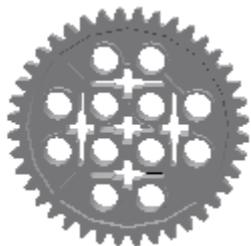

Алгоритмы управления мобильным LEGO-роботом. Простейшие регуляторы



С.А.Филиппов,
Санкт-Петербургский Физико-математический лицей №239

А.Л.Фрадков,
Санкт-Петербургский государственный университет,
Институт проблем машиноведения РАН

Сотрудничество с ВУЗами

- **Киберфизическая лаборатория – совместный проект**
- Институт проблем машиноведения Российской академии наук (ИПМаш РАН)
- СПбГУ, Мат-Мех факультет
 - Кафедра теоретической кибернетики
- Физико-математический лицей №239
- СПб НИУ ИТМО, факультет компьютерных технологий и управления



План занятия

- Задачи регулирования
- Релейный регулятор
- П-регулятор
- Управление двигателем
- Задачи слежения
- Движение по черной линии
- ПД-регулятор: движение вдоль стены



Задачи регулирования

- Под **регулятором** будем понимать **устройство**, состоящее из чувствительного элемента (**датчика**), исполнительного элемента (**мотора**) и закона управления (**алгоритма**).
- В задачах регулирования характерно **циклическое управление** с высокой частотой
- Примеры задач:
 - Управление положением серводвигателя
 - Контроль расстояния до объекта
 - Следование по заданной траектории
 - Следование вдоль стены
 - Балансирование
 - Полети др.



Типы регуляторов

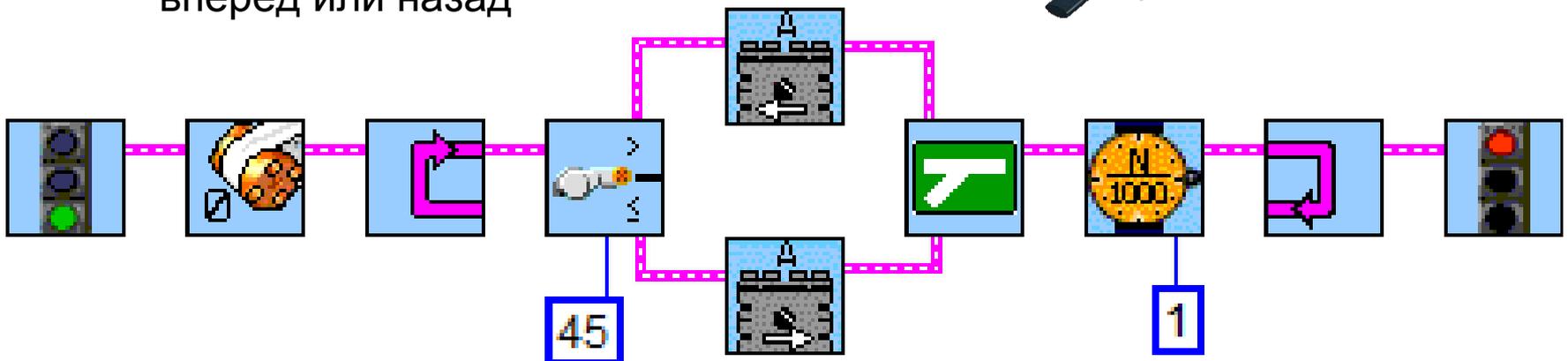
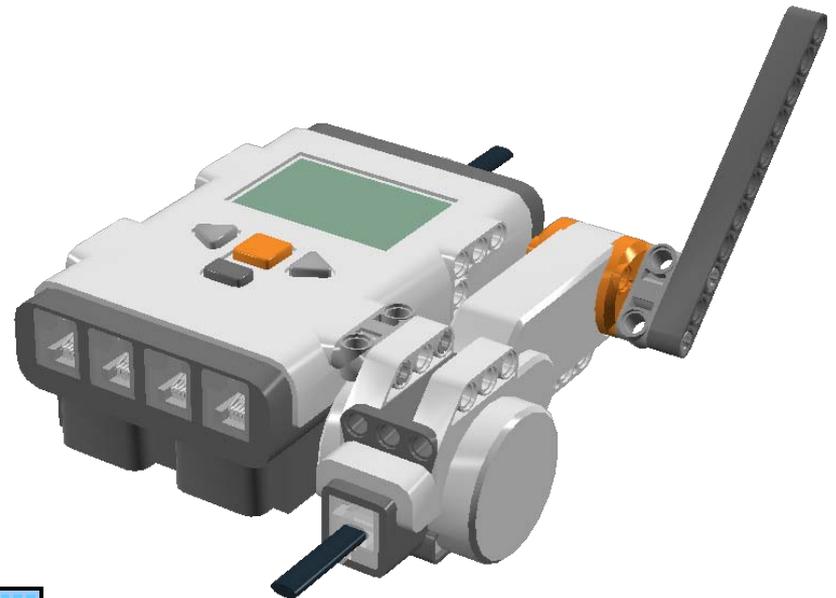
- Релейный (двухпозиционный)
 - Вырабатывает управляющее воздействие двух видов
 - Пропорциональный
 - Управление по отклонению системы от заданного состояния
 - Дифференциальный
 - Управление по скорости отклонения
 - Интегральный
 - Управление по накопленной ошибке
- и др.



