Teaching Children Proportional Control using ROBOLAB 2.9

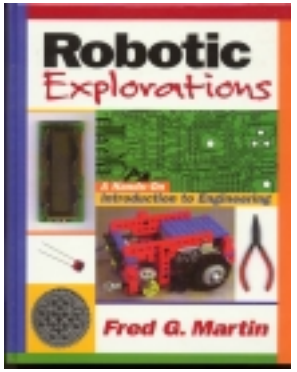
By Dr C S Soh

robodoc@fifth-r.com

Objective

- Using ROBOLAB 2.9, children can experiment with proportional control the same way as undergraduates in MIT and other learned institutions are doing.

Inspiration



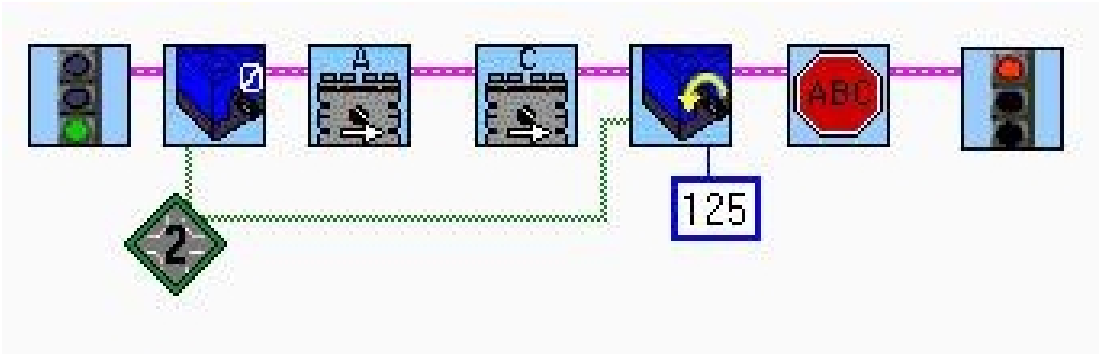
- Fred G. Martin “Robotic Explorations - A Hands-On Introduction to Engineering”.
- Using mobile robotics systems to introduce undergraduates to engineering design and problem solving.
- Chap 5.2.1 - Proportional Control.

Bang-bang vs Proportional Control

Power is no good without control

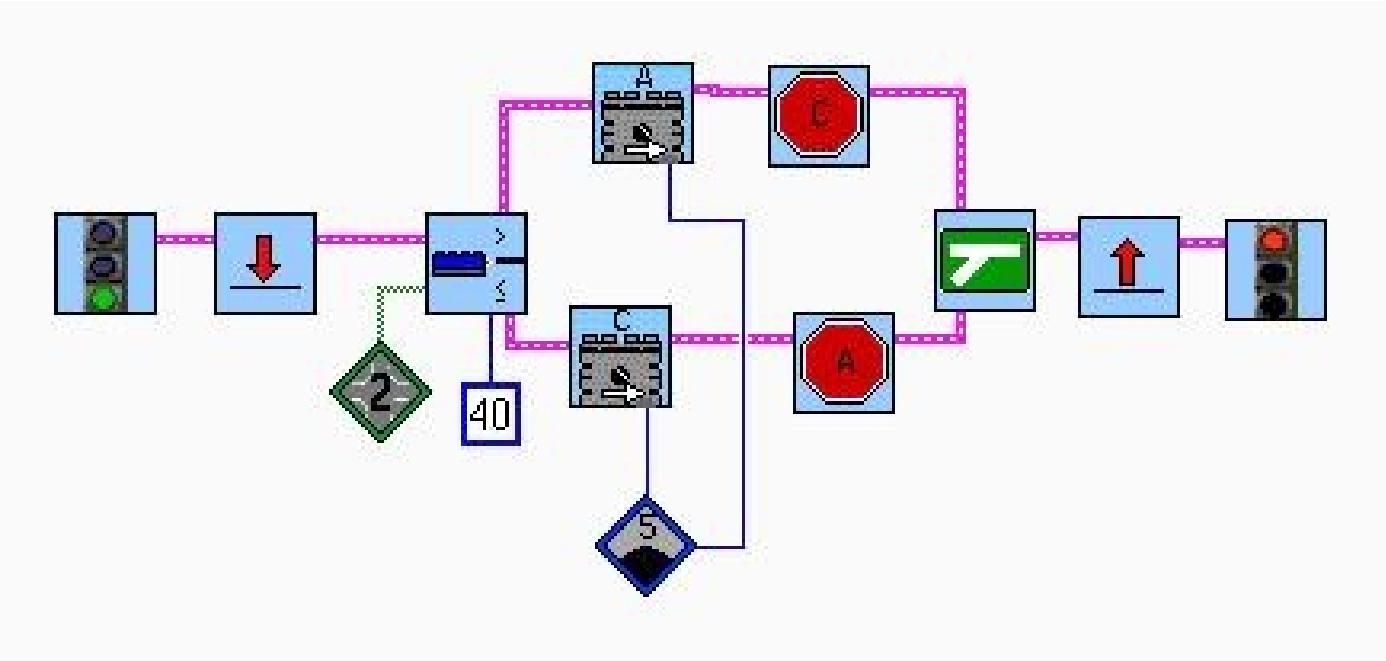
Bang-bang (on/off, all or nothing) control

- Go full speed, then abrupt stop!

