

Презентация

Пропорциональное регулирование

Логинов Андрей Анатольевич

Ранее изученный нами принцип релейного регулирования, предполагал, что устройство управления будет воздействовать на выходные устройства несколькими постоянными величинами управляющего воздействия: например, когда робот движется вдоль границы черного и белого цвета, то на моторы подается 100 и 0 % мощности, когда датчик находится над чёрным цветом, и 0 и 100 % мощности – когда датчик находится над белым цветом. Контроллер не может самостоятельно подать на моторы. Например: 10 и 80 %, или 35 и 8 % мощности. Поэтому релейный регулятор не отличается точностью управления. Но что делать, когда требуется именно точность? Роботы сегодня работают в самых разных сферах. И в таких, как медицина, космонавтика, производство высокоточного оборудования без точности действий роботу не обойтись. На помощь нам придет пропорциональный принцип регулирования. Его ещё нельзя назвать самым точным, но он гораздо точнее релейного регулирования.

Желаемое поведение робота (уставку) будем хранить в специальной переменной. Робот всё время будет сверять своё поведение с ней как с эталоном, и если случится некоторая разница, говорят, что робот допустил ошибку (невязку). Тогда он должен перерасчитать своё поведение так, чтобы эту ошибку устранить.

- Рассмотрим принцип пропорционального регулирования на примере следования по линии на двух датчиках света.

За уставку, как правило, берут разность между показаниями правого и левого датчиков света в момент, когда робот ещё неподвижен.

Однако если показания датчиков равны, их разность будет равна нулю. В этом случае уставкой можно пренебречь, а ошибкой считать берут разность между показаниями правого и левого датчиков света, которую будет рассчитывать контроллер каждую миллисекунду при движении робота:

$err = sensor1 - sensor2$, где $sensor1, sensor2$ – это сигналы датчиков (ошибка может быть как положительной, так и отрицательной)

Для устранения ошибки её усиливают, умножая на число. Это число называется коэффициентом пропорциональности.

$u = err * 2.2$, где: u – это управляющее воздействие.

Теперь необходимо рассчитанным управляющим воздействием воздействовать на выходные устройства (моторы) для исправления ошибки:

$mB = 80 + u$

$mC = 80 - u$

где mB и mC – это переменные для хранения мощности правого и левого моторов, а 80 – это средняя скорость

Если ошибки нет:

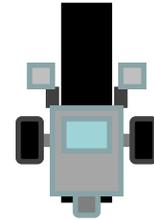
$$\text{err} = \text{sensor1} - \text{sensor2} = 0$$

$$u = \text{err} * 2 = 0 * 2 = 0,$$

$$mB = 80 + u = 80 + 0 = 80$$

$$mC = 80 - u = 80 - 0 = 80$$

На правом и левом моторе мощности равны, робот едет прямо.



Если ошибка не большая (2):

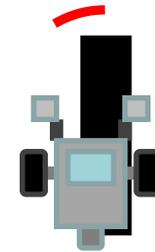
$$\text{err} = \text{sensor1} - \text{sensor2} = 2$$

$$u = \text{err} * 2 = 2 * 2 = 4,$$

$$mB = 80 + u = 80 + 4 = 84$$

$$mC = 80 - u = 80 - 4 = 76$$

На правом и левом моторе мощности отличаются не значительно, робот слегка смещается в сторону исправления ошибки.



Если ошибка не большая (10):

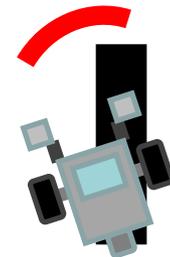
$$\text{err} = \text{sensor1} - \text{sensor2} = 10$$

$$u = \text{err} * 2 = 10 * 2 = 20,$$

$$mB = 80 + u = 80 + 20 = 100$$

$$mC = 80 - u = 80 - 20 = 60$$

На правом и левом моторе мощности сильно отличаются, робот делает резкий поворот в сторону исправления ошибки.