Релейный регулятор: управление мотором

Энкодер – датчик оборотов,
встроенный в двигатель NXT

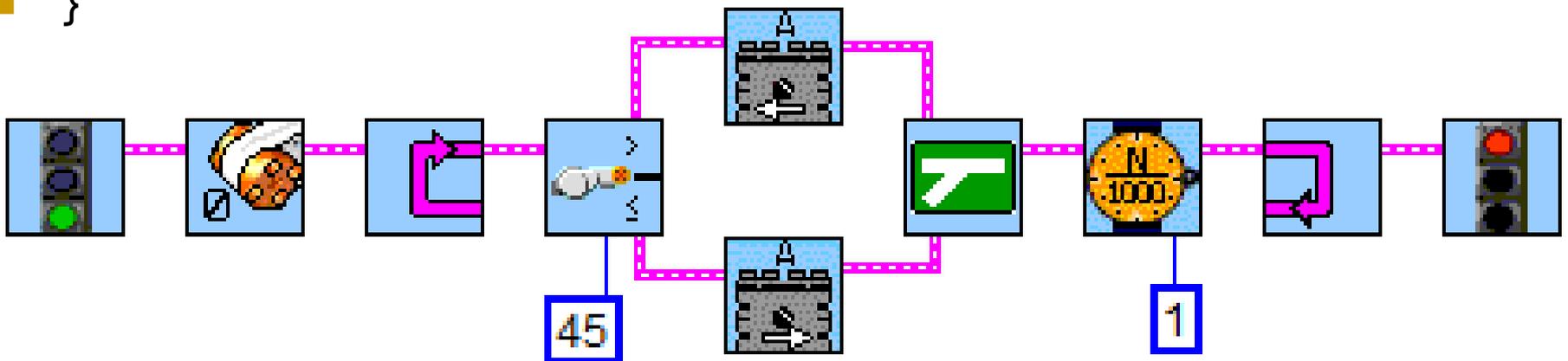
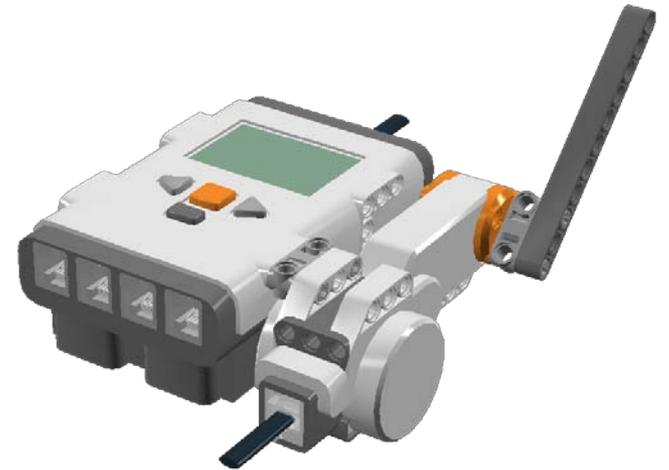
- Задача
 - Установить мотор в заданное положение 45°
 - Удерживать мотор в заданном положении
- Решение
 - Обнулить исходное положение энкодера
 - 1000 раз в секунду проверять положение энкодера и
 - оказывать управляющее воздействие на моторы вперед или назад



Релейный регулятор: управление мотором

```
task main()  
{  
  int alpha=45;  
  nMotorEncoder[motorA]=0;  
  while(true){  
    if(nMotorEncoder[motorA]>alpha)  
      motor[motorA]=-100;  
    else  
      motor[motorA]=100;  
    wait1Msec(1);  
  }  
}
```

Энкодер – датчик оборотов,
встроенный в двигатель NXT



Пропорциональный регулятор

В задачах автоматического регулирования **управляющее воздействие** $u(t)$ обычно является функцией динамической ошибки – отклонения $e(t)$ регулируемой величины $x(t)$ от ее заданного значения $x_0(t)$:

$$e(t) = x_0(t) - x(t)$$

Пропорциональный регулятор – это устройство, оказывающее управляющее воздействие на объект пропорционально его отклонению от заданного состояния.

$$u_0(t) = ke$$

Здесь k – это коэффициент усиления регулятора.