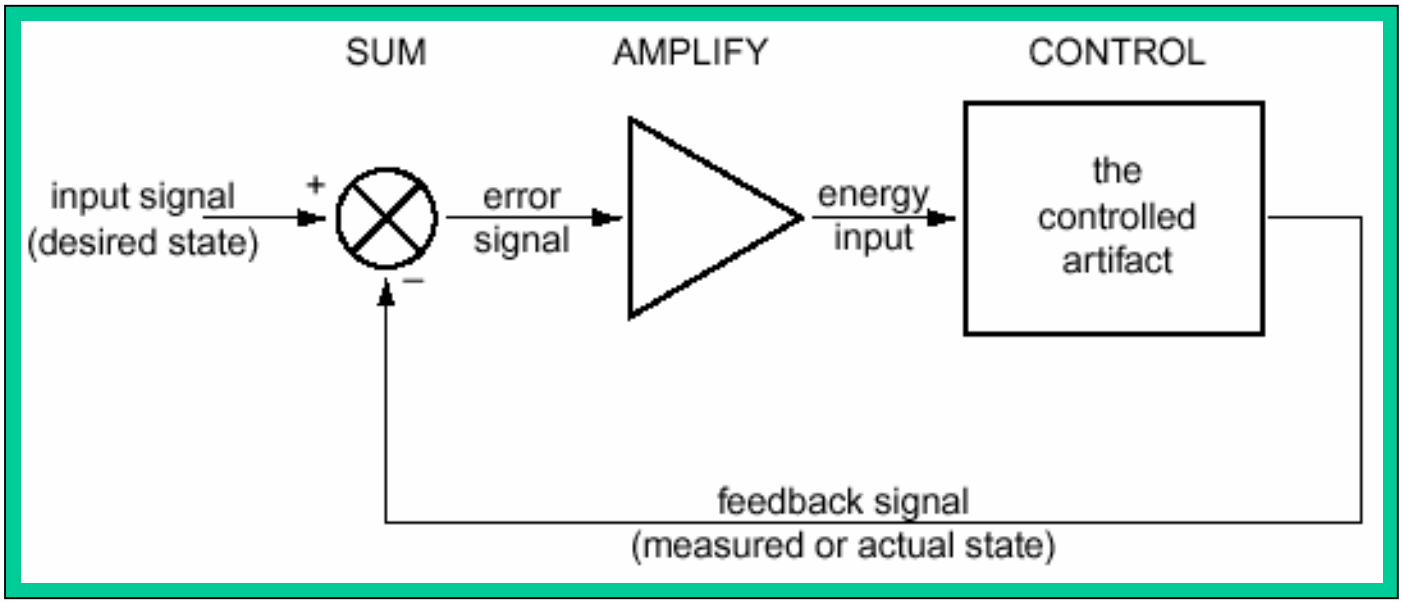


Bang-bang Line Following

- Waggles about the line



Feedback Control



We need to know where we are
in relation to where we want to be

Proportional Control

- Response of the control algorithm is proportional to the amount of the error.
- Example: a motorist approaching a traffic light that has just turned red.
- He/she would slow down gradually as he/she gets near to the traffic light.
- **Warning:** some motorists are bang-bang!

Investigating Proportional Control

Doing it like MIT

MIT vs Kids

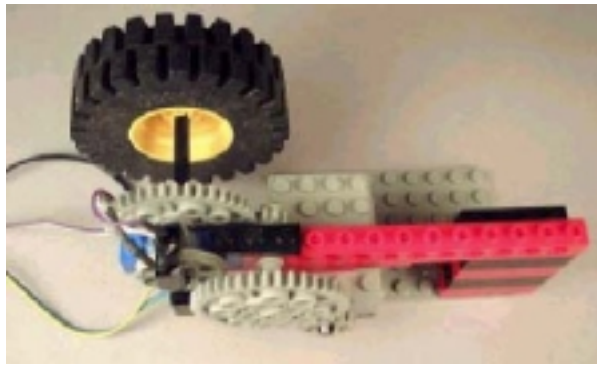
MIT

- LEGO parts
- Handyboard micro-controller
- Interactive C software
- C language

Kids

- LEGO parts
- RCX micro-controller
- ROBOLAB 2.9 software
- Graphical language based on LabView

MIT Setup

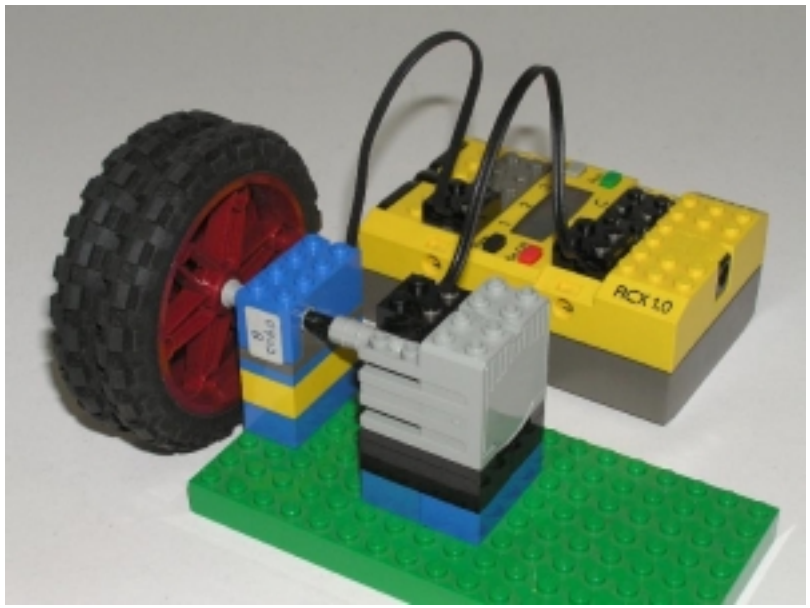


Yep, they use LEGO at MIT, too

The proportional-derivative control test system includes a dc motor driving a two-stage gear reduction, and a large LEGO wheel which gives the system a fair bit of momentum (load on the system).

At the middle stage of the gearing, a quadrature-based shaft encoder keeps track of the shaft position.

Kids' Setup



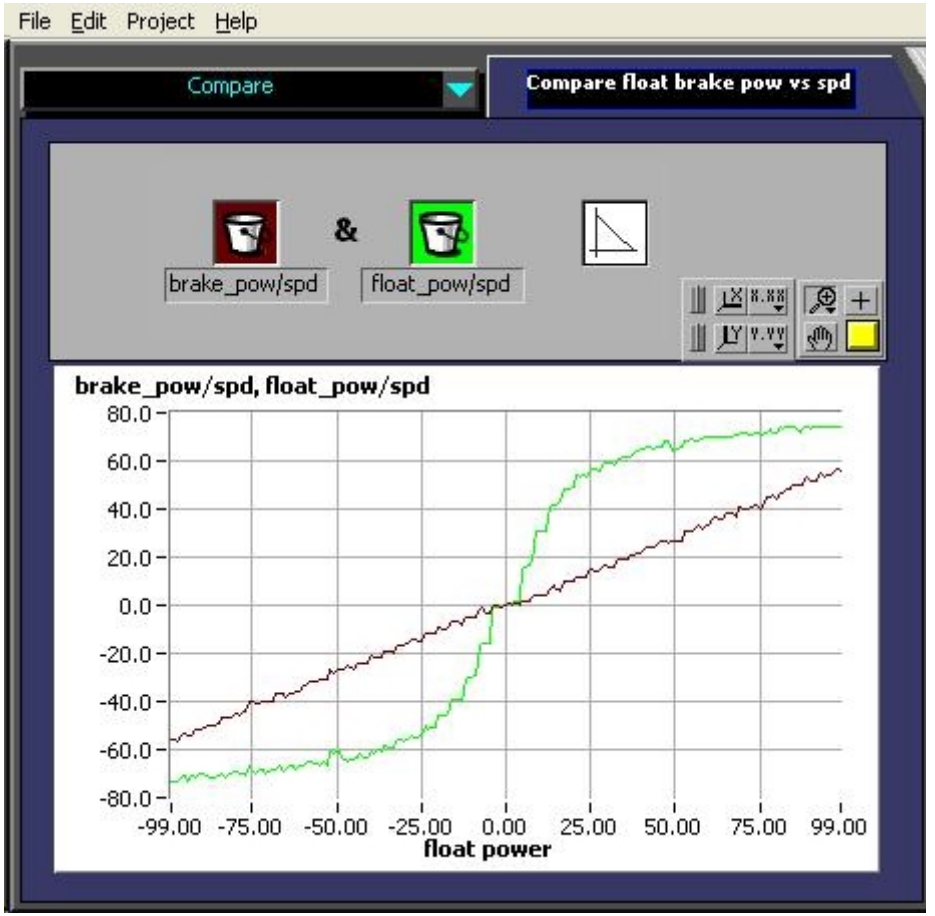
- A LEGO 9v geared motor is connected directly to a rotation sensor.
- Two of the large motor-cycle wheels with tyres were used as the load.

