). Одним из способов «усиления» таких маленьких статических оши-

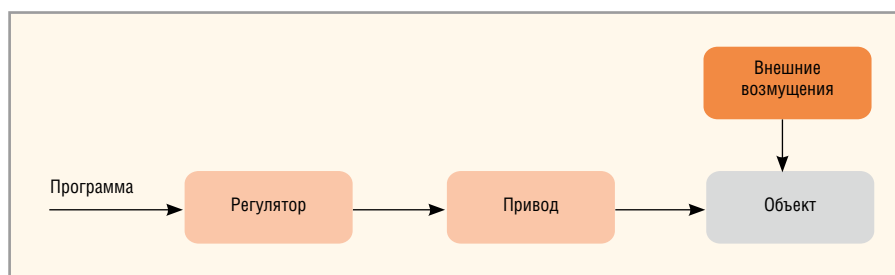


Рис. 2. Система программного управления

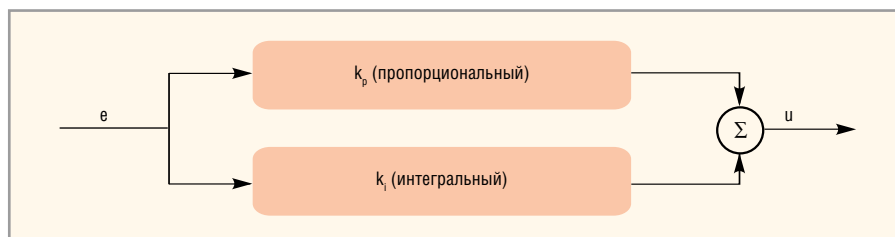


Рис. 3. Схема ПИ-регулятора

бок является использование накопленной ошибки с течением времени, что обеспечивает значительный уровень управляющего сигнала. С другой стороны, если ошибка слежения равна нулю, то накопленное значение также будет нулевым. В этом заключена идея интегрального управления.

Выход интегрального регулятора пропорционален интегралу ошибки слежения за промежуток времени:

$$u_i = k_i \int_0^t e(\tau) d\tau,$$

где  $k_i > 0$  – константа.

В случае дискретного времени в цифровых системах интеграл заменяется суммой ошибок, и интегральный регулятор реализуется в виде сумматора значений ошибок. Этот подход легко реализуется в виде схемы рекурсивного обновления [2]:

$$E_t = \delta t e_t + E_{t-1},$$

$$u_{i,t} = k_i E_t,$$

где  $E_t$  – накопленная ошибка на временном шаге  $t$ ,  $k_i$  – интегральное усиление и  $u_{i,t}$  – выход интегрального регулятора в момент времени  $t$ .

Эта дискретная схема обновления предполагает, что управляющие действия производятся периодически. Коэффициент  $\delta t$  представляет собой промежуток времени между последовательными управляющими воздействиями, выраженный в единицах измерения времени (если мы измеряем время в секундах и производим 100 управляющих действий в секунду, то  $\delta t = 0,01$ , если же мы измеряем время в днях и производим одно обновление

в день, тогда  $\delta t = 1$ ). Конечно,  $\delta t$  могло бы быть включено в коэффициент усиления регулятора  $k_i$ , но это означало бы, что при изменении частоты обновления усиление регулятора тоже бы изменилось. Поэтому лучше их разделить:  $\delta t$  включает временной интервал между последовательными обновлениями, а  $k_i$  независимо управляет вкладом интегрального члена на выход регулятора.

Зачастую пропорциональный и интегральный регуляторы включаются в схему параллельно (см. рис. 3). Такая реализация называется пропорционально-интегральным регулятором, или ПИ-регулятором, который наиболее часто используется на практике.