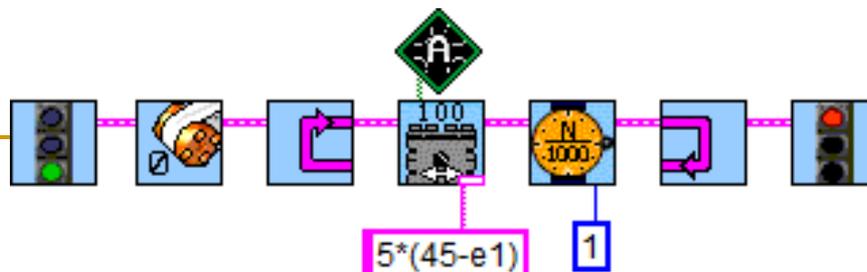


Пропорциональный регулятор: управление мотором

- Пусть $e1$ – показания датчика оборотов на моторе А. Тогда управление мощностью мотора задается формулой:

$45-e1$ – отклонение энкодера от положения 45°

- **$Motor[MotorA]=k*(45-e1)$,**
- где k – усиливающий коэффициент, например 5.
- **$nMotorEncoder[motorB]=0$;**
- **$while(true) \{$**
- **$e1=nMotorEncoder[motorB]$;**
- **$Motor[MotorA]=5*(45-e1)$,**
- **$wait1Msec(1)$;**
- **$\}$**



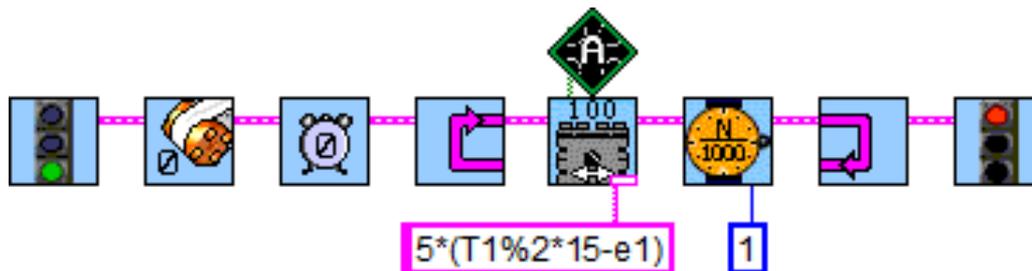
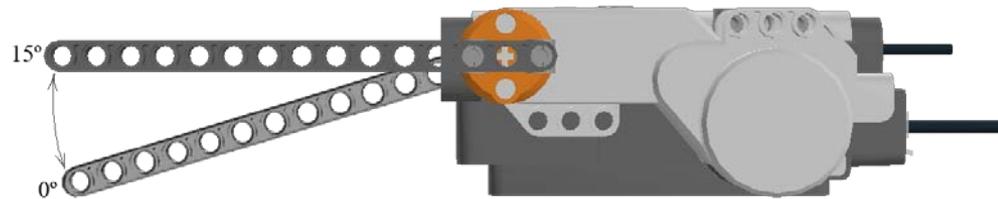
Манипуляторы – применение точного позиционирования двигателей



Робот-барабанщик: изменение положения двигателя

- Использование целочисленного деления на 2 для получения из показаний таймера T1 ряда чисел 0,1,0,1...
- Изменение положения двигателя 10 раз в секунду по формуле $T1\%2*15$ от 0° до 15° .
- Перед стартом опустить палочку до уровня стола

В NXT имеется 4 встроенных таймера: T1, T2, T3, T4, такт по умолчанию 0,1с



```
nMotorEncoder[motorB]=0;
ClearTimer(T1);
while(true){
    e1=nMotorEncoder[motorB];
    motor[motorA]=
        5*(timer100[T1]%2*15-e1);
    wait1Msec(1);
}
```

Попробуйте применить формулу $T1\%5\%2*15$ и придумать свою!

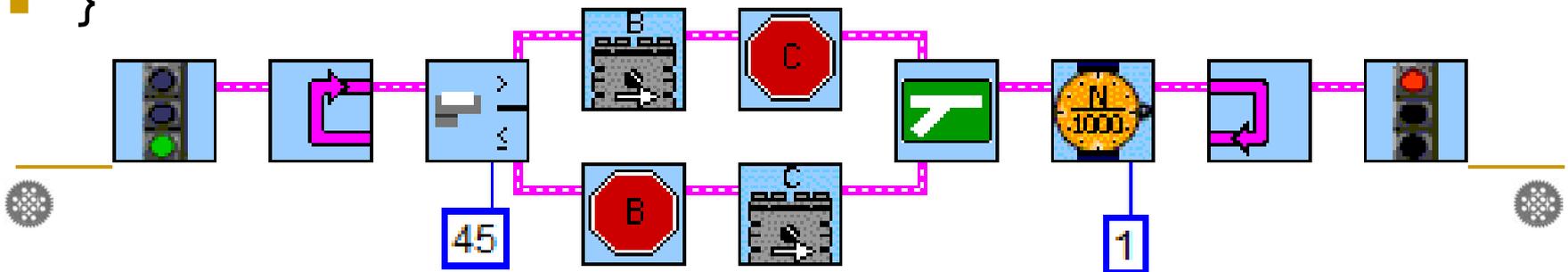
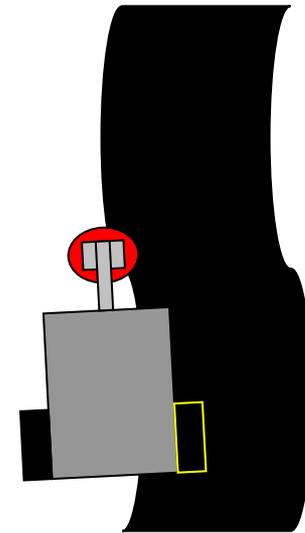
Использование пропорционального регулятора в работе-барабанщике



