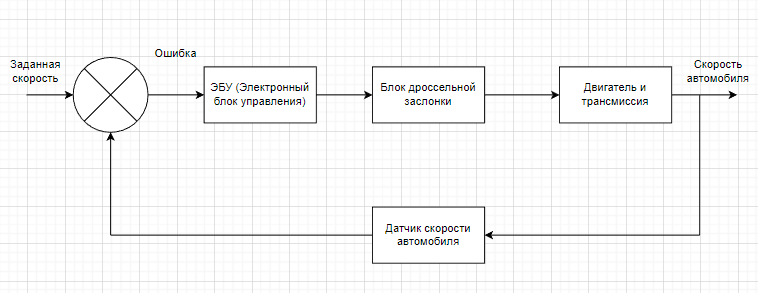
**Система автоматического управления для контроля скорости автомобиля (Круиз-контроль).**

1. **Описание**

Круиз-контроль — устройство, поддерживающее постоянную скорость автомобиля, автоматически прибавляя газ при снижении скорости движения и уменьшая при её увеличении, к примеру, на спусках или подъемах, без участия водителя.

Датчики реагируют на скорость автомобиля и в зависимости от заданных параметров меняется скорость. Если, например, на пути автомобиля встречается спуск, датчик регистрирует повышение скорости и снижает ее путем сбрасывания газа.

1. **Функциональная схема**



Входным сигналом для контроллера является ошибка, представляющая собой разницу между требуемой скоростью и фактической скоростью. В соответствии с этой ошибкой контроллер выдает сигнал дроссельной заслонке. Угол открытия дроссельной заслонки, изменит обороты двигателя, а также может изменить передачу в КПП.

1. **ПИД-регулятор**

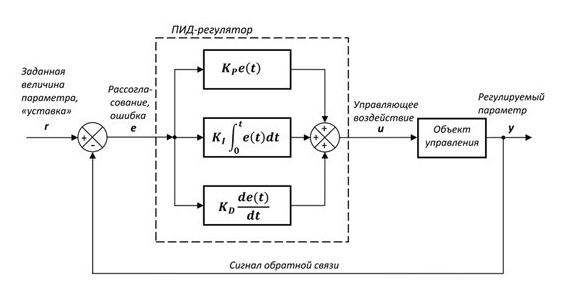
PID представляет собой комбинацию усилительного, интегрального и дифференцируемого сигнала ошибки. Эта комбинация дает очень хороший метод управления, который используется в большинстве промышленных приложений, где каждая часть дает некоторые преимущества для общего отклика системы.

В нашем случае объектом управления является дроссельная заслонка, процессом – степень её открытия, исполнительным блоком – мозги автомобиля (ЭБУ), которые будет высчитывать необходимую скорость.

На входе подается скорость, указанная водителем, а на выходе – фактическая скорость, которая в дальнейшем регулируется.

Реализация исполнительного блока будет представлять из себя ПИД-регулятор (Пропорционально-интегрально-дифференцирующий). ПИД-регулятор принимает сигнал ошибки e(t). Такой сигнал рассчитывается следующим образом: вычитание текущего состояния процесса из заданного. Затем ошибка проходит через три составляющие:

* **Усилительное звено P** — пропорциональная составляющая. Фактически отражает в себе ошибку регулирования. Чем больше ошибка, тем больше командное отклонение регулятора и тем больше будет управляющий сигнал. Данное звено можно описать формулой: . В этой формуле коэффициент отвечает за усиление ошибки.
* **Интегрирующее звено I** – интегральная составляющая. Она, в свою очередь, суммирует разность текущего и заданного значения, то есть ошибки и умножает ее на время прошлого расчета , то есть берет интеграл от ошибки по времени. Данное звено можно описать формулой: . В этой формуле – интегральный коэффициент.
* **Дифференцирующее звено D –** дифференциальная составляющая, представляет собой разность текущей и предыдущей ошибки, деленную на , предназначена для противодействия отклонениям от целевого значения, которые прогнозируются в будущем. Другими словами – это производная от ошибки по времени. Данное звено можно описать формулой: . В этой формуле коэффициент – позволяет настроить вес, или резкость компенсации.



Исходя из вышеперечисленного, управляющий сигнал можно описать следующей функцией, зависящей от t:

Ошибка вычисляется по следующей формуле:

,

где SP(t) – заданное значение и PV(t) – текущее значение.