Выход интегрального регулятора зависит не только от мгновенного значения ошибки, но и от интеграла (или суммы) наблюдаемых ошибок слежения с начального момента времени. Эта зависимость от предыдущих значений приводит к тому, что интегральный регулятор обладает нетривиальной динамикой, которая может изменить качественное поведение всей замкнутой системы. В частности, интегральный регулятор может приводить к колебаниям, даже если управляемая система им не подвержена. При сохранении положительной ошибки слежения интегральная составляющая в регуляторе начнёт расти и результатом будет положительное значение входа системы, которое сохраняется даже после того, как ошибка слежения будет устранена. Вследствие чего выходной сигнал системы управления будет «промахиваться», и ошибка слежения станет отрицательной. В свою очередь ошиб-

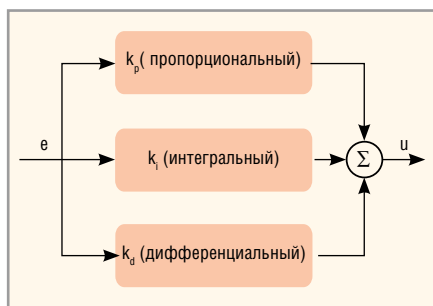


Рис. 4. Схема ПИД-регулятора

ка слежения уменьшает значение интегрального компонента.

В зависимости от выбранных значений для коэффициентов усиления регулятора  $k_p$  и  $k_i$ , эти колебания могут затухать более или менее быстро. Настройка регулятора представляет собой поиск таких значений параметров, которые бы приводили к приемлемому динамическому поведению системы с обратной связью.

Одним из основных положений ТАУ является линейность отношения между входом и выходом управляемого объекта:  $y = Nu$ . Это означает, что в установленном режиме мы имеем ненулевой выход при отличном от нуля входе. В цепи обратной связи мы пытаемся минимизировать ошибку слежения, но также мы используем эту ошибку в качестве входа регулятора. Каким же образом при отсутствии ошибки мы можем поддерживать выход ненулевым?

В случае с пропорциональным регулятором этого сделать нельзя. При пропорциональном управлении системе необходимо поддерживать остаточную, отличную от нуля, ошибку слежения для того, чтобы получить ненулевой выход. Но мы можем управлять ошибкой слежения вплоть до нуля и при этом сохранить выход системы управления ненулевым, при условии, что мы можем включить интегральное звено в регулятор.

Можно также включить дифференциальное звено в регулятор. В то время как интегральное звено отслеживает прошлые значения, дифференциальный регулятор пытается предвидеть будущее. Производная является скоростью изменения некоторой величины. Так, если производная ошибки слежения положительна, это означает, что ошибка слежения в настоящий момент растёт. И наоборот. Таким образом, корректировку можно производить сразу, противодействуя, тем самым, росту ошибки, даже если её значение ещё мало.

Выход дифференциального регулятора пропорционален производной ошибки слежения:

$$u_d(t) = k_d \frac{de(t)}{dt},$$

где  $k_d > 0$  – константа.

В случае компьютерной реализации с дискретным временем, можно аппроксимировать производную  $e$ , изменением  $e$  с предшествующего момента времени. Дифференциальный регулятор может быть реализован в виде:

$$u_{d,t} = k_d \frac{e_t - e_{t-1}}{\delta t},$$

где  $\delta t$  – временной интервал между последовательными обновлениями.

Подобно интегральному регулятору, дифференциальный регулятор зависит от предыдущих значений  $e$ , следовательно, привносит свою нетривиальную динамику в систему.

Интегральные регуляторы очень «доброжелательны» и часто используются вместе с пропорциональными регуляторами, что нельзя сказать о дифференциальном управлении. Проблемой в последнем случае является потенциальное присутствие высокочастотного шума на входе регулятора.

Шумовая составляющая будет осциллировать возле нулевой отметки  $e$ , таким образом, в интегральном регуляторе она аннулируется. Однако если взять производную зашумлённого сигнала, то это только усилит влияние шума. Поэтому часто бывает необходимо сглаживание сигнала. Это добавляет сложности и нетривиальной динамики в регулятор. Также появляется риск «потери цели» при дифференциальном управлении. Если сгладить сигнал слишком сильно, то будут теряться изменения в сигнале, которые должен использовать дифференциальный регулятор.

Ещё одной проблемой, связанной с дифференциальным управлением, является влияние резкого изменения уставки, что приводит к моментальному скачку значения на выходе дифференциального регулятора.

В то время как пропорциональное управление занимает центральное место в системах с обратной связью, а интегральное управление требуется для устранения статической ошибки, на практике дифференциальное управление используется не так уж и часто. Это не должно быть неожиданностью. Исследования показывают, что более

95% регуляторов, используемых в различных прикладных областях, принадлежат к типу ПИ-регуляторов [2].

