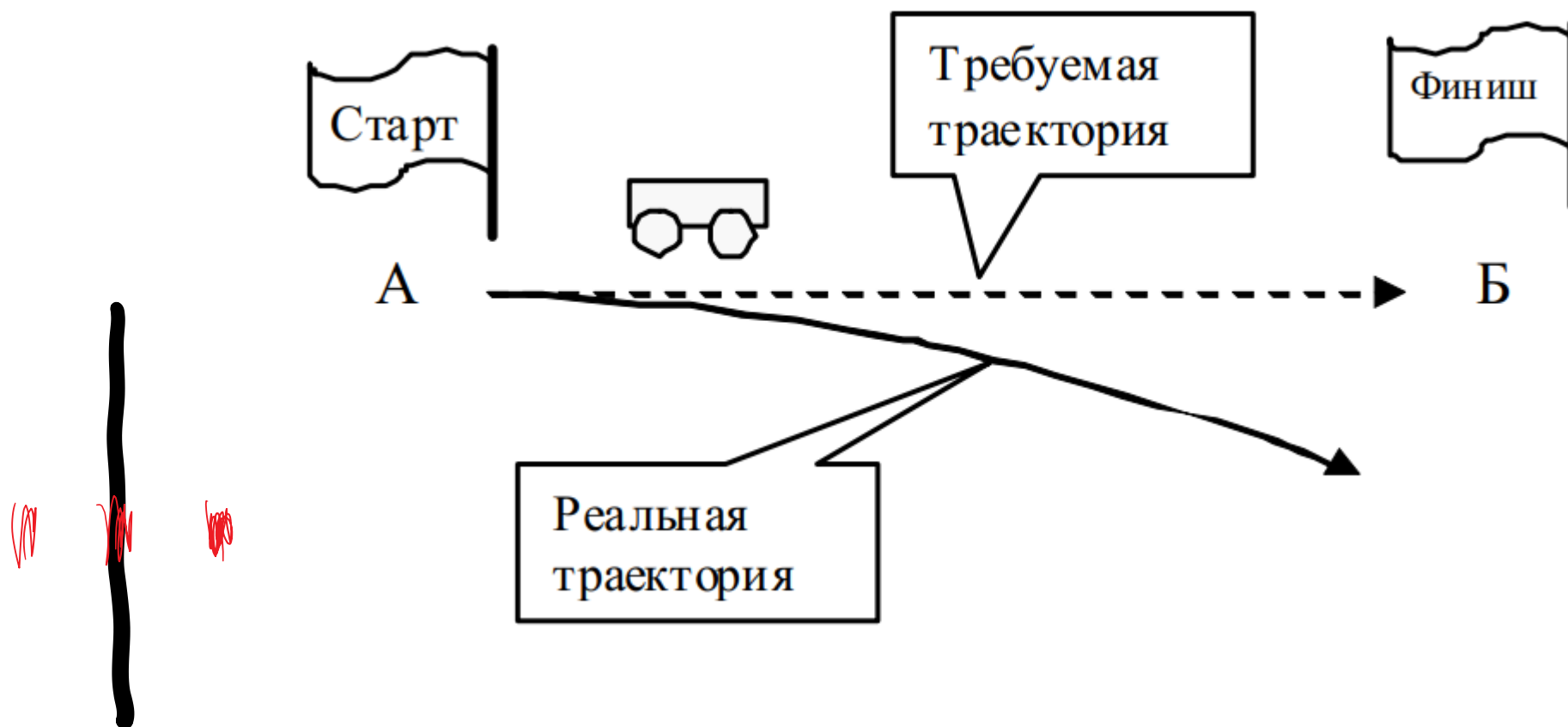


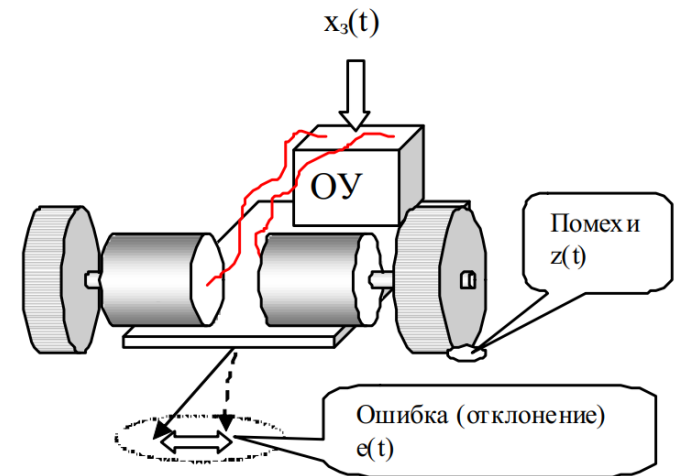
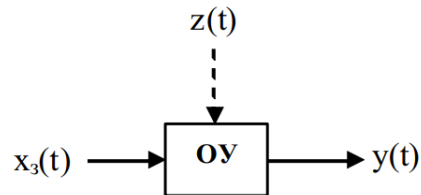
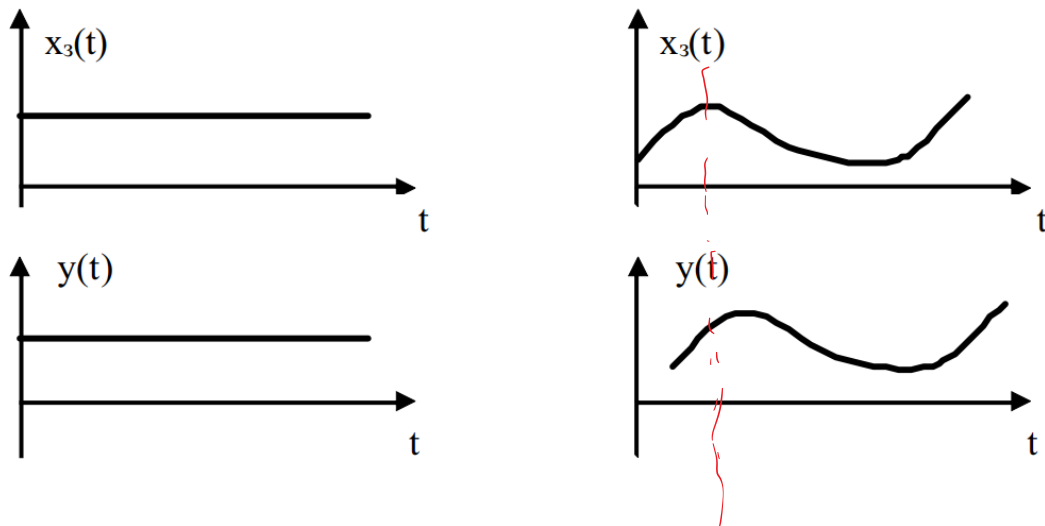
ПИД-регулятор