Релейный регулятор: движение вдоль границы черного и белого

*Определяем показания датчика освещенности на границе и записываем в переменную **grey***

```
■ int grey=45;  
■ while (true) {  
■   if (SensorValue[S1]>grey) {  
■     motor[motorB]=100;  
■     motor[motorC]=0;  
■   }  
■   else{  
■     Motor[MotorB]=0;  
■     Motor[MotorC]=100;  
■   }  
■   wait1Msec(1);  
■ }
```

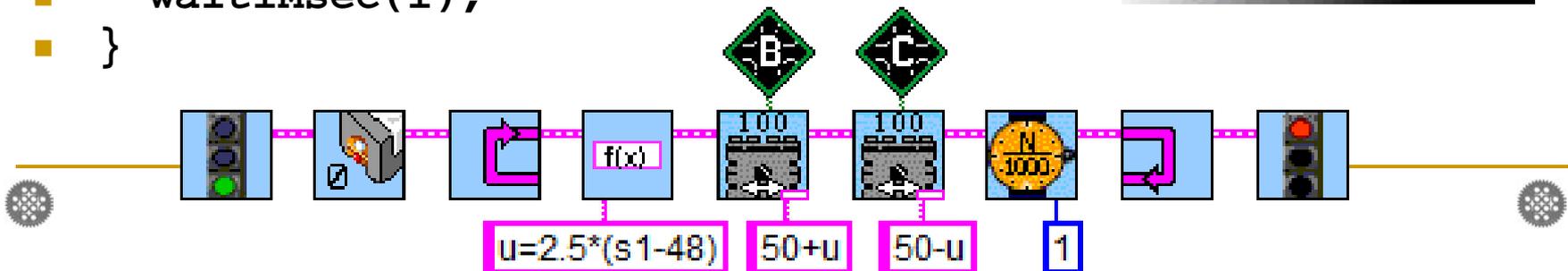
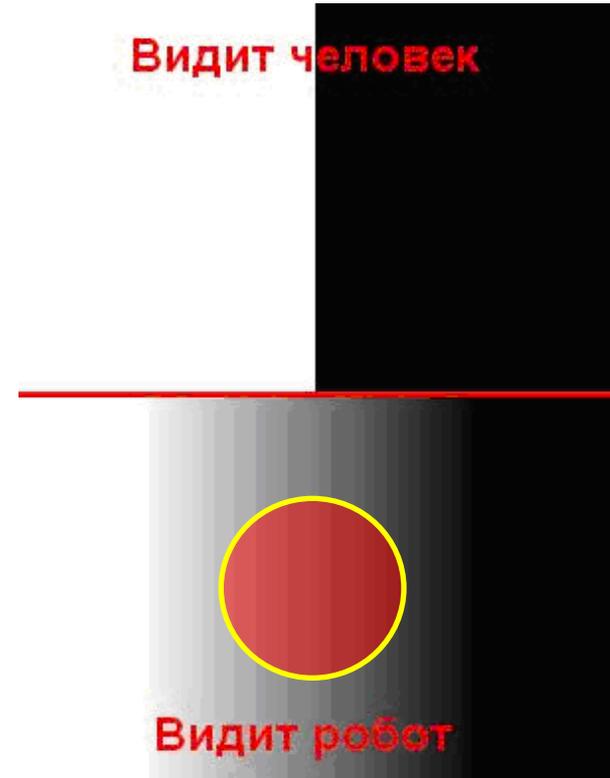


Пропорциональный регулятор:

ДВИЖЕНИЕ ПО ЛИНИИ

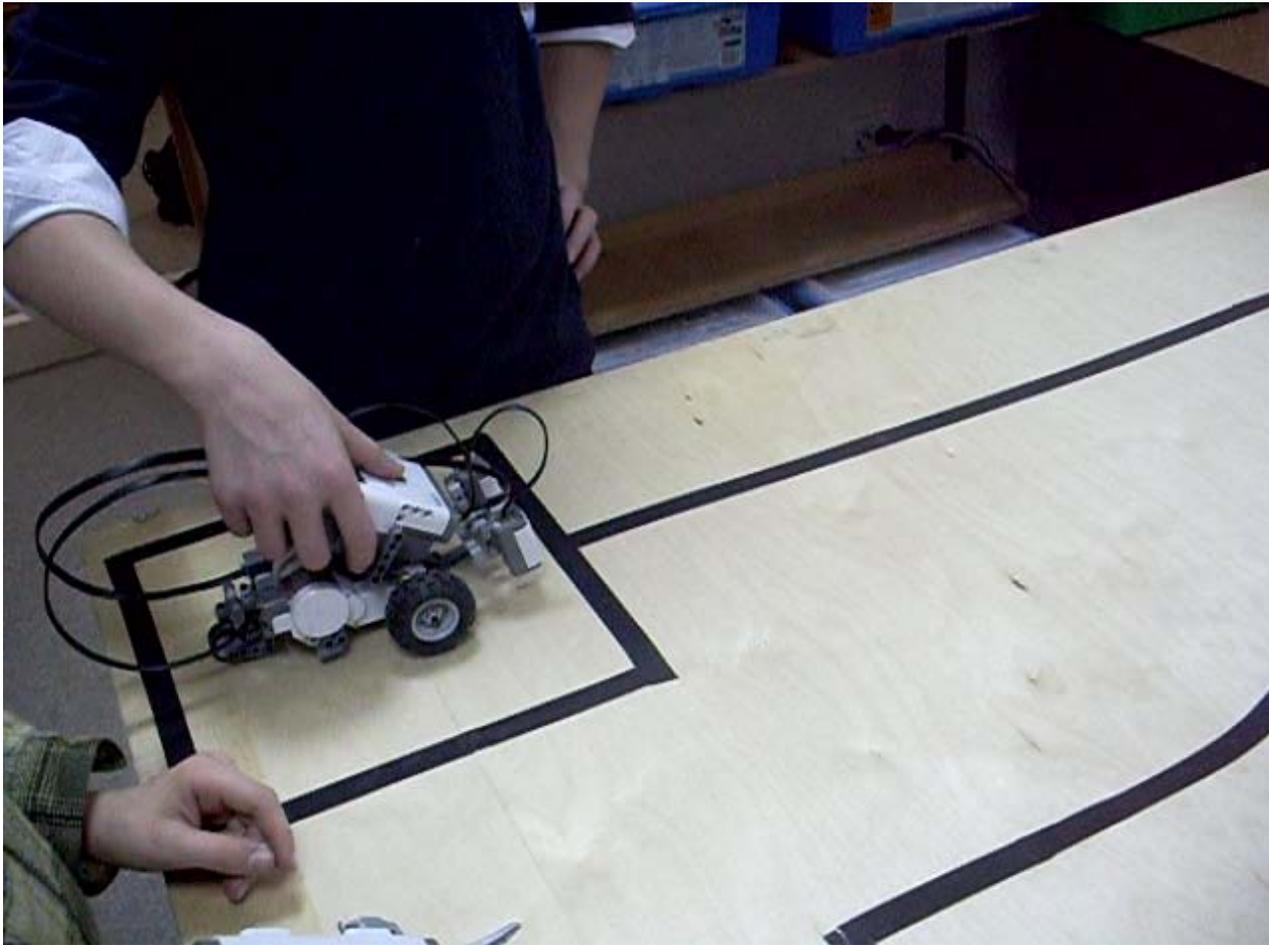
- Также как и в релейном регуляторе, необходимо определить среднее значение grey между черным и белым. Это будет то состояние датчика освещенности $s1$, к которому должна стремиться система.
- Управляющее воздействие u сочетается с базовой скоростью **50%**

- `while(true)`
- `{`
- `u=k*(s1-grey);`
- `motor[motorB]=50+u;`
- `motor[motorC]=50-u;`
- `wait1Msec(1);`
- `}`



Пропорциональный регулятор:

ДВИЖЕНИЕ ПО ЛИНИИ С ДВУМЯ ДАТЧИКАМИ

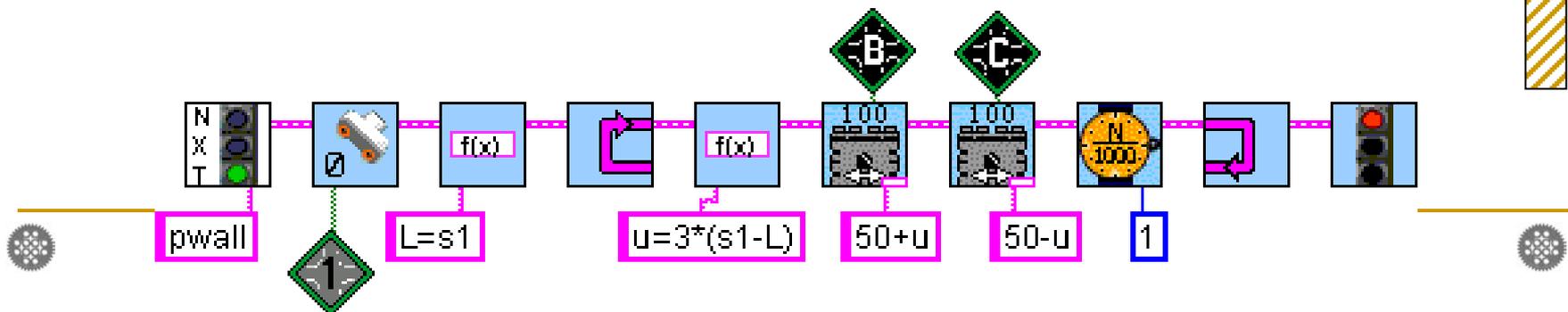
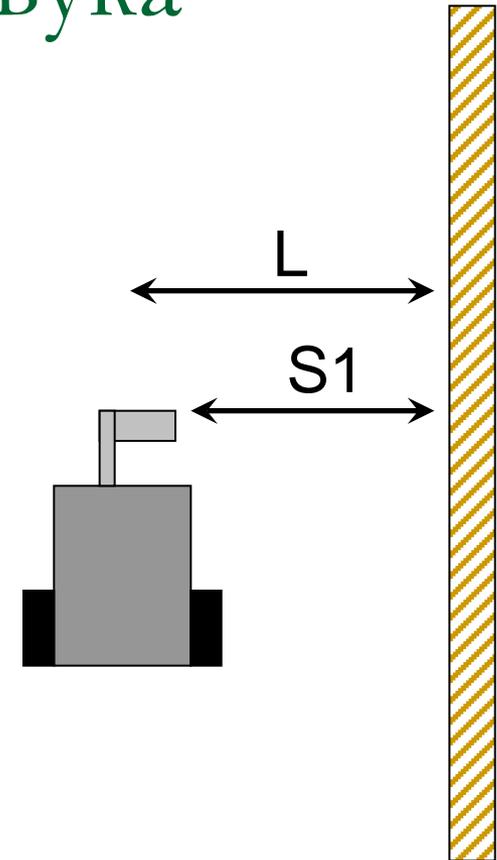


Движение вдоль стенки

С помощью датчика ультразвука

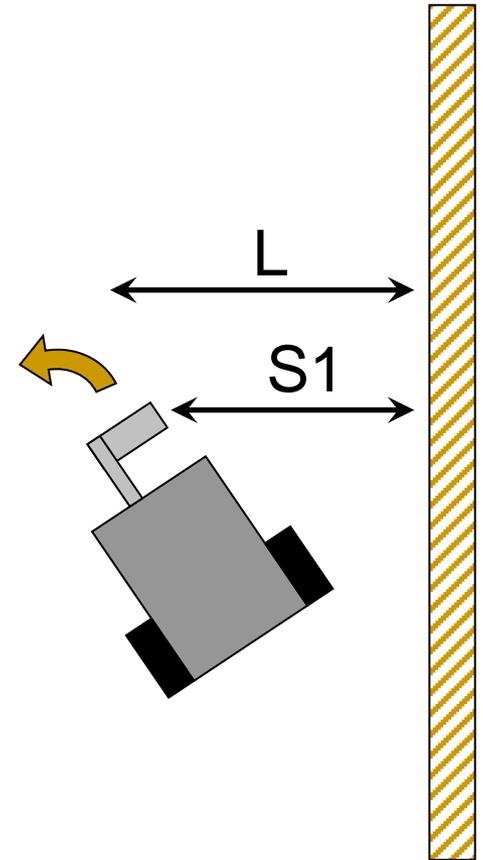
- Движение вдоль стенки на заданном расстоянии L
- $S1$ – текущие показания датчика
- Используется пропорциональный регулятор

```
L=SensorValue[S1];  
while(true)  
{  
    u=k*(SensorValue[S1]-L);  
    Motor[MotorB]=50+u;  
    Motor[MotorC]=50-u;  
    wait1msec(1);  
}
```