Регулятор, включающий все три компонента (пропорциональный, интегральный и дифференциальный), называют ПИД-регулятором (см. рис. 4). Выход этого регулятора представляет собой комбинацию трёх его составляющих:

$$u_{PID}(t) = u_p(t) + u_i(t) + u_d(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{de(t)}{dt}.$$

Такая форма представления удобна для теоретической работы. Для практических приложений используется альтернативное представление, в котором коэффициенты объединены в общий коэффициент усиления:

$$u_{PID}(t) = k \left[ e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right].$$

Оба новых параметра  $T_i$  и  $T_d$  имеют ту же размерность, что и время. Обе формулы для  $u_{PID}(t)$  эквивалентны и их параметры соотносятся как:

$$k = k_p, T_i = \frac{k_p}{k_i}, T_d = \frac{k_d}{k_p}.$$

Конечно, числовые значения параметров различны. При сравнении значений параметров регулятора нужно не забыть, к какому из двух представлений они относятся.

В схемах обновления с дискретным временем интегральная и дифференциальная составляющие достаточно просто реализуются в программном коде. Следующий класс реализует ПИД-контроллер:

```
class PidController:
    def __init__( self, kp, ki, kd=0 ):
        self.kp, self.ki, self.kd = kp, ki, kd
        self.i = 0
        self.d = 0
        self.prev = 0
    def work( self, e ):
        self.i += DT*e
        self.d = ( e - self.prev )/DT
        self.prev = e
        return self.kp*e + self.ki*self.i + self.kd*self.d
```

Здесь множитель  $DT$  обозначает размер шага  $\delta t$ , который задаёт интервал между последовательными управляющими воздействиями и выражается в тех же единицах, что и время.

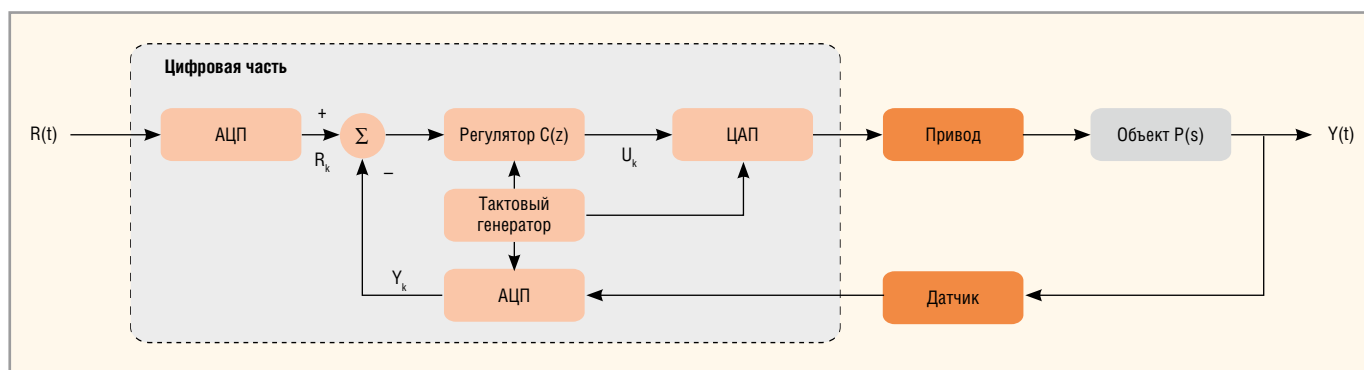


Рис. 5. Цифровая часть системы управления

**ПРИМЕР РЕАЛИЗАЦИИ ЦИФРОВОЙ ЧАСТИ СИСТЕМЫ УПРАВЛЕНИЯ ДВИЖЕНИЕМ МОБИЛЬНОГО РОБОТА**

Большинство современных регуляторов, использующихся в системах управления, реализуются с применением микроконтроллеров и микропроцессоров. В случае реализации контроллера на базе компьютерной системы управления процесс чтения с аналоговых входов и установка выходов в определённые состояния происходят с некоторой периодичностью. По сравнению с вариантом аналоговой реализации это является недостатком, так как процесс выборки вносит временную задержку в контур управления. На рисунке 5 показана общая структурная схема системы управления, на которой выделена цифровая часть [3].