Using ROBOLAB 2.9 Advanced Features

- ROBOLAB 2.9 uses new firmware that allows setting 100 motor power levels in both forward and reverse direction.
- Furthermore, it is possible to set PWM to use brake instead of float in between pulses.
- This provides a very linear relationship between motor power and resultant motor speed which is necessary for proportional control to be successful.

Motor Linearity with ROBOLAB 2.9

This chart shows clearly the advantage of braking in between PWM pulses to achieve a more linear response between motor power and speed.



Brown = brake
Green = float

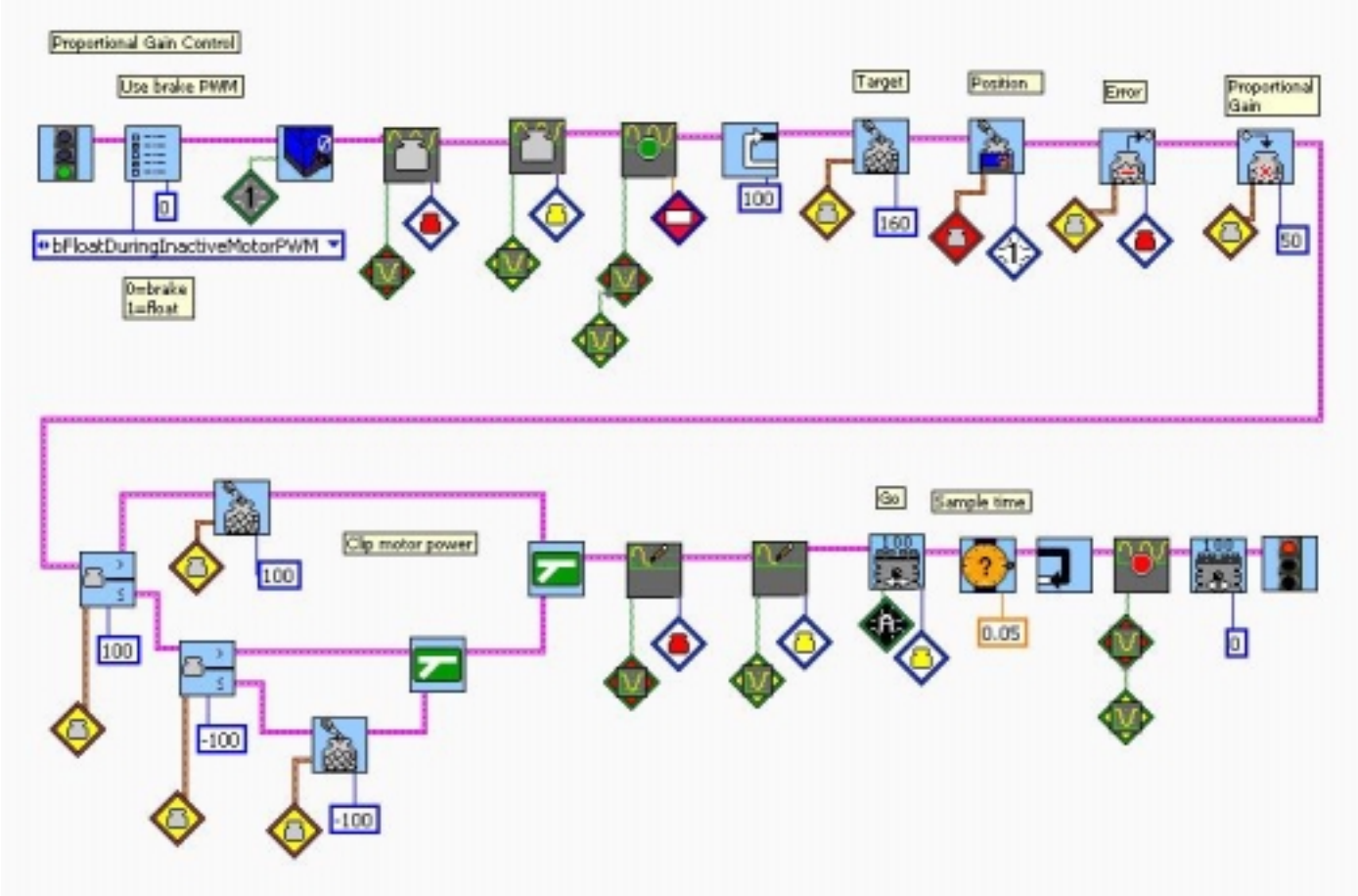
Method

- The aim is to get to a rotational position of 160 ticks corresponding to 10 revolutions of the wheels.
- Proportional gain was adjusted from 1, 5, 10, 20 and 50.
- Motor power was clipped to between -100 and 100.
- Sample time of 0.05 sec was used.
- Position (ticks) and motor power were data logged for 100 sample points.

