

Презентация

Пропорционально-  
дифференциальное  
регулирование в среде  
LEGO Mindstorms Education  
EV3

Логинов Андрей Анатольевич

Принцип пропорционального регулирования применим во многих областях применения робота: движение по линии, движение вдоль стенки, движение по коридору, управление манипулятором (сервомотор). Иногда робот оказывается в таких условиях, когда пропорциональная составляющая, помогает ему в одних условиях, но она же в других условиях оказывается слишком велика и мешает роботу. Например, при следовании по линии на участке небольших поворотов, пропорциональная составляющая справляется очень хорошо, но стоит роботу войти в крутой поворот, ошибка значительно возрастает и робот резко реагирует на неё. Его выбрасывает на другую сторону линии, там ошибка так же огромна, но уже с другим знаком, и робота кидает в другую сторону. И так некоторое время робот не может успокоиться и делает резкие рывки из стороны в сторону. Не редко это всё заканчивается сходом с линии. Чтобы предотвратить такое поведение, необходимо ввести такую силу, которая будет, как бы, успокаивать реакцию робота при большой ошибке. Это будет возможно, когда в формулу управления, кроме пропорциональной будет добавлена ещё и дифференциальная составляющая. Итак, пропорциональная составляющая управляет поведением робота всегда, а дифференциальная составляющая подключается к управлению только когда ошибка становится очень большой, причём дифференциальная составляющая при этом несколько подавляет пропорциональную

Пропорциональное регулирование:

$err = sensor1 - sensor2$

$u = err * 2.2$

где:  $u$  – это управляющее воздействие

$sensor1, sensor2$  – это сигналы датчиков

$err$  – это ошибка (невязка, при условии, что уставка равна 0)

$2.2$  – это коэффициент пропорциональности, усиливающий ошибку

Пропорционально-дифференциальное регулирование:

$err = sensor1 - sensor2$

$u = err * 2.2 + (err - erold) * 15$

где:  $u$  – это управляющее воздействие

$sensor1, sensor2$  – это сигналы датчиков

$err$  – это ошибка (невязка, при условии, что уставка равна 0)

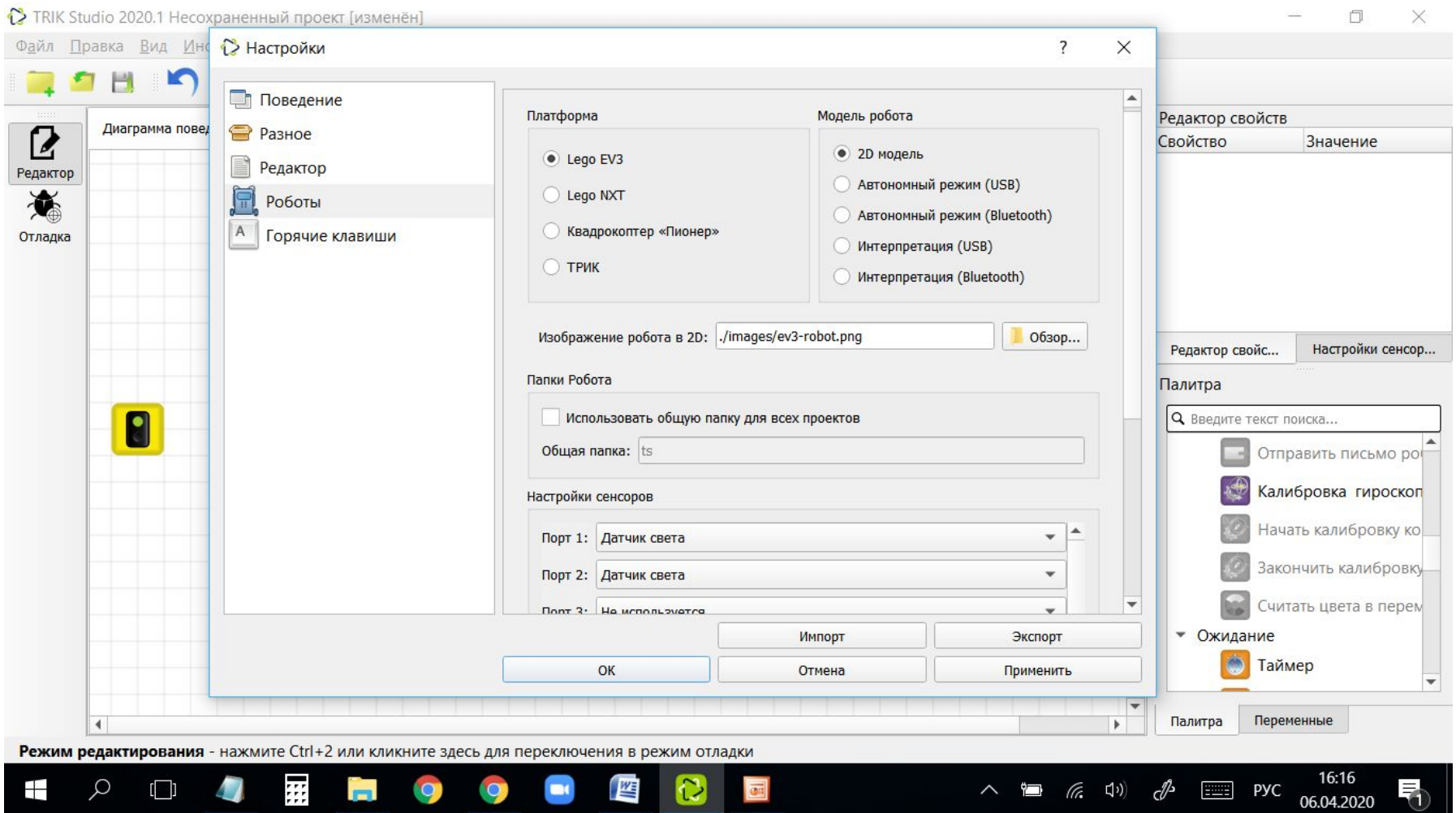
$2.2$  – это коэффициент пропорциональности, усиливающий ошибку