Задача следования по траектории

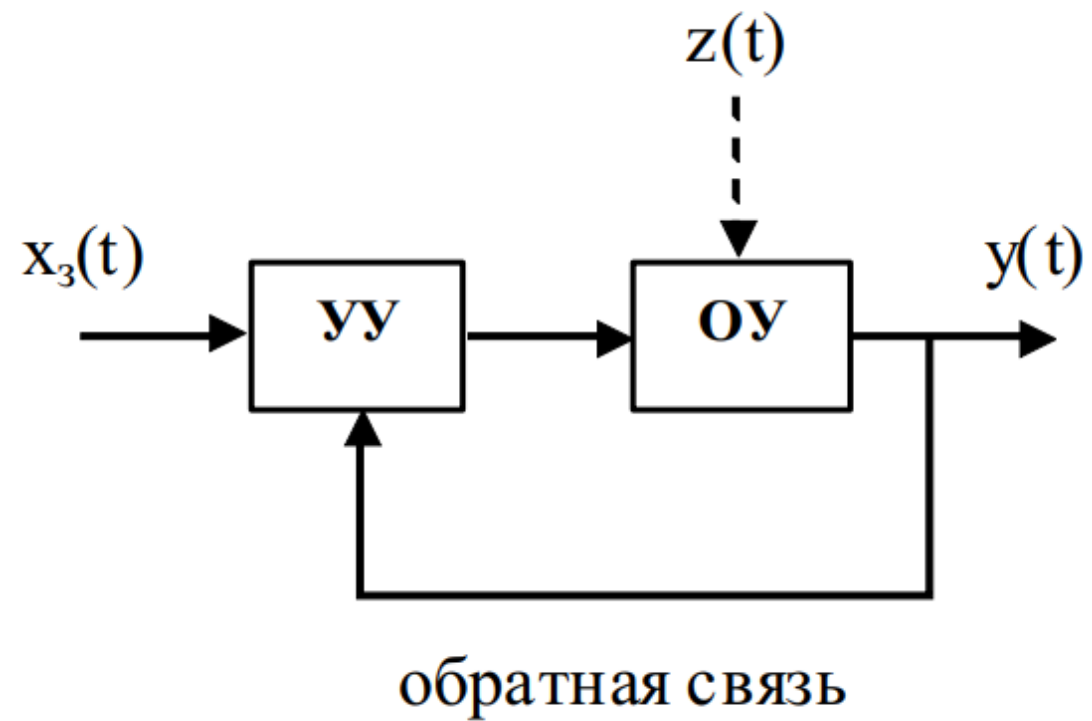
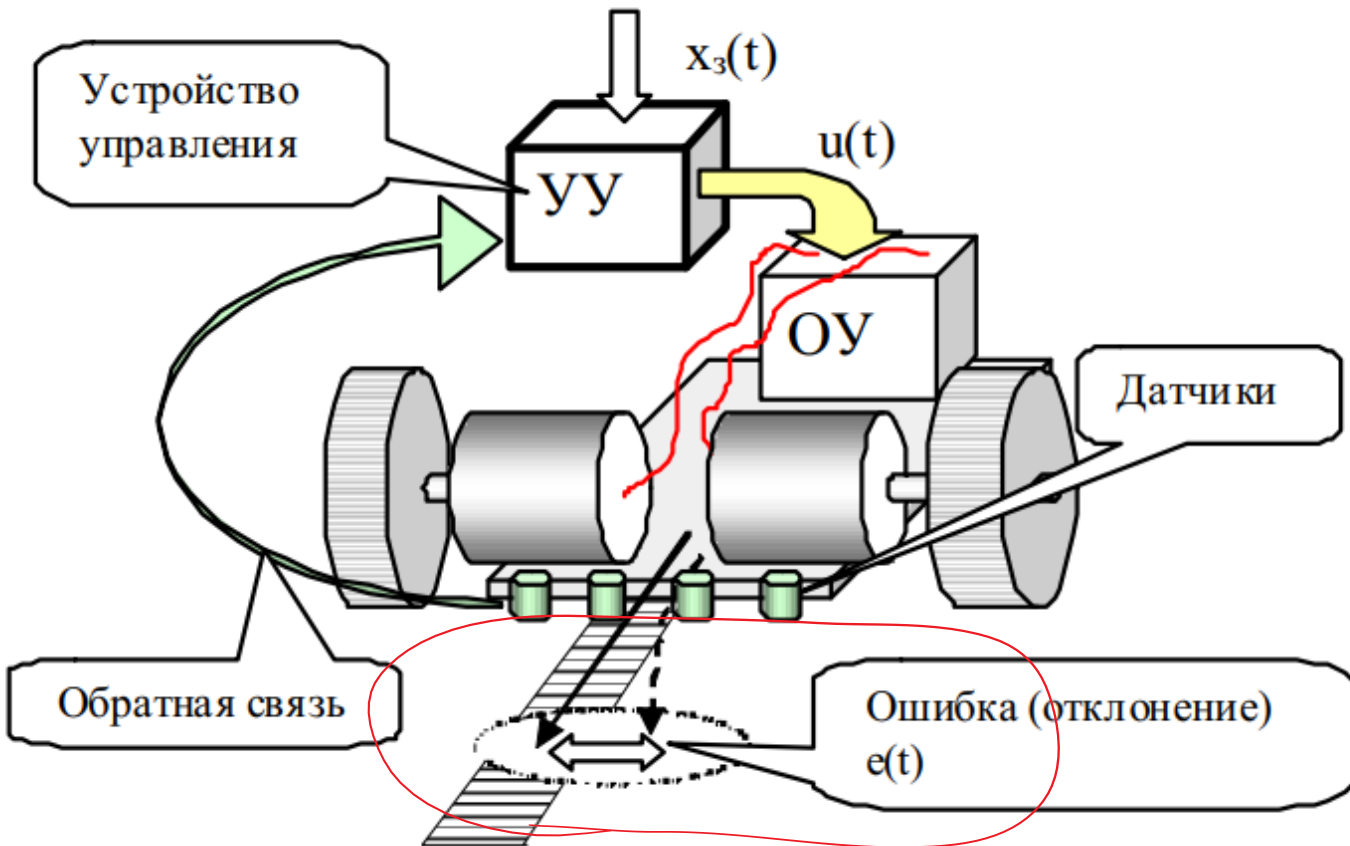