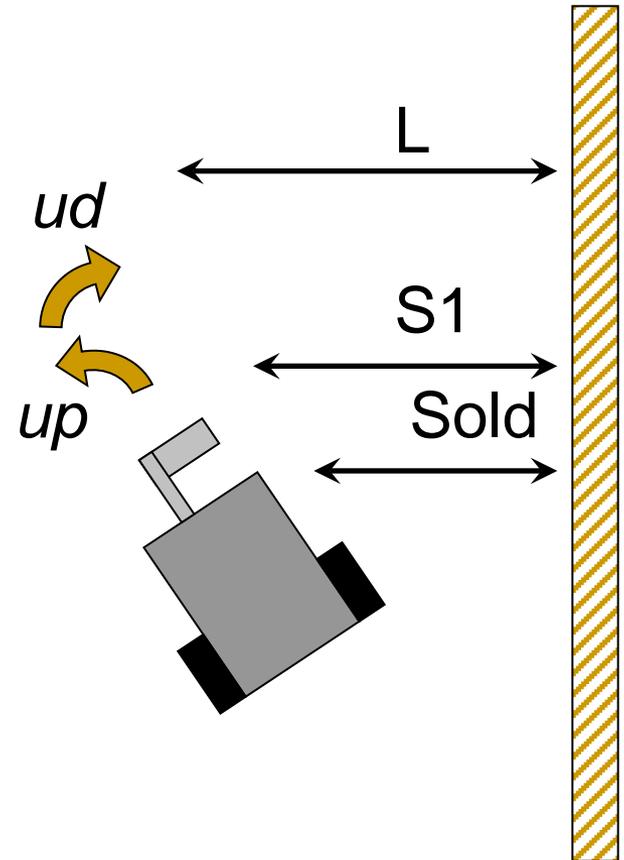
Движение вдоль стенки: недостатки П-регулятора

- Предложенное управление не обладает свойством асимптотической устойчивости
- Работает только на малых углах отклонения
- Допустимы ситуации, при которых ультразвуковой датчик теряет контакт со стенкой
- $u_p = k * (s_1 - L)$
- Вывод: требуется дифференциальная составляющая



Движение вдоль стенки с помощью ПД-регулятора

- $u_d = k * (s_1 - s_{old}) / \Delta t,$
- где s_1 – текущее расстояние до стенки, s_{old} – расстояние на предыдущем шаге.
- Дифференциальная составляющая компенсирует пропорциональную
- $u = u_p + u_d$
- $u = k_1 * (s_1 - L) + k_2 * (s_1 - s_{old})$

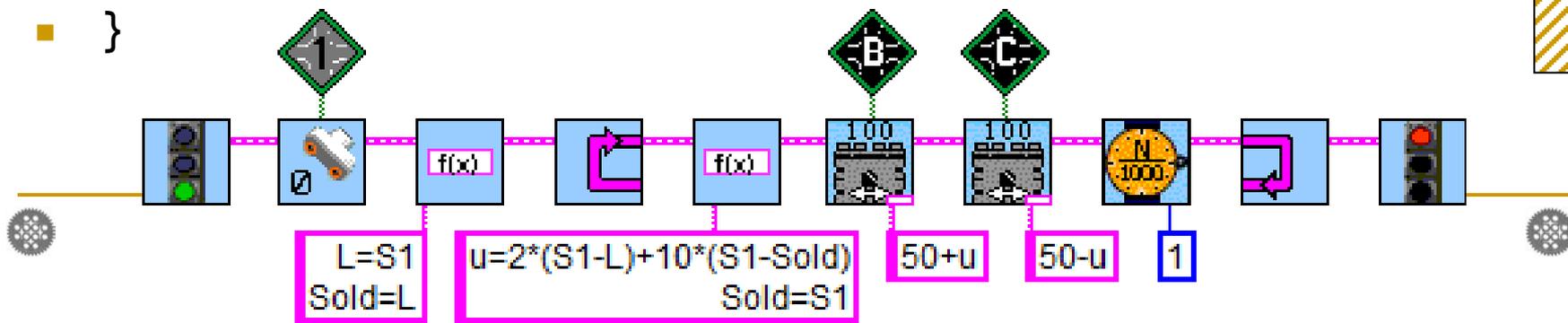
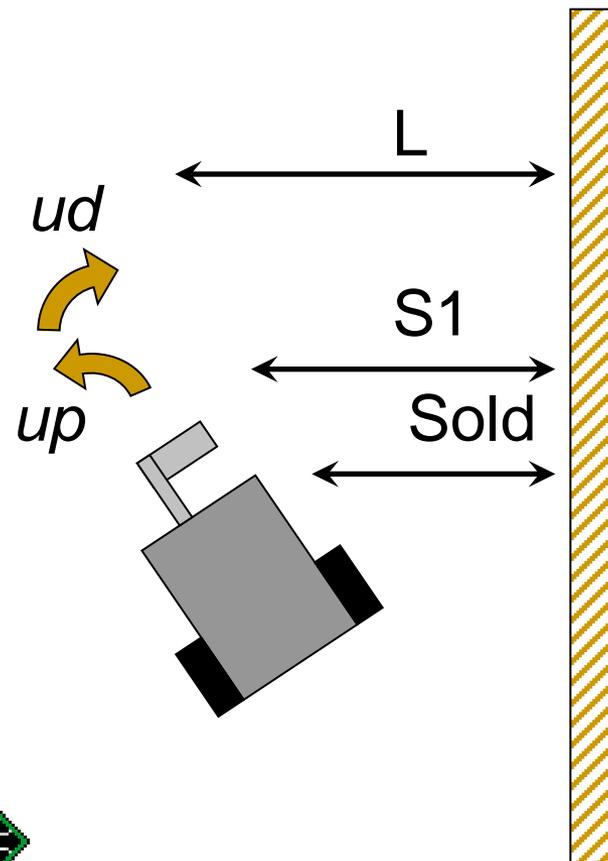