$erold$  – это старая ошибка (которая была посчитана в предыдущей итерации цикла). Для того, чтобы программа смогла сделать первый расчет, в переменную  $erold$  перед циклом необходимо записать 0

Для подготовки расчетов в следующей итерации, в переменную  $erold$

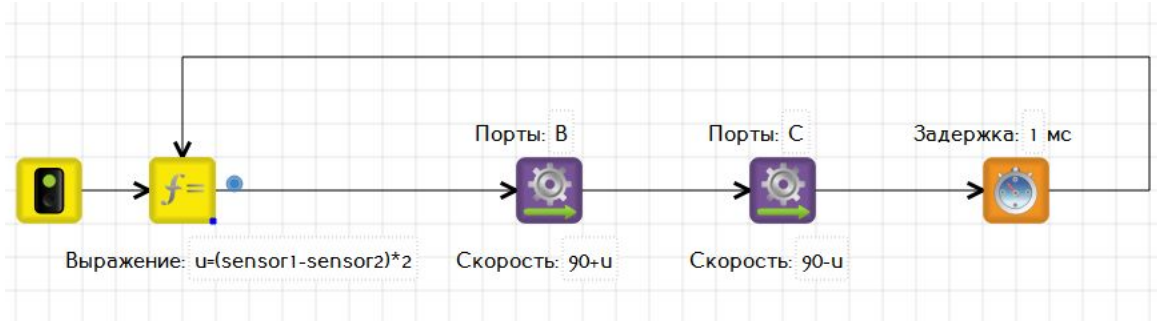
необходимо записать текущую ошибку  $err$ , которая к следующей итерации «состарится» и будет уже старой ошибкой.