Терминология в контексте задачи

- Регулятор – система, управляющая движением тележки
- Объект управления – блок, управляющий напряжением, подаваемым на двигатели, двигатели и вся ходовая часть
- Задающее воздействие - сигнал, определяющий то, что должно быть на выходе объекта управления $x_3(t)$, $y(t)$ – реальный выходной сигнал

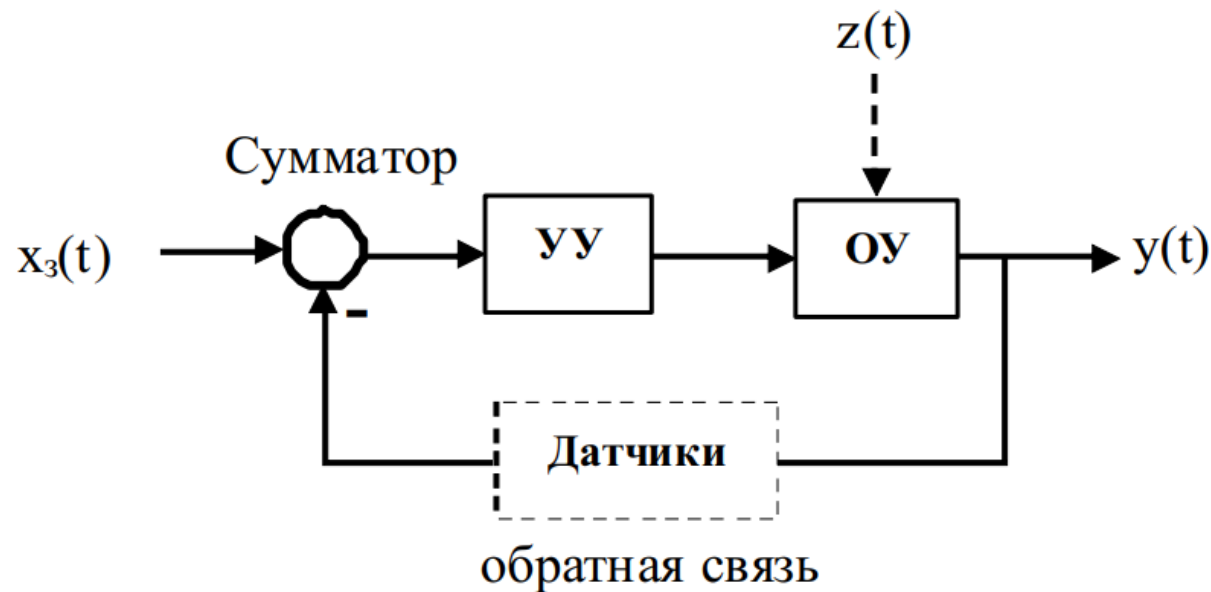


Устройство управления



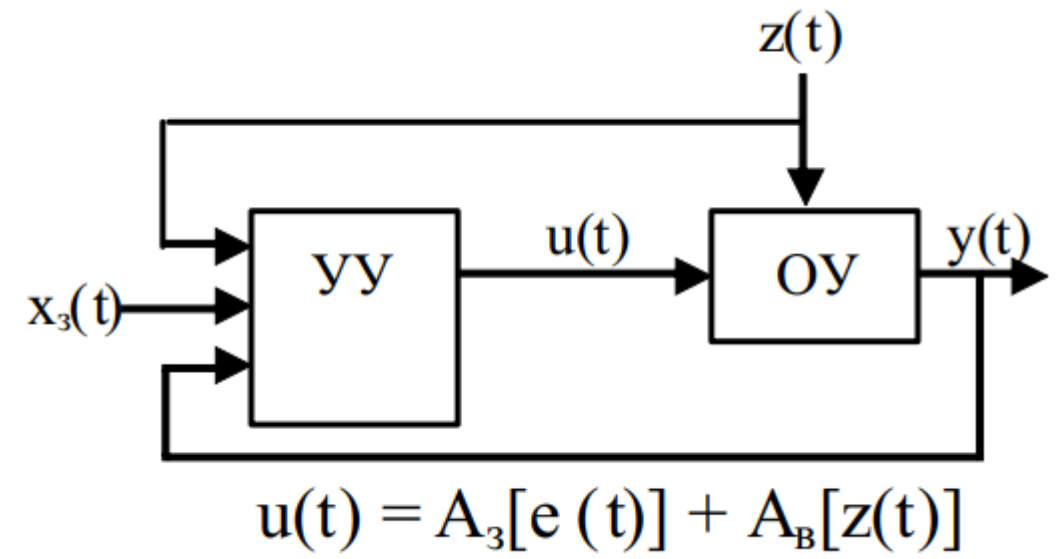
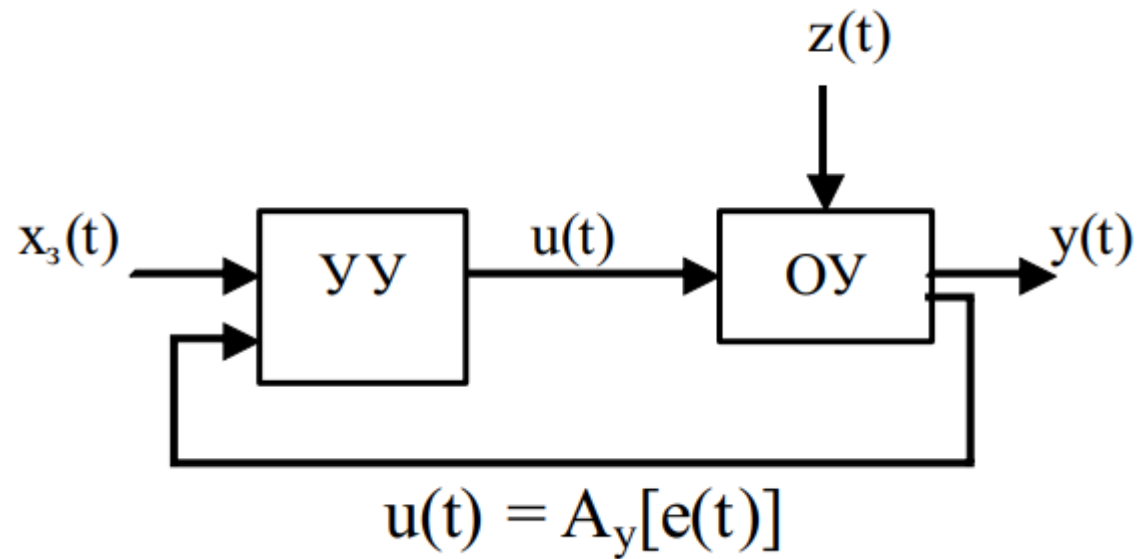
Обратная связь

- это способ учета ошибок в управлении. Осуществляется с помощью датчиков, которые эти ошибки в управлении измеряют.



