

ОСНОВЫ РОБОТОТЕХНИКИ НА БАЗЕ КОНСТРУКТОРА LEGO MINDSTORMS NXT.

ЗАНЯТИЕ 6. ОРИЕНТАЦИЯ НА МЕСТНОСТИ: ОБЪЕЗЖАЕМ СТЕНЫ

Когда-нибудь в недалеком будущем роботы выйдут в коридоры, на улицы и станут нашими соседями не только по лабораториям. Движение по дорожной разметке (черной линии на белом фоне) было рассмотрено в предыдущей статье. Теперь научим робота объезжать стены.

ДВИЖЕНИЕ ВДОЛЬ СТЕНКИ

Решим такую задачу. Робот должен двигаться вдоль стенки на заданном расстоянии L (рис. 1), которое определяется в момент старта. Предположим, что левое колесо робота управляется мотором В, правое — мотором С, а датчик расстояния, подключенный к порту 1, закреплен несколько впереди корпуса тележки (это важно!) и направлен на стенку справа по ходу движения (рис. 5.7).

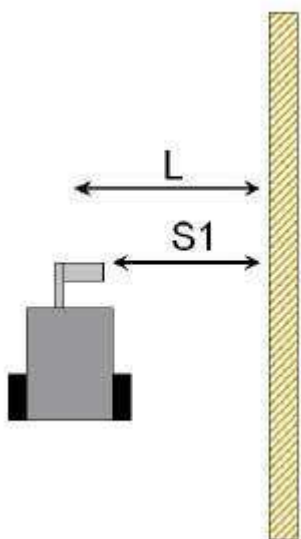


Рис. 1. Задача движения вдоль стенки на расстоянии

Расстояние до стенки в настоящий момент времени, которое показывает датчик, обозначим $S1$. Измеряется оно в сантиметрах. Моторы двигаются со средней скоростью 50% от максимума, но при отклонении от заданного курса на них осуществляется управляющее воздействие u . Обозначим это следующим образом:

Motor[MotorB]=50+ u ;

Motor[MotorC]=50- u ;

Осталось определить, чему будет равно управляющее воздействие. Это нетрудно сделать с помощью пропорционального регулятора:

$$u = k \cdot (S1 - L).$$

Таким образом, при $S1 = L$ робот не меняет курса и едет прямо. В случае отклонения его курс корректируется. Здесь k — это некоторый усиливающий коэффициент, определяющий воздействие регулятора на систему. Для робота NXT средних размеров коэффициент k может колебаться от 1 до 10, в зависимости от многих факторов. Предлагаем подобрать его самостоятельно (рис. 3).

Та же программа на языке RobotC. Не забудьте объявить датчик ультразвука (Sonar) через меню **Robot / Motor and Sensors / Setup**. Желательно задать датчику имя, которое будет использоваться вместо $S1$.

```
task main() {
```

```
float u, k=3;
```

```
int L=SensorValue[S1];
```

```

while(true) {
u=k*(SensorValue[S1]-L);
motor[motorB]=50+u;
motor[motorC]=50-u; wait1Msec(1);
}
}

```

В данном случае П-регулятор будет эффективно работать только при малых углах отклонения. Кроме того, движение практически всегда будет происходить по волнообразной траектории. Сделать регулирование более точным позволит введение новых принципов, учитывающих отклонение робота от курса.

ПРОПОРЦИОНАЛЬНО ДИФФЕРЕНЦИАЛЬНЫЙ РЕГУЛЯТОР

В некоторых ситуациях П-регулятор может вывести систему из устойчивого состояния (рис. 2а). Например, если робот направлен от стенки, но находится по отношению к ней ближе заданного расстояния, на моторы поступит команда еще больше повернуть от стенки, в результате чего с ней может быть утрачен контакт (датчик расстояния получает отраженный сигнал практически только от перпендикулярной поверхности).