Можно показать, что для устойчивого достижения цели коэффициент k_2 при дифференциальной составляющей должен превышать k_1



Движение вдоль стены с помощью ПД-регулятора

- В качестве первоначального значения Sold принимаем текущее расстояние от стены, которое будет поддерживаться
- `Sold=L=S1;`
- `while(true)`
- {
- `u = k1*(S1-L) + k2*(S1-Sold);`
- `motor[motorB]=50+u;`
- `motor[motorC]=50-u;`
- `Sold=S1;`
- `wait1Msec(1);`
- }



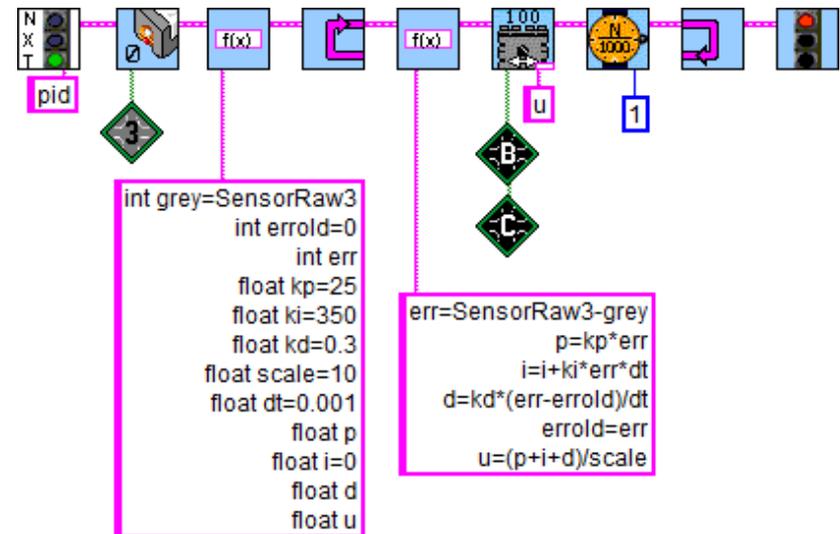
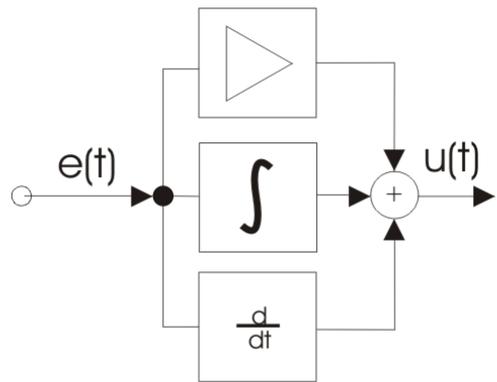
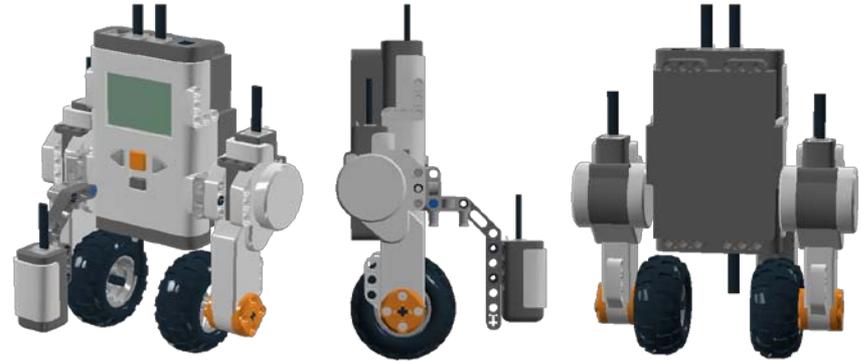
Объезд предметов с помощью ПД-регулятора



Балансирующий робот

ПИД-регулятор

- Балансирующий робот на базе датчика освещенности
- Классический ПИД-регулятор
- Интегральная составляющая как сумма длительностей ошибок
- Весь алгоритм в виде блок-схемы с выделенным математическим блоком



Стабилизация сигвея - увлекательная задача для семиклассников





Результаты работы

- **Основы** элементарной теории управления оказались доступны учащимся **5-7 классов**
- Вместо **дифференциальных уравнений** были использованы **разностные**, хорошо соответствующие дискретному характеру взаимодействия объекта и регулятора при компьютерном управлении
- Предложенные способы реализации мобильных роботов способствовали росту интереса учащихся к робототехнике
- Была усовершенствована методика преподавания робототехники в школе



Благодарю за внимание!

Сергей Александрович Филиппов
Физико-Математический лицей № 239
Санкт-Петербург
safilippov@gmail.com