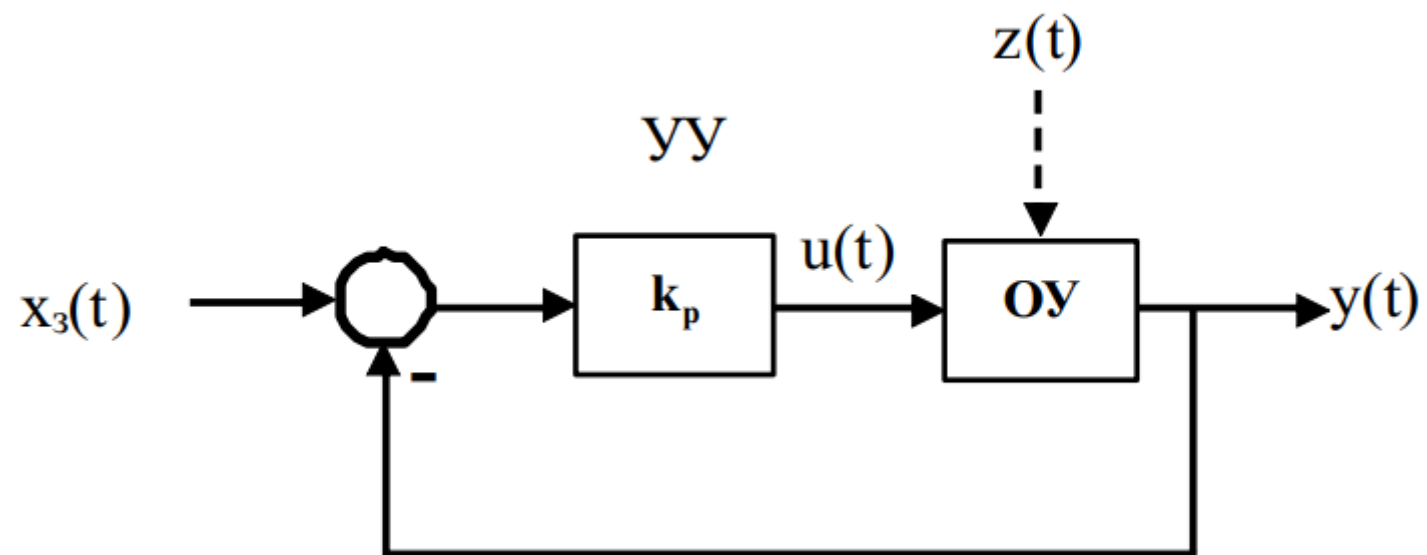
Система с отрицательной обратной связью

Примеры

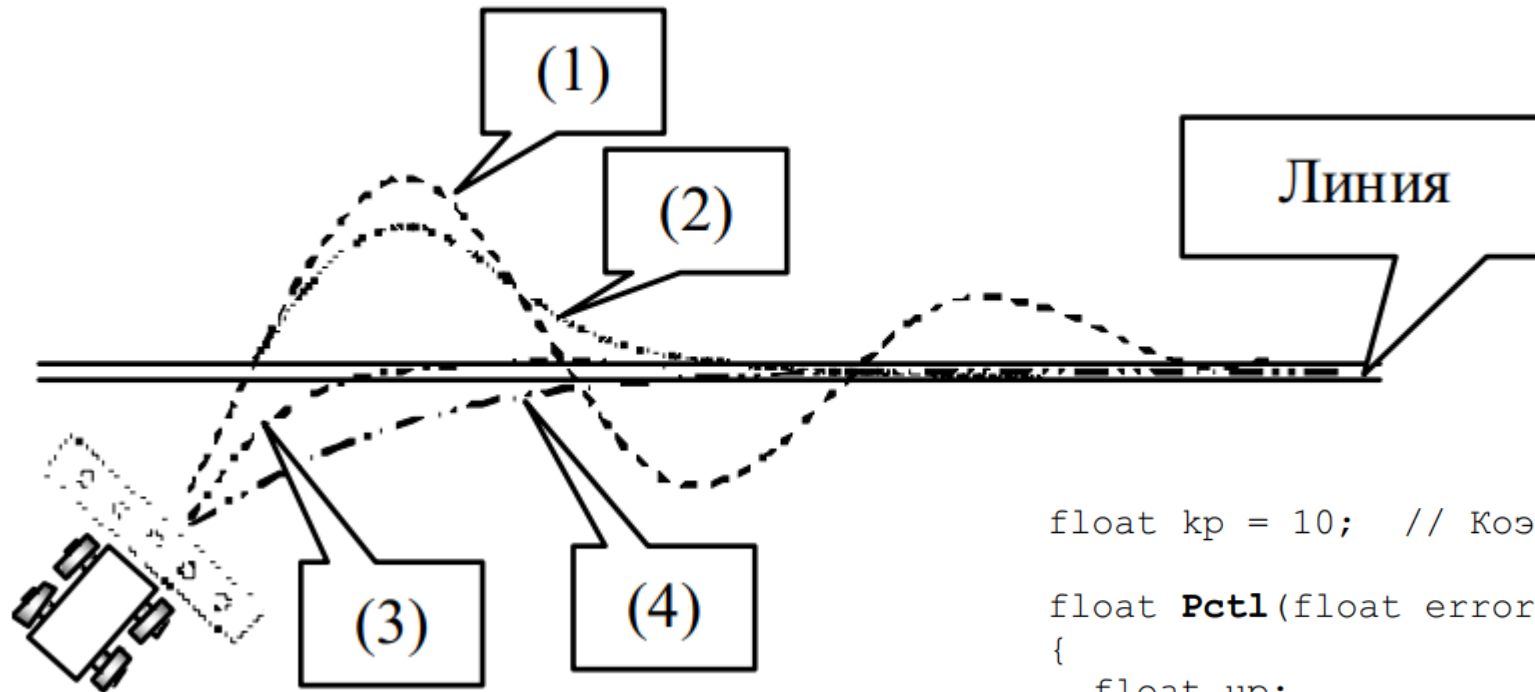


$U(t)$ – сигнал от устройства управления

Пропорциональный регулятор



Качество управления

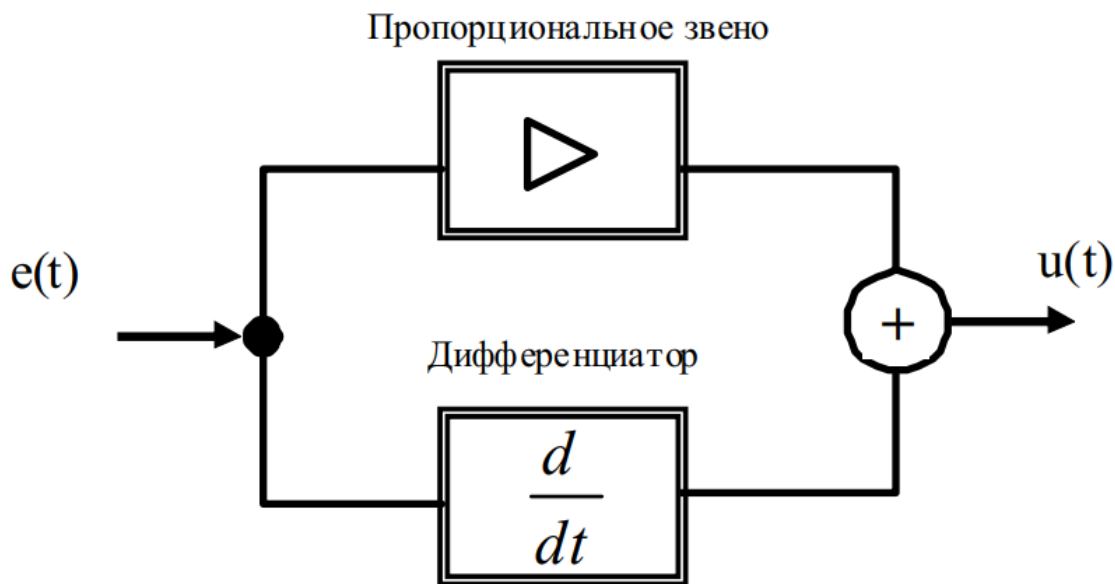


```
float kp = 10; // Коэффициент пропорционального звена  
  
float Pctl(float error)  