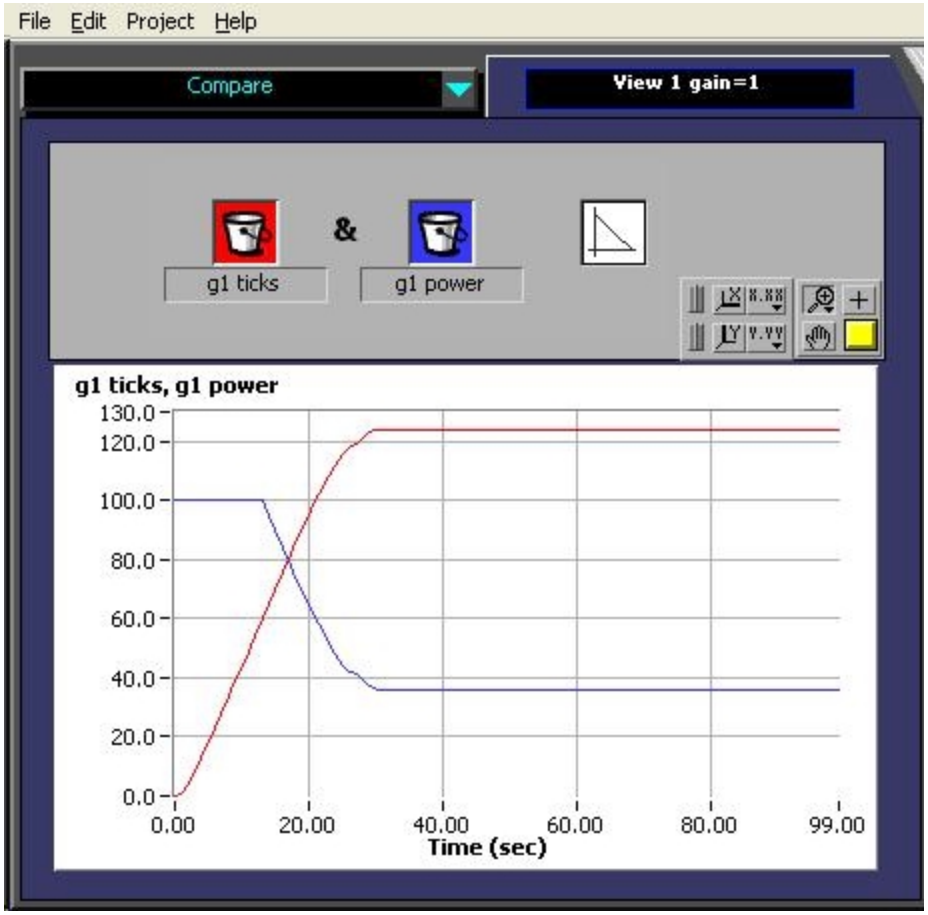
Program in words

- Find the error between the target and actual position (in rotation sensor clicks)
- Multiply the error term by the gain factor
- Keep the resulting number between -100 and 100
- Set the motor power accordingly

