

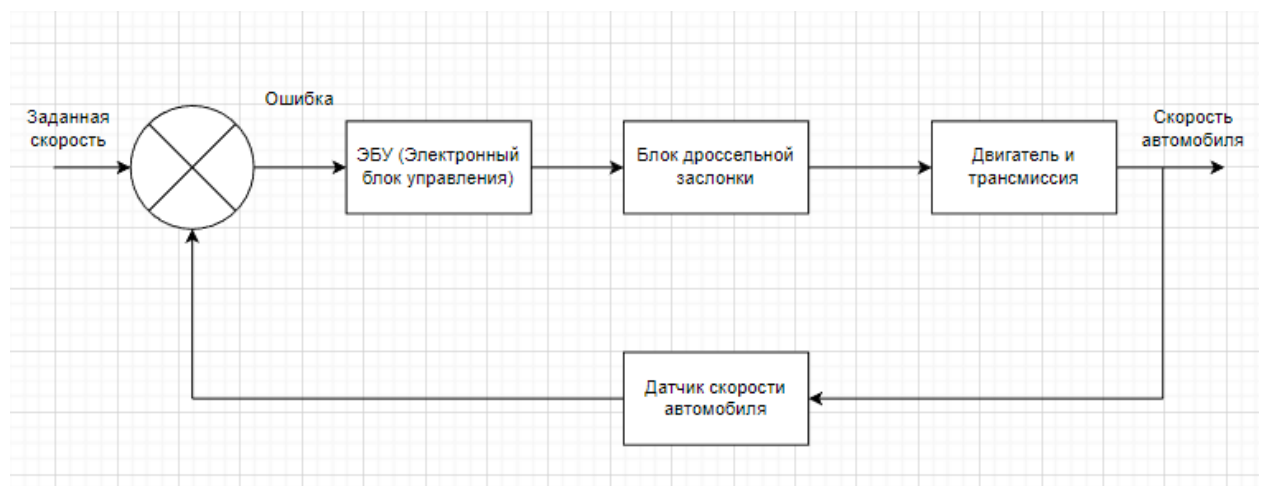
Система автоматического управления для контроля скорости автомобиля (Круиз-контроль).

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.

2. 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.

3. ПИД-регулятор

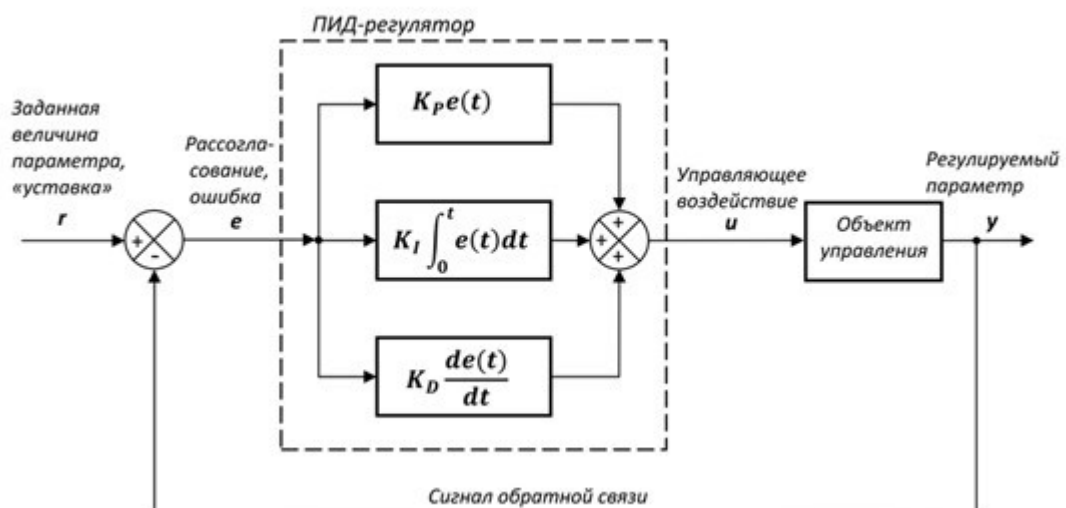
PID представляет собой комбинацию усилительного, интегрального и дифференцируемого сигнала ошибки. Эта комбинация дает очень хороший метод управления, который используется в большинстве промышленных приложений, где каждая часть дает некоторые преимущества для общего отклика системы.

В нашем случае объектом управления является дроссельная заслонка, процессом – степень её открытия, исполнительным блоком – мозги автомобиля (ЭБУ), которые будут 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 Р** — 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: $P(t) = K_P * e(t)$. В этой формуле коэффициент K_P 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 $d\tau$, то есть берет интеграл от ошибки по времени. Данное звено можно описать формулой: $I(t) = K_I * \int_0^t e(t)dt$. В этой формуле K_I – интегральный коэффициент.
- **Дифференцирующее звено D** – дифференциальная составляющая, представляет собой разность текущей и предыдущей ошибки, деленную на dt , предназначена для противодействия отклонениям от целевого значения, которые прогнозируются в будущем. Другими словами – это производная от ошибки по времени. Данное звено можно описать формулой: $D(t) = K_D * \frac{de(t)}{dt}$. В этой формуле коэффициент K_D – позволяет настроить вес, или резкость компенсации.