{  
    float up;  
    up = kp*error;  
    return up;  
}
```

1, 2 - колебательность, перерегулирование и 4 - большое время переходного процесса

Пропорционально-дифференциальный регулятор

$$u_d(t) = k_d^*(y(t) - y(t-1))$$



```
float kp = 10;    // Коэффициент пропорционального звена
float kd = 1;     // Коэффициент дифференциального звена
float old y = 0;   // Предыдущее значение сигнала
```

```
float PDct1(float error, float y)
{ float up, ud;
```

```
// Пропорциональная компонента
ur = kp*error;
```

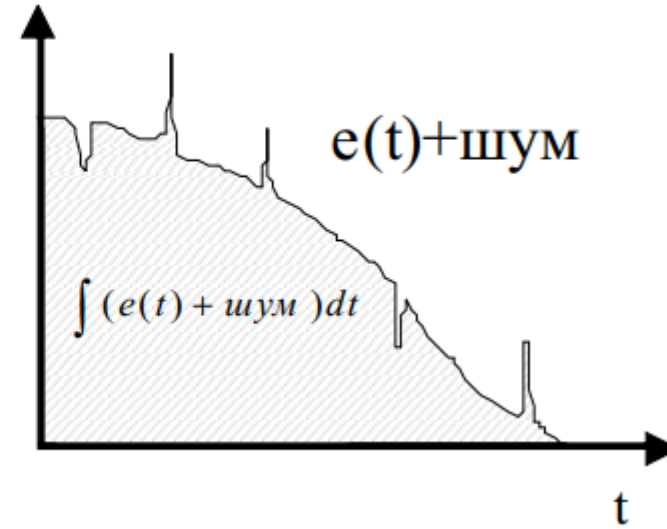
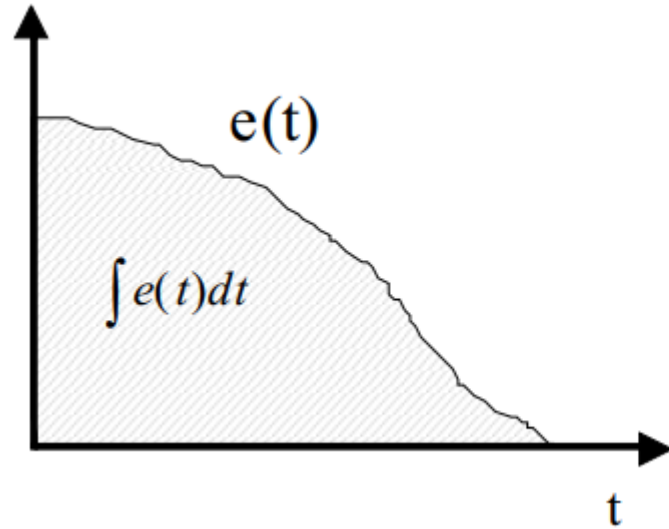
```
// Дифференциальная компонента
ud = kd*(y-old_y);
old_y = y;
```

```
return up+ud;
```