Program in ROBOLAB 2.9

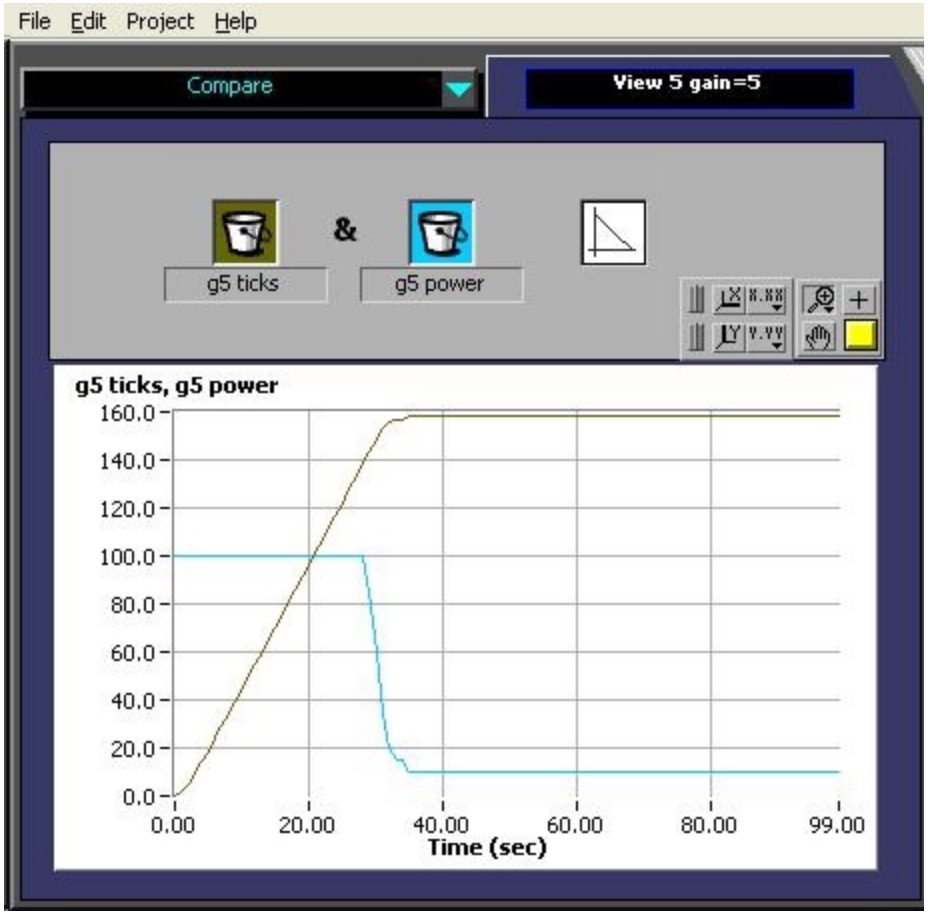


Results: Gain=1



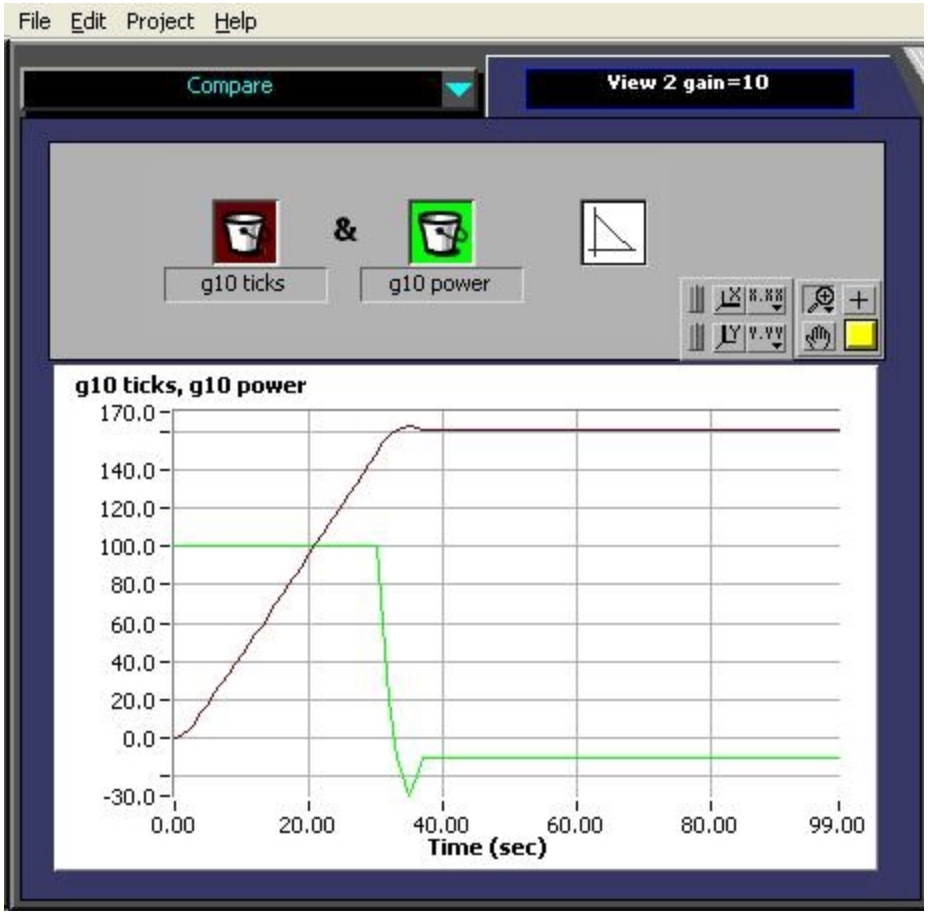
- Note that full power is applied at the start.
- Motor power is reduced as it gets closer to the target position.
- But the wheel falls considerably short of the target position of 160 ticks.
- If this were a car, it will never reach its destination! Think of that.

Results: Gain=5



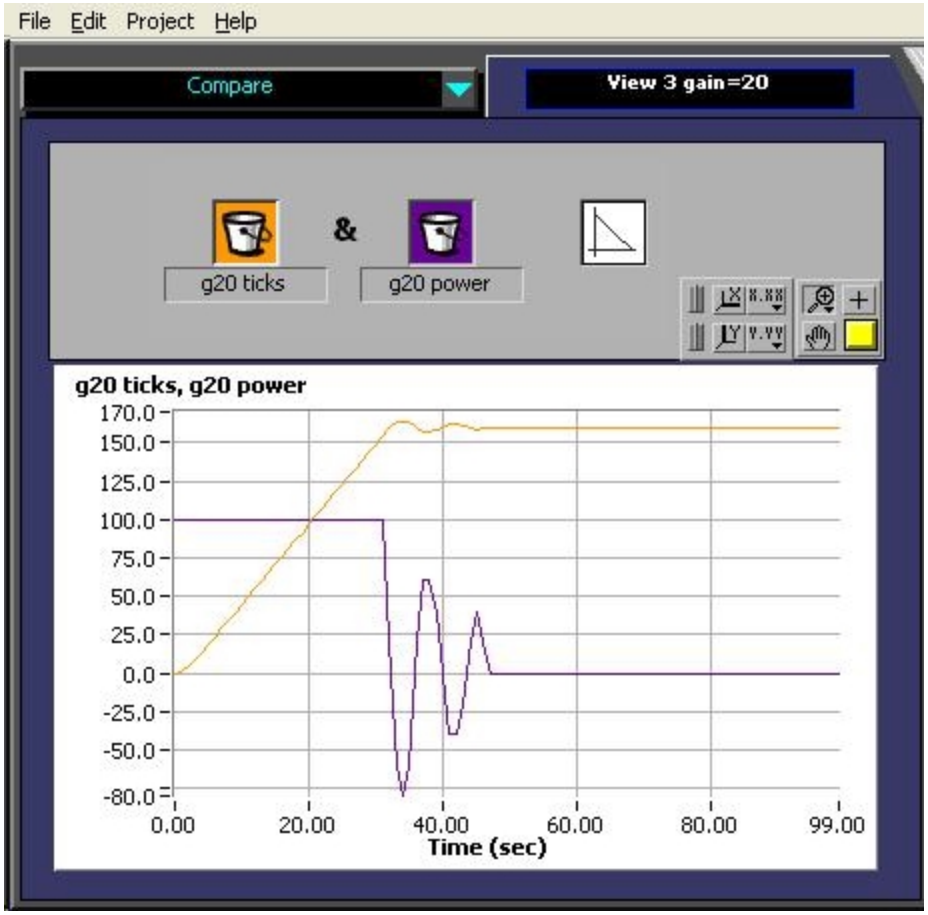
- This is much better.
- The wheel is close to but not exactly on target.
- The slight difference is known as the offset.