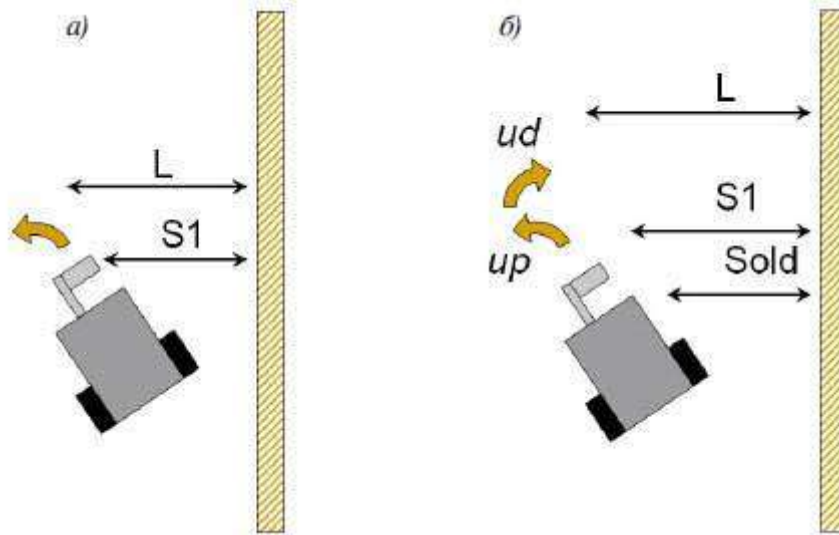


Рис. 2. Проблема пропорционального регулятора .
потеря контакта со стенкой (а). Необходима дифференциальная составляющая (б)

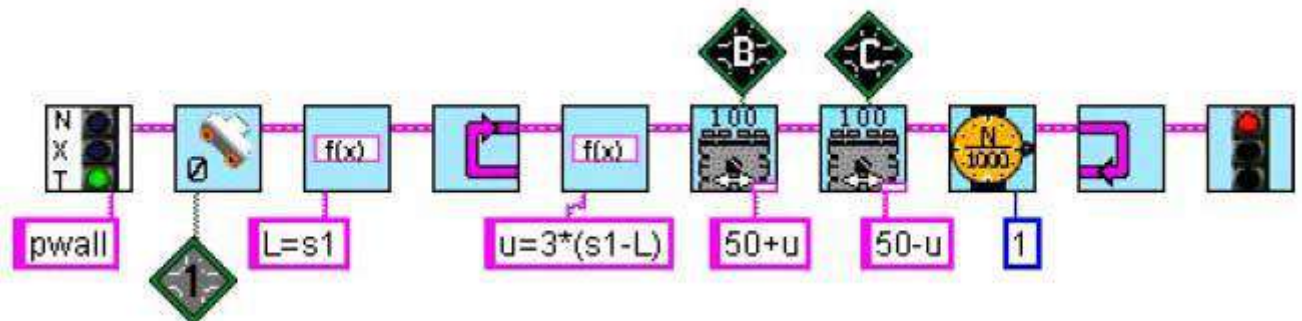


Рис. 3. Алгоритм движения вдоль стенки на основе пропорционального регулятора

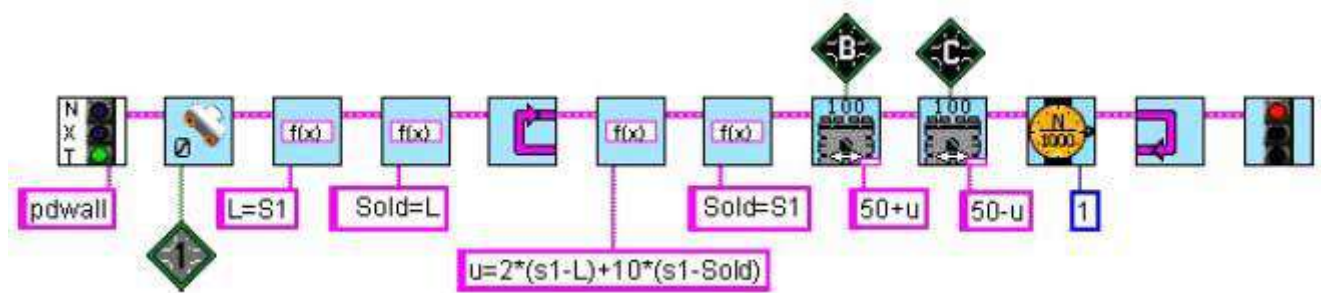


Рис. 4. Алгоритм движения вдоль стены, основанный на ПД-регуляторе. Для защиты от подобных ситуаций добавим в регулятор дифференциальную составляющую, которая будет следить за направлением движения робота (рис. 2б). Иными словами, вектор скорости будет влиять на управляющее воздействие. Известно, что скорость находится как