1

Реагирует на скорость изменения величины отклонения, но чувствителен к шумам (необходимо ставить фильтр низких частот) и к частоте сбора данных

Интегральная составляющая



$$u_i(t) = k_i * \int e(t)dt \approx k_i * \sum e(t)\Delta t$$

Пропорционально-интегральный закон управления

$$u_{pi}(t) = u_p(t) + u_i(t) = k_p * e(t) + k_i * \int e(t)dt \approx k_p * e(t) + k_i * \sum e(t)\Delta t$$

ПИ-регулятор

```
float kp = 10;    // Коэффициент пропорционального звена
float ki = 0.001; // Коэффициент интегрального звена

#define iMin -0.2 // Минимальное значение интегратора
#define iMax 0.2  // Максимальное значение интегратора

float iSum = 0;   // Сумма ошибок (значение, накопленное в интеграторе)

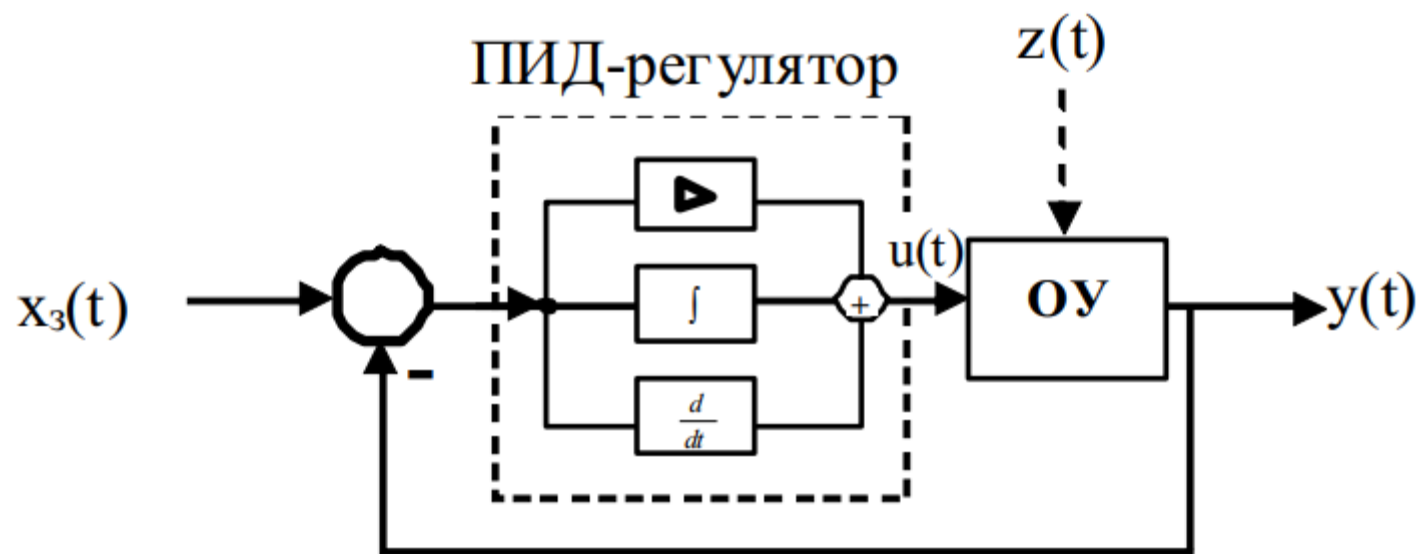
float Pictl(float error)
{
    float up, ui;

    // Пропорциональная компонента
    up = kp*error;

    // Интегральная компонента
    iSum = iSum+error; // Накапливаем (суммируем)
    if(iSum<iMin) iSum = iMin; // Проверяем граничные значение
    if(iSum>iMax) iSum = iMax;
    ui = ki*iSum;

    return up+ui;
}
```

ПИД-регулятор



$$u_{pid}(t) = u_p(t) + u_i(t) + u_d(t) = k_p * e(t) + k_i * \int e(t) dt + k_d * (y(t) - y(t-1))$$