Results: Gain=10



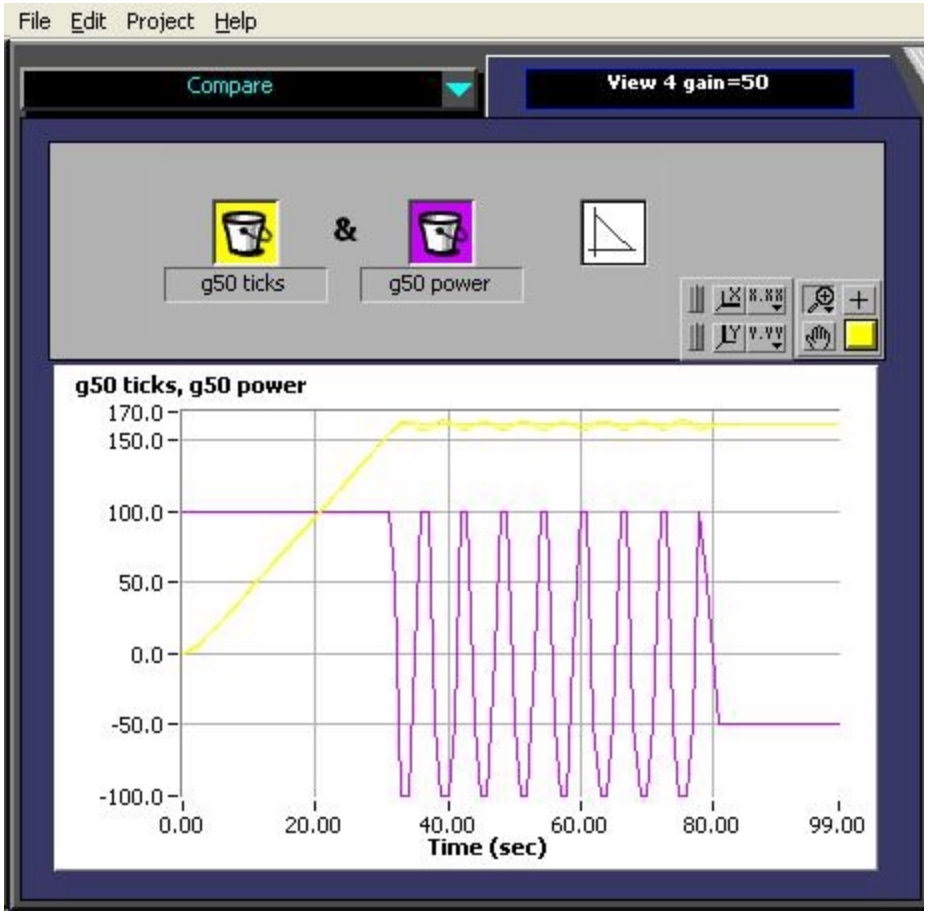
- OK, the wheel comes to rest at the target position.
- Note that there is a slight overshoot before reaching the target position.
- So the motor had actually to reverse direction to get back on target.

Results: Gain=20



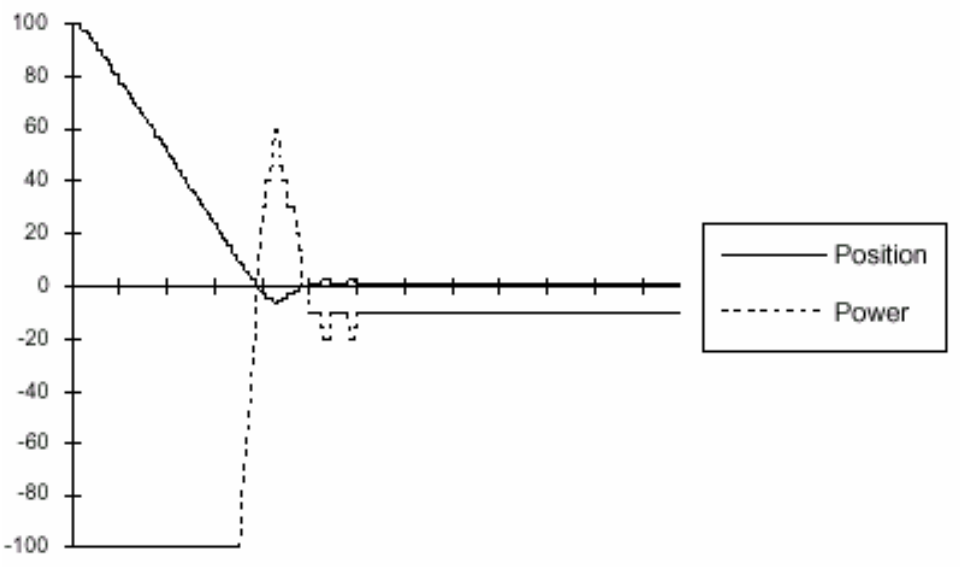
- At this stage, the kids want to rack up the gain!
- But as the gain gets larger, oscillations has crept in.

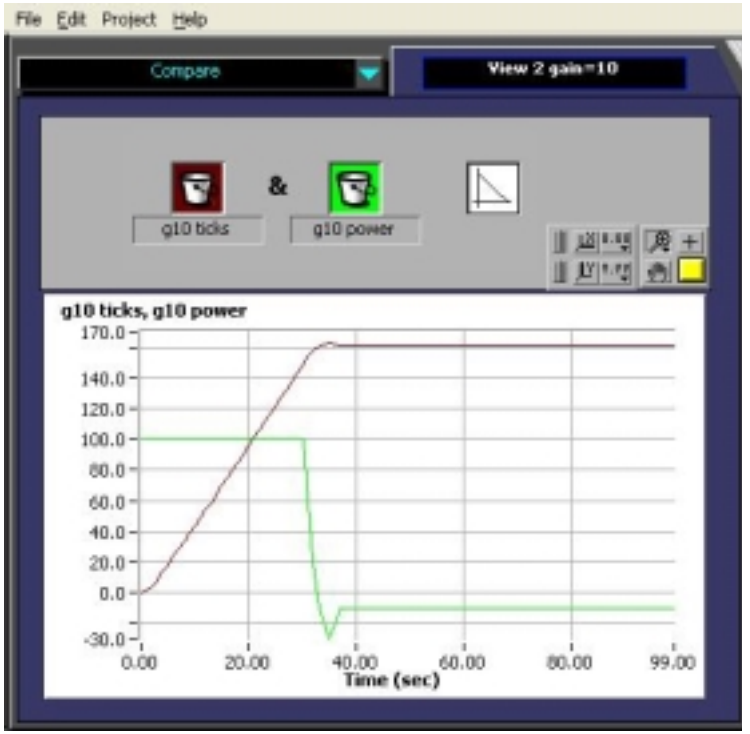
Results: Gain=50



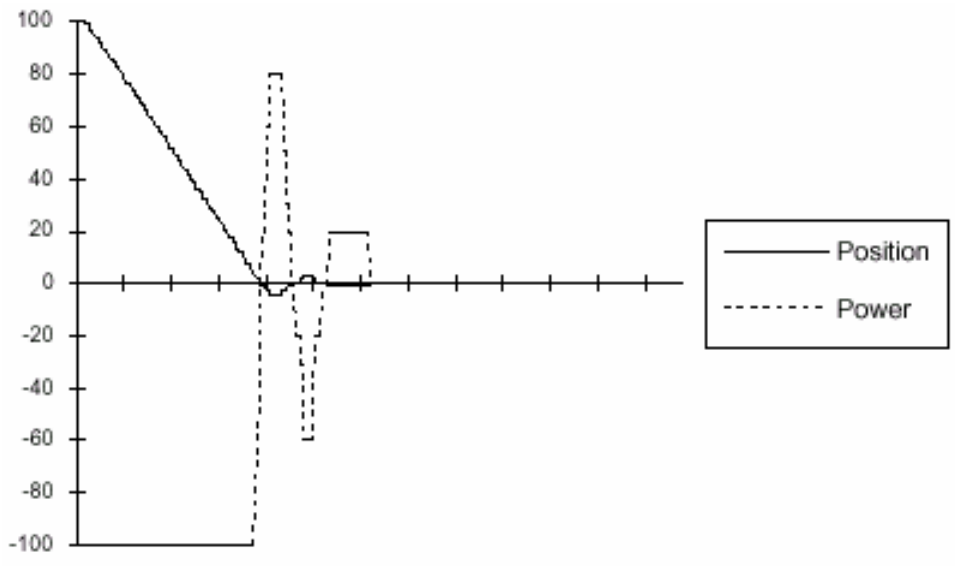
- This is what you get when you are too greedy!
- The system has become unstable.
- This is something engineers what to avoid.
- LEGO Engineers should do likewise!