# Для начала настроим робота в Trik Studio

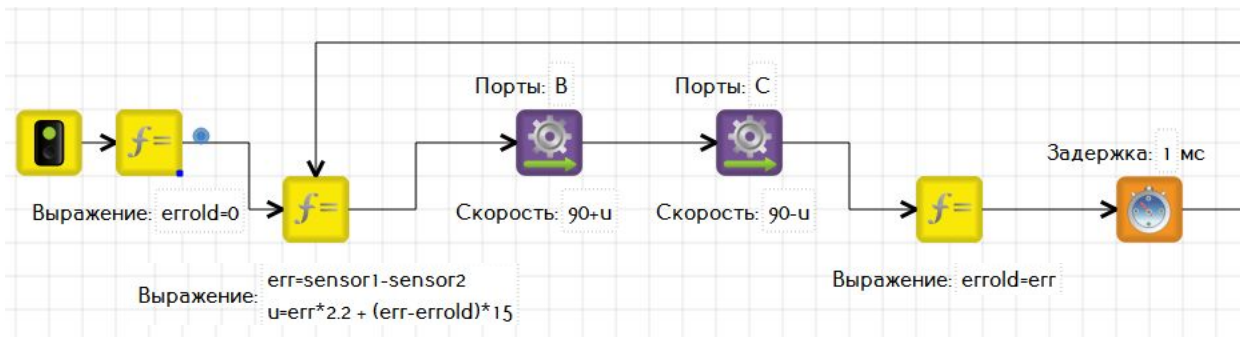


Lego EV3, датчики света подключены к портам 1 и 2. Модель робота – 2D модель

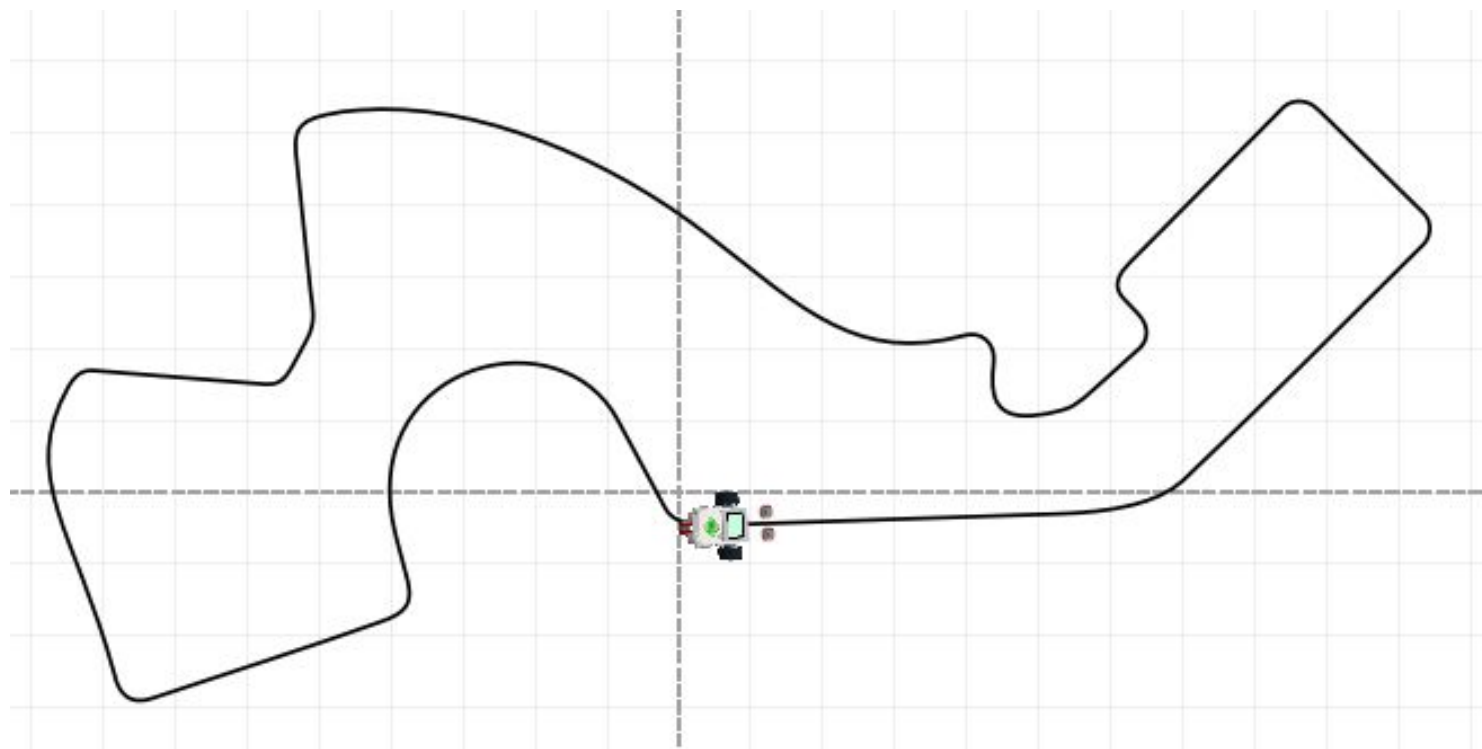
А это - блок диаграмм для робота с двумя датчиками освещенности, написанная на основе принципа пропорционального регулирования:



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



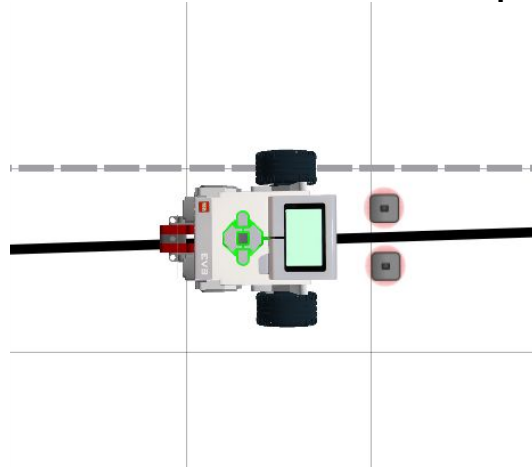
# Отладка робота в Trik Studio



Для отладки робота используем виртуального робота на основе двухмоторной тележки и поле `Follow_narrow_line`, которое имеется в коллекции Trik Studio во вкладке «Картинки».

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

Робот должен выглядеть примерно так:



Несмотря на то, что датчики света не имеют видимых соединений с роботом, они будут перемещаться вместе с ним, оставаясь на своих местах относительно корпуса робота.

Дальнейшая настройка производится так же, как и на реальном роботе: поворот не в ту сторону – поменяй местами датчики или моторы на вкладке моторы, теряет линию на поворотах – подбирай коэффициент пропорциональности или увеличивай разницу в скоростях между правым и левым колесом

Датчики не должны быть слишком узко, но и не слишком широко