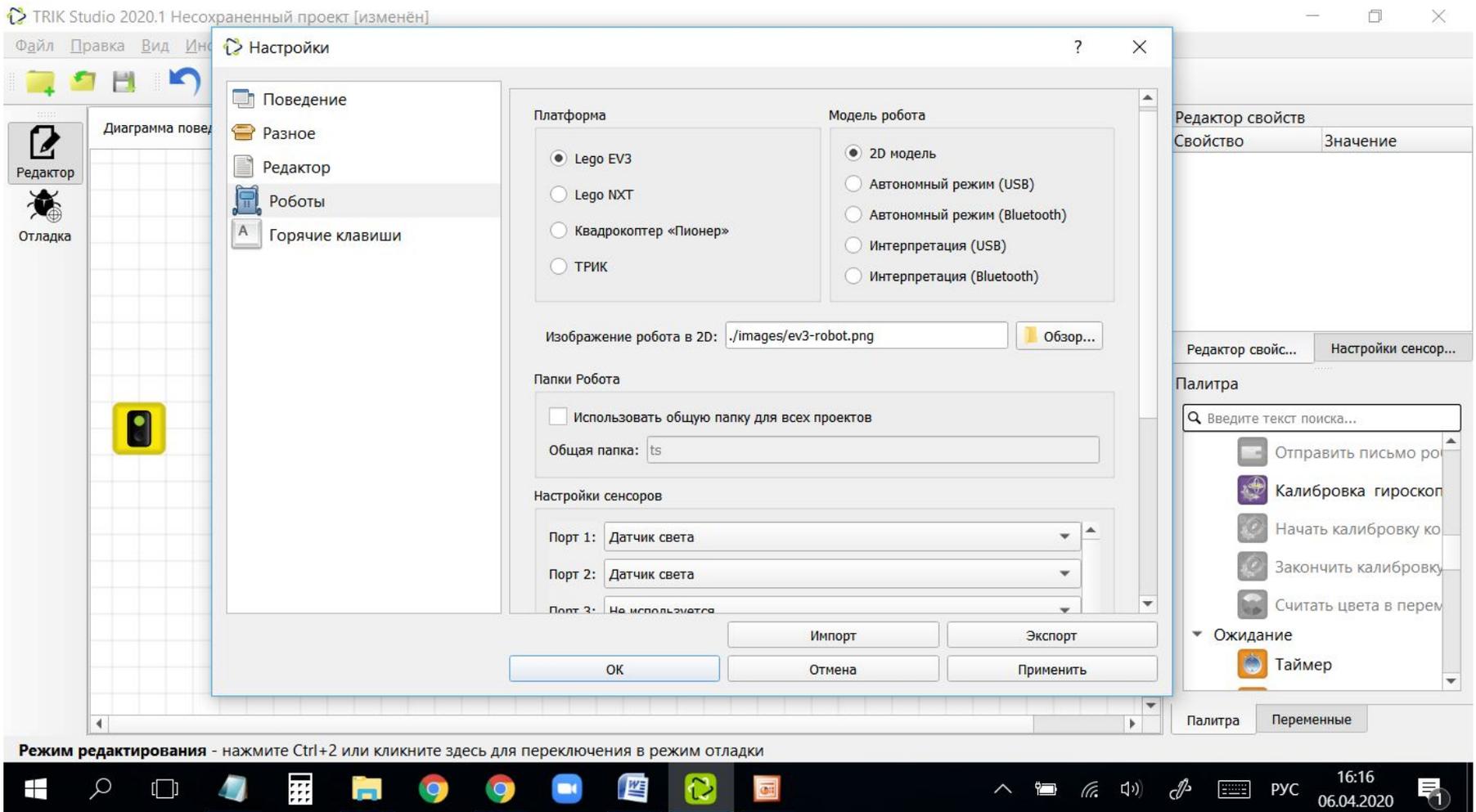
Суть пропорционального
регулирования:

Большая ошибка – сильная
реакция на неё

Маленькая ошибка – слабая
реакция на неё

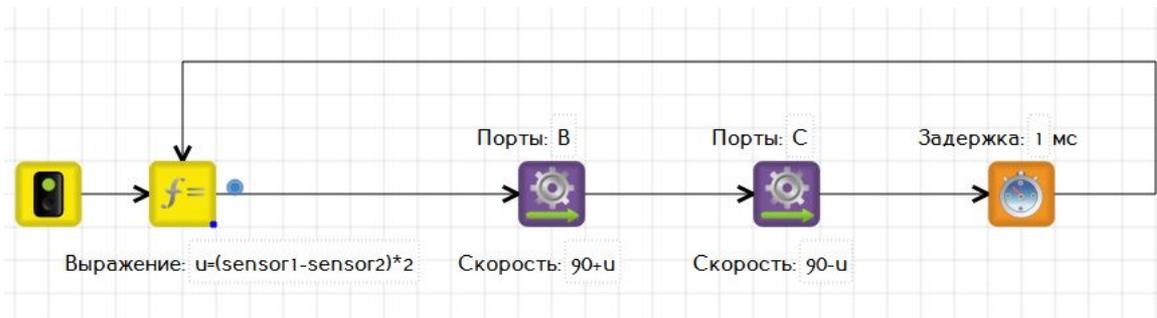
Нет ошибки – нет изменения
поведения

Для начала настроим робота в Trik Studio

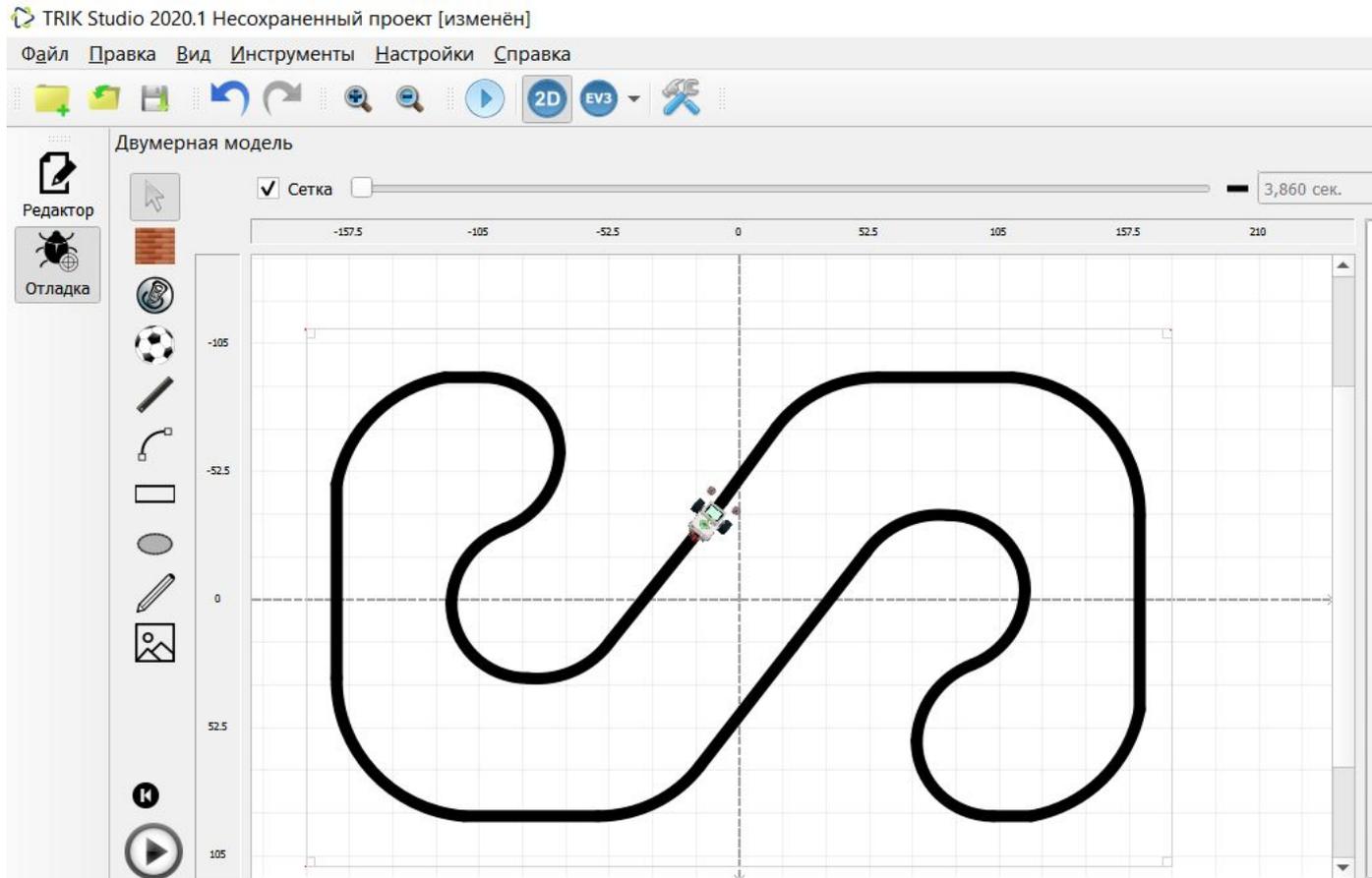


Lego EV3, датчики света подключены к портам 1 и 2. Модель робота – 2D модель

А это - блок диаграмм для робота с двумя датчиками освещенности, написанная на основе принципа пропорционального регулирования:



Отладка робота в Trik Studio



Для отладки робота используем виртуального робота на основе двухмоторной тележки и стандартное поле для следования по линии, которое имеется в коллекции Trik Studio.

Домашнее задание: Настройкой робота добиться минимального времени прохождения «круга»