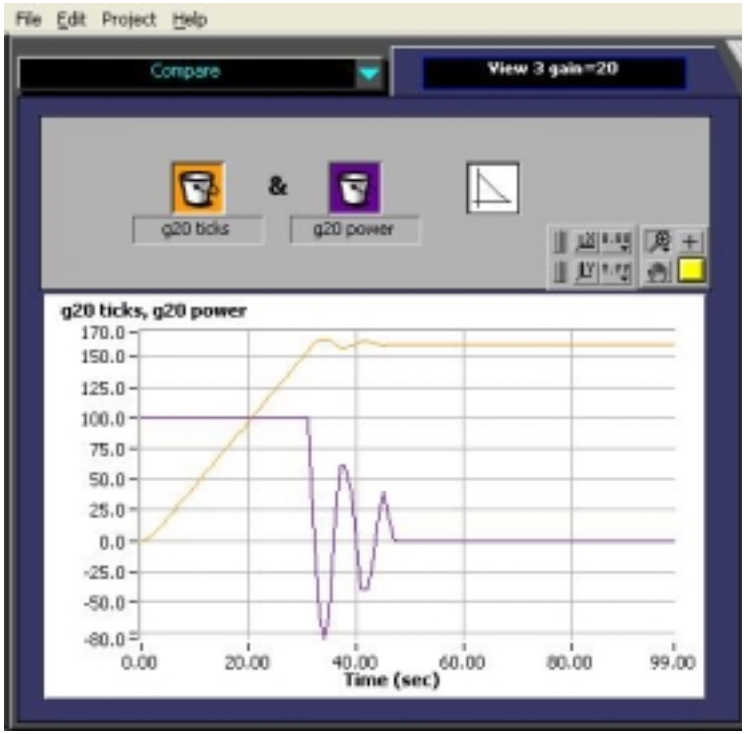
Validation: MIT Gain=10



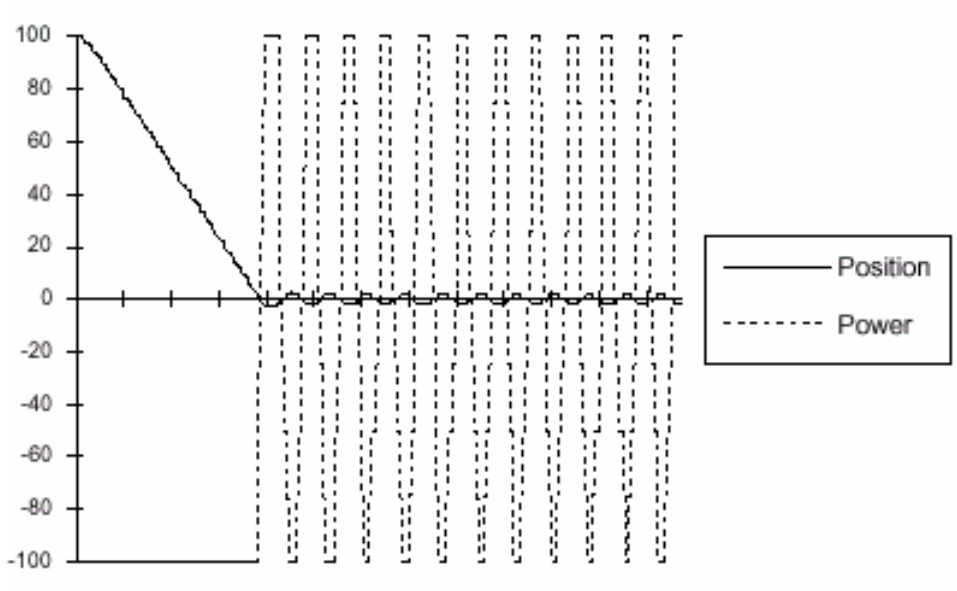


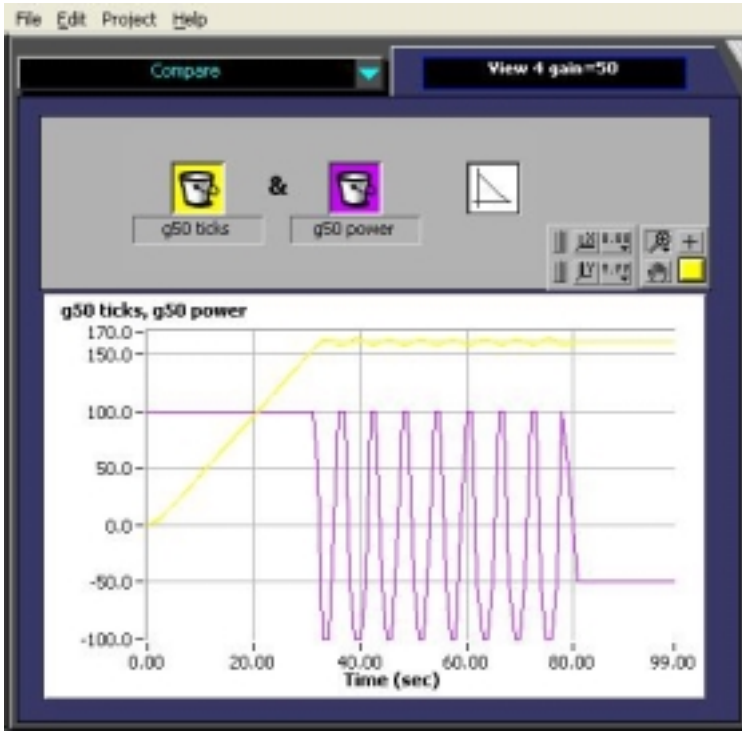
Validation: MIT Gain=20





Validation: MIT Gain=50





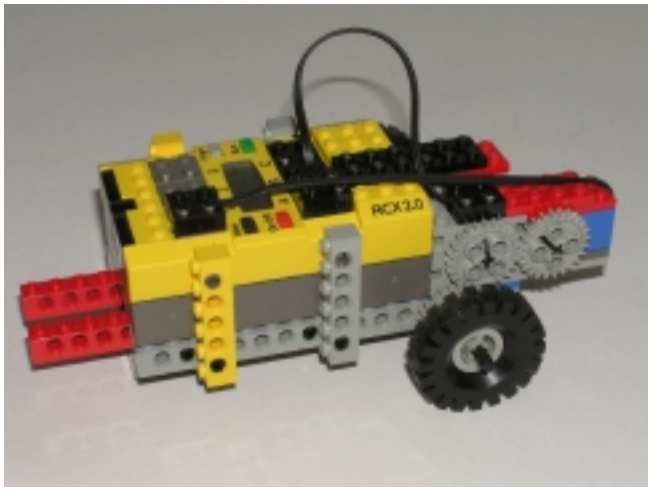
Extension

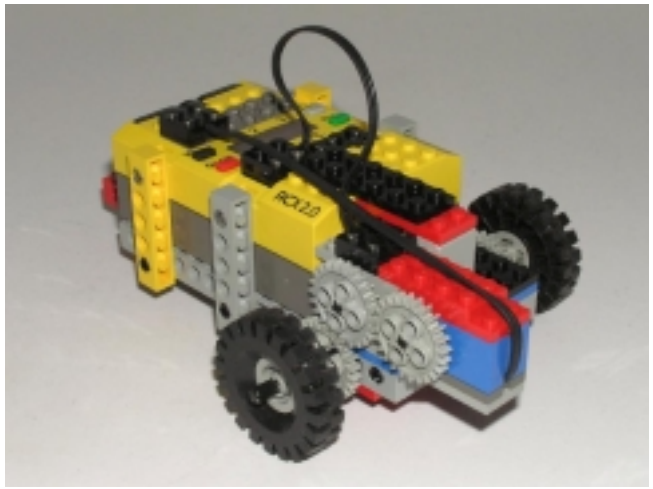
- After the wheel has come to rest but while data is being logged, try to turn the wheel by hand.
- What do you feel?