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

$$MV(t) = K_P * e(t) + K_I * \int_0^t e(t)dt + K_D * \frac{de(t)}{dt}$$

Ошибка вычисляется по следующей формуле:

$$e(t) = SP(t) - PV(t),$$

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

Напишем передаточные функции для звеньев ПИД регулятора:

- Усилительное звено: $W_1(p) = K_P$
- Интегрирующее звено: $W_2(p) = \frac{K_I}{p}$
- Дифференцирующее звено: $W_3(p) = \frac{K_D * Tp}{Tp + 1}$

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

$$W(p) = K_P + \frac{K_I}{p} + \frac{K_D * Tp}{Tp + 1}$$

4. 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

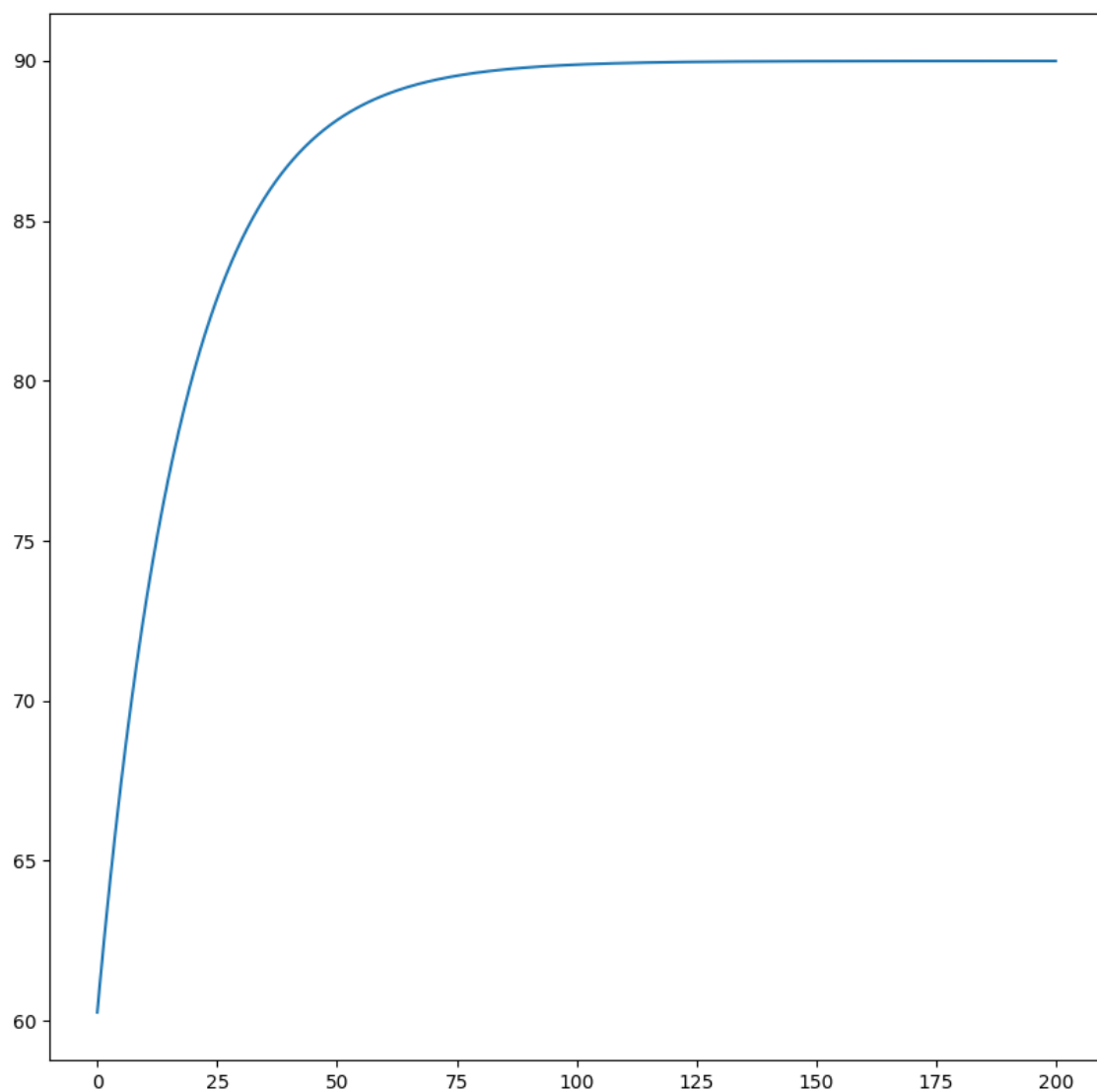
    t = np.arange(0, 200, dt)

    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.