При программной реализации закона управления удобно будет разделить последовательность команд при обработке прерывания на четыре функции, как это показано на рисунке 6. Очень важно чтение входного сигнала с датчиков и синхронизация записи управляющего сигнала в цифро-аналоговый преобразователь (ЦАП) с правильной частотой. Поэтому эти процессы находятся в первой функции. В следующей функции, после сохранения значения счётчика и извлечения значений ЦАП, которые уже вычислены на предыдущем шаге, генерируются значения эталонных напряжений. Следом идут функции, включающие фильтры, которые производят скалярное или же векторное управление выходами. Наконец, реализуется пользовательский интерфейс для проверки значений параметров, а также для дальнейшей настройки и отладки.

Создание эталонной траектории является стартовой точкой. Эталонная траектория используется в качестве целевого значения в процессе управле-

ния и представляет собой набор значений уставки при каждой обработке прерывания. В связи с этим она называется профилем движения. Правильное создание этого профиля важно, поскольку из-за того, что целью практического сервоуправления является не регулирование, а слежение, ошибки отслеживания и фиксации очень чувствительны к нему. Общий профиль трапецевидной формы вычисляется просто и требует постоянного тока в периоды ускорения и замедления. Однако сильные толчки (вследствие резких изменений ускорения и, следовательно, резких изменений приложенной силы) могут являться причиной серьёзных повреждений в динамических системах и вызывать нежелательные вибрации. Поэтому нужен сглаженный профиль, что требует более затратных по времени вычислений. Простой профиль движения можно задать при помощи полинома третьей степени:

$$\begin{aligned} y(t) &= a + bt + ct^2 + dt^3, \\ \dot{y}(t) &= b + 2ct + 3dt^2, \\ \ddot{y}(t) &= 2c + 6dt. \end{aligned}$$

На практике, уравнения, описывающие профиль движения, нормализуют и программируют в виде функций. Приняв ограничения:

$$\begin{aligned} y(0) &= 0, y(1) = 1, \\ \dot{y}(0) &= 0, \dot{y}(1) = 0, \end{aligned}$$

получим уравнения, которые уже реализуются в программном коде:

$$\begin{aligned} y\left(\frac{t}{T}\right) &= \left[ 3\left(\frac{t}{T}\right)^2 - 2\left(\frac{t}{T}\right)^3 \right] S, \\ \dot{y}\left(\frac{t}{T}\right) &= \left[ \frac{6}{T}\left(\frac{t}{T}\right) - \frac{6}{T}\left(\frac{t}{T}\right)^2 \right] S, \\ \ddot{y}\left(\frac{t}{T}\right) &= \left[ \frac{6}{T^2} - \frac{12}{T^2}\left(\frac{t}{T}\right) \right] S, \\ \ddot{\ddot{y}}\left(\frac{t}{T}\right) &= -\frac{12}{T^3} S. \end{aligned}$$

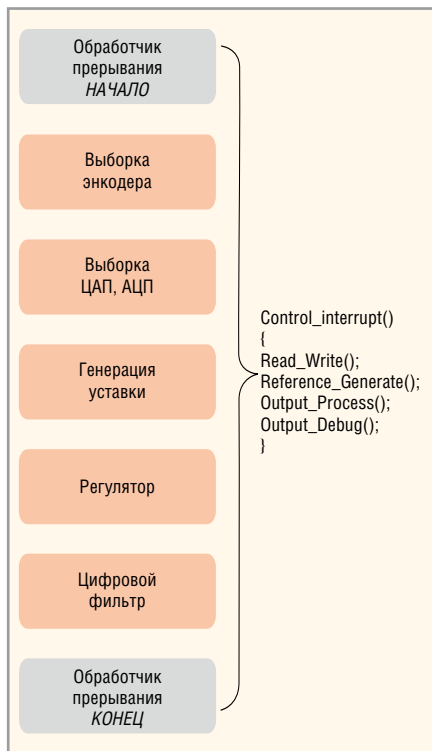
Здесь S – расстояние и T – время движения.