$$v = \Delta s / \Delta t,$$

где Δs - это изменение расстояния за промежуток времени Δt . Определим дифференциальную составляющую через скорость отклонения робота от заданного положения:

$$ud = k * (S1 - Sold) / \Delta t,$$

где $S1$ - текущее расстояние до стенки, $Sold$ - расстояние на предыдущем шаге. Поскольку замеры производятся через равные промежутки времени, то Δt можно принять за константу, взяв $k2 = k * \Delta t$.

$$ud = k2 * (S1 - Sold)$$

Таким образом, ПД-регулятор описывается формулой из двух слагаемых

$$u = up + ud = k1 * (S1 - L) + k2 * (S1 - Sold).$$

Можно доказать математически, что для устойчивого достижения цели коэффициент $k2$ при дифференциальной составляющей должен превышать $k1$.

Алгоритм движения вдоль стенки на ПД-регуляторе в целом будет выглядеть так:

```
task main() {
float u, k1=2, k2=10;
int Sold, L;
Sold=L=SensorValue[S1];
while(true) {
u= k1*(SensorValue[S1]-L) + k2*(SensorValue[S1]-Sold);
motor[motorB]=50+u;
motor[motorC]=50-u;
Sold=SensorValue[S1];
wait1Msec(1);
}
}
```

КОНСТРУКЦИЯ ДЛЯ ДВИЖЕНИЯ ВДОЛЬ СТЕНКИ

Базовая конструкция тележки описана в статье «Управление мобильным роботом» (№ 3, 2010). Остается только правильно прикрепить датчик расстояния и подключить его на 1 порт (рис. 5 - 7).



Рис. 5. Датчик расстояния слегка выносится вперед



Рис. 6. Ориентацию датчика можно варьировать



Рис. 7. Оптимальный вариант . вертикальное расположение датчика

ДАТЧИК ПОД УГЛОМ

Описанный выше робот сможет объезжать стены только при малых отклонениях от прямой линии. Рассмотрим вариант, при котором на пути движения будут возникать серьезные повороты, вплоть до прямых углов. Потребуется внести модификации и в конструкцию и в программу (рис. 8).

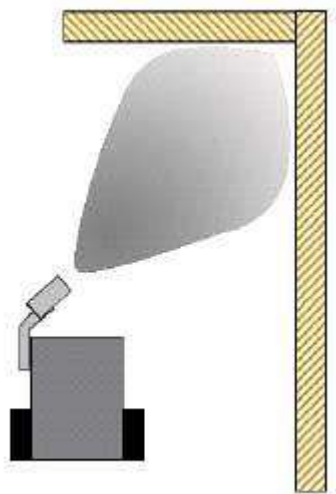


Рис. 8. Датчик расстояния устанавливается под острым углом к направлению движения

Во-первых, робот должен будет смотреть не только направо, но и вперед. Ставить второй датчик довольно затратно. Однако можно воспользоваться эффектом того, что ультразвуковой датчик имеет расширяющуюся область видимости (рис. 8). Это напоминает угловое зрение человека: кое-что он может увидеть краем глаза. Стоит воспользоваться таким свойством и разместить датчик расстояния не перпендикулярно курсу движения, а под острым углом (рис. 9 - 11). Так можно «убить сразу двух зайцев». Во-первых, робот будет видеть препятствия спереди, во-вторых, более стабильно будет придерживаться курса вдоль стены, постоянно находясь на грани видимости. Таким образом, без добавления новых устройств будет получено более эффективное использование возможностей датчика.



Рис. 9. Крепление для датчика размещается на левой стороне ...

Важное замечание. При старте робота его надо будет направлять датчиком строго на стену, чтобы процесс считывания начального значения прошел без помех. Очевидно, что изменение конструкции влечет изменение коэффициентов регулятора k_1 и k_2 . Обычно подбор начинается с пропорционального коэффициента при нулевом дифференциальном. Когда достигнута некоторая стабильность на небольших отклонениях, добавляется дифференциальная составляющая.