Напишем передаточные функции для звеньев ПИД регулятора:

* Усилительное звено:
* Интегрирующее звено:
* Дифференцирующее звено:

Все элементы регулятора соединены параллельно, соответственно общая передаточная функция будет являться суммой всех его составляющих:

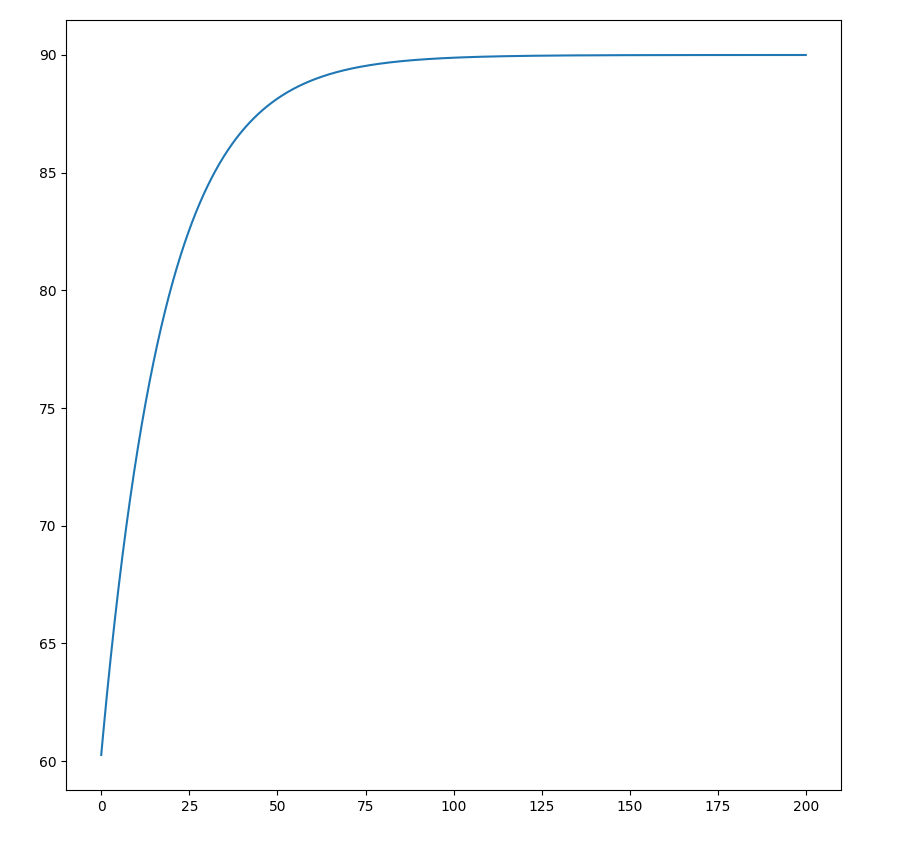
1. **Моделирование цепи управления**

Для наглядности была написана программа на языке Python, которая отрисует график процесса управления данной цепи.

В качестве констант у нас два значения скорости в начале – start\_speed и желаемая скорость – set\_speed. В цикле вычисляем ошибку, дискретное значение управляющего сигнала, а также значение скорости.

def main():  
 *# Начальная скорость в км/ч* start\_speed = 60  
 *# Заданная скорость в км/ч* set\_speed = 90  
 *# Текущая скорость в км/ч* current\_speed = start\_speed  
 *# Предыдущая на шаг измерения скорость в км/ч* previous\_speed = start\_speed  
 *# Разница между предыдущей и текущей скоростями* diff = 0  
 *# Шаг* dt = 0.1  
 *# Коэффициенты ПИД-регулятора  
 # K\_P = 2.2135  
 # K\_I = 0.5845  
 # K\_D = 1.7255* K\_P = 1  
 K\_I = 0  
 K\_D = 0.7255  
 *# Ошибка* current\_error = 0  
 *# Предыдущее значение ошибки* previous\_error = 0  
  
 t = np.arange(0, 200, dt)  
  
 P = 0  
 D = 0  
 I = 0  
 speed = np.zeros(t.size)  
 *# Изменение скорости* for i in range(t.size):  
 current\_error = set\_speed - current\_speed  
 P = K\_P \* current\_error  
 I = I + K\_I \* current\_error  
 D = K\_D \* (current\_error - previous\_error)  
 current\_speed = current\_speed + diff \* dt + ((P + I + D) \* pow(dt, 2) / 2)  
 speed.data[i] = current\_speed  
 diff = current\_speed - previous\_speed  
 previous\_speed = current\_speed  
 previous\_error = current\_error  
  
 plt.gcf().set\_size\_inches(SCREEN\_SIZE, SCREEN\_SIZE)  
 plt.plot(t, speed)  
 plt.show()

Подобрав нужные значения коэффициентов, получаем следующий график.



Данный график отображает работу ПИД – регулятора по установлению и поддержанию скорости на определенном уровне.