Во время работы для высокоскоростных сервосистем профиль скачка является константой, с большими изменениями на старте и при остановке. Это означает, что требуется высокая механическая прочность. Даже при том, что вычислительные затраты возрастут, возможно использование профилей, описываемых полиномами пятого или седьмого порядка, для того, чтобы удовлетворить условиям ускорения или скачка.

Профиль движения часто сохраняется в оперативной памяти (ОЗУ) в виде таблицы или же помещается в функцию Reference\_Generate() обработки прерывания, как это показано на рисунке 6. В большинстве задач слежения в сервоуправлении создаётся профиль позиции. Тем не менее скорость и ускорение, также вычисляемые в функции обработки прерывания в реальном времени, используются как входы для предупредования. Они улучшают результаты отслеживания и работы алгоритма:

$$\begin{aligned} POS_{ref} &= PosTabe(\text{указатель}) \times \text{Расстояние} \\ VEL_{ref} &= POS_{ref} - POS_{ref,old} \quad (1) \\ ACC_{ref} &= VEL_{ref} - VEL_{ref,old} \end{aligned}$$

Здесь  $VEL_{ref}$  и  $ACC_{ref}$  не имеют реальных эталонных значений скорости и ускорения. На практике эталонная скорость масштабируется на время прерывания, а ускорение – на квадрат времени прерывания, соответственно. Кроме того, ограничение разрядности регистров цифрового сигнального процессора (DSP) или микропроцессора может стать причиной ошибок. Происходит это из-за потери значащих разрядов во время вычисления профиля скорости из профиля позиции с использованием простой формулы разности в алгоритме (1). Это озна-



**Рис. 6. Процедура обработки прерывания для цифровой части системы управления**

чает, что профиль скорости включает периодический сигнал шума. Этот вид цифровой ошибки создаёт реальный шум в движении и ухудшает работу системы управления, если профили скорости, ускорения и скачка используются в качестве входов для систем предупреждения в высокоскоростном слежении.

Вместо простой разности в алгоритме (1) можно использовать численные дифференциальные уравнения (2) с обратными дифференциалами первого порядка:

$$\begin{aligned} \dot{y}_k &= \frac{y_k - y_{k-1}}{\Delta t} \\ \ddot{y}_k &= \frac{y_k - 2y_{k-1} + y_{k-2}}{\Delta t^2} \\ \dddot{y}_k &= \frac{y_k - 3y_{k-1} + 3y_{k-2} - y_{k-3}}{\Delta t^3} \end{aligned} \quad (2)$$

Сохранение каждого профиля в виде таблицы может быть ещё одним способом, но только при условии наличия достаточного количества памяти.

Важно отметить, что задание максимальных значений параметров движения в управляющей системе, таких как скорость, ускорение или даже скачок, следует производить на шаге создания профиля. Если профиль позволяет системе нарушать заданные ограничения, то вероятность непредсказуемого поведения в процессе эксплуатации сильно возрастает при пропуске импульсов энкодером.

Системы, работающие с непрерывным временем, преобразуются в системы с дискретным временем, используя Z-преобразование. Система с дискретным временем используется для получения математической модели, которая описывает динамику физического процесса в точках выборки, хотя физический процесс всё ещё является системой непрерывного времени. Преобразование Лапласа используется для анализа системы управления в области изображений, используя комплексные числа. В большинстве случаев при разработке контроллеров и фильтров используются инструменты в области изображений. Для того чтобы реализовать это в виде программного кода, необходимо понять Z-преобразование. Все контроллеры и фильтры, созданные в области изображений, могут быть легко преобразованы в программном коде через Z-преобразование, являющееся видом цифровых разностных последовательностей.

В качестве примера рассмотрим ПИД-регулятор. В форме операторных функций этот контроллер имеет вид:

$$\frac{Y(s)}{E(s)} = K_p + \frac{K_i}{s} + sK_v \quad (3)$$

Существует несколько способов преобразования из частотной области в дискретную. Для сохранения стабильности часто используются обратные алгоритмы Эйлера и Тастина [4]. Несмотря на то, что алгоритм Тастина является более точным, в следующей процедуре используется алгоритм Эйлера. После подстановки обратного уравнения Эйлера в (3)

$$s \cong \frac{z-1}{zT}$$

получим следующий дискретный вид:

$$\frac{Y(z)}{E(z)} = \frac{\alpha + \beta z^{-1} + \gamma z^{-2}}{T(1-z^{-1})}$$

где

$$\begin{aligned} \alpha &= K_p T^2 + K_p T + K_v, \\ \beta &= -K_p T - 2K_v, \\ \gamma &= K_v. \end{aligned}$$

Преобразуем, используя разностное уравнение:

$$T(y_k - y_{k-1}) = \alpha e_k + \beta e_{k-1} + \lambda e_{k-2}$$

$$y_k = y_{k-1} + \frac{1}{T}(\alpha e_k + \beta e_{k-1} + \lambda e_{k-2}) \quad (4)$$

$$y_k - y_{k-1} = \frac{\alpha}{T} e_k + \frac{\beta}{T} e_{k-1} + \frac{\gamma}{T} e_{k-2}.$$

В программном коде это может выглядеть примерно следующим образом:

$$\begin{aligned} e_k &= p_{k,ref} - p_k \\ e_k^v &= v_{k,ref} - v_k = (p_{k,ref} - p_{k-1,ref}) - \\ &\quad - (p_k - p_{k-1}) = e_k - e_{k-1} \\ \text{sum}_k &= \text{sum}_{k-1} + e_k = \sum_{j=0}^k e_j \\ y_k &= K_{p,c} \cdot e_k + K_{v,c} (e_k - e_{k-1}) + K_{i,c} \cdot \text{sum}_k \\ y_k - y_{k-1} &= (K_{p,c} \cdot e_k - K_{p,c} \cdot e_{k-1}) + \\ &\quad + K_{v,c} (e_k - e_{k-1}) - K_{v,c} (e_{k-1} - e_{k-2}) + \\ &\quad + (K_{i,c} \cdot \text{sum}_k - K_{i,c} \cdot \text{sum}_{k-1}) = \\ &= (K_{p,c} + K_{v,c} + K_{i,c}) e_k - \\ &\quad - (K_{p,c} + 2K_{v,c}) e_{k-1} + K_{v,c} \cdot e_{k-2}. \end{aligned} \quad (5)$$

Здесь  $p_k$  обозначает позицию,  $v_k$  – скорость,  $ref$  обозначает эталонное значение и  $c$  означает код.

Сравнив параметры в (4) и (5), получим:

$$\begin{aligned} \frac{\alpha}{T} &= K_p + \frac{K_v}{T} + K_i T = (K_{p,c} + K_{v,c} + K_{i,c}), \\ \frac{\beta}{T} &= -K_p - \frac{2K_v}{T} = -(K_{p,c} + 2K_{v,c}), \\ \frac{\gamma}{T} &= \frac{K_v}{T} = K_{v,c}. \end{aligned}$$

Отсюда видно, что существует зависимость между проектной и кодированной формами коэффициентов усилений:

$$\begin{aligned} K_{p,c} &= K_p, \\ K_{v,c} &= \frac{K_v}{T}, \\ K_{i,c} &= K_i T. \end{aligned}$$

При увеличении частоты выборки в той же самой системе кодированный коэффициент усиления  $K_v$  должен быть увеличен, а кодированный коэффициент усиления  $K_i$  должен быть уменьшен. С помощью этого метода разрабатываемый регулятор может быть представлен в виде кода для цифрового сигнального процессора или микропроцессора. Однако для получения приемлемых результатов работы системы управления необходимо проводить анализ и моделирование алгоритмов управления.

**ЛИТЕРАТУРА**

1. Поляков К.Ю. Основы теории цифровых систем управления. Учебное пособие. СПб. СПбГМТУ. 2006.
2. Janert Philipp K. Feedback Control for Computer Systems. O'Reilly. 2014. P. 39–49.
3. Siciliano Bruno, Khatib Oussama. Handbook of Robotics. Springer. 2008. P. 153–155.
4. Pupalakis Peter J. Bilinear Transformation Made Easy. International Conference on Signal Processing and Technology (ICSPAT). Dallas TX. 2000.