- What do you notice as the proportional gain is increased?

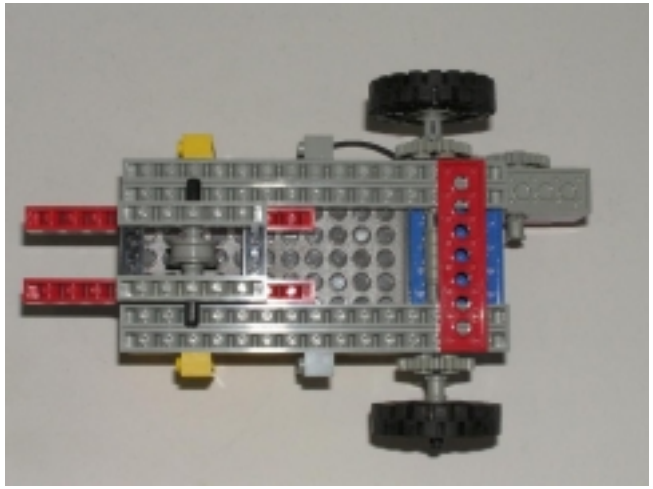
Next Lesson...

- In the next assignment, we will try to include the other components of PID control, namely derivative and integral elements.
- Now you know why you have to study all that differentiation and integration stuff.

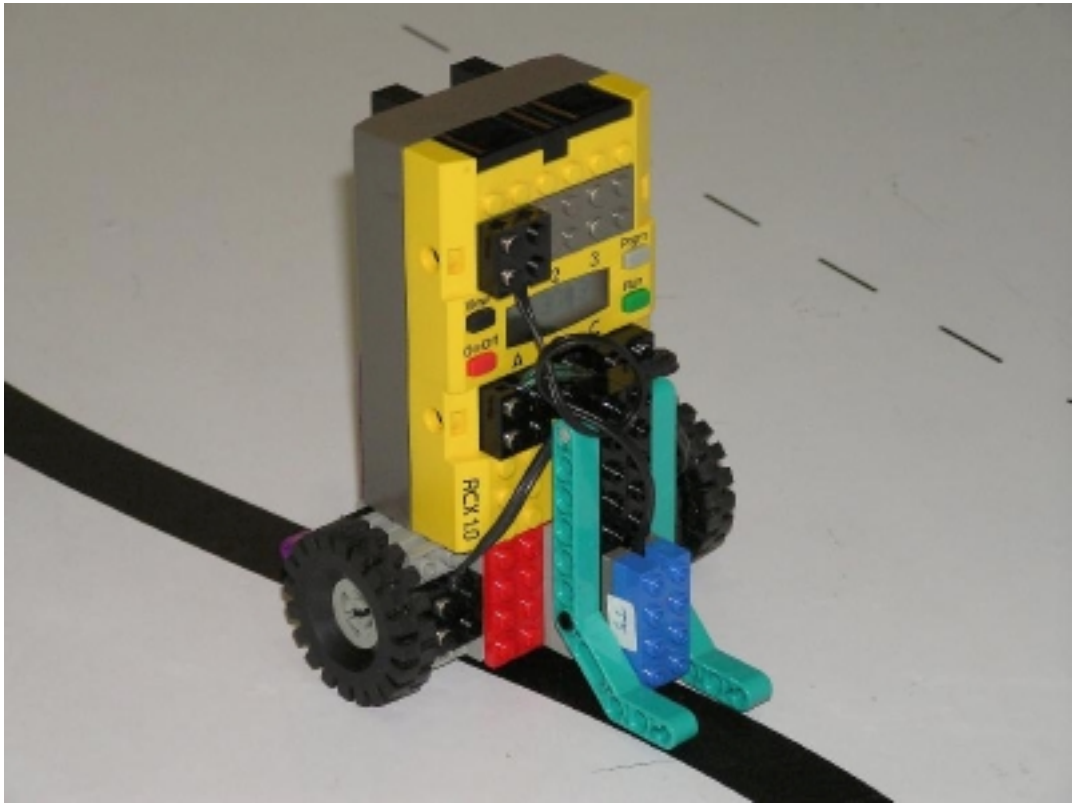
Car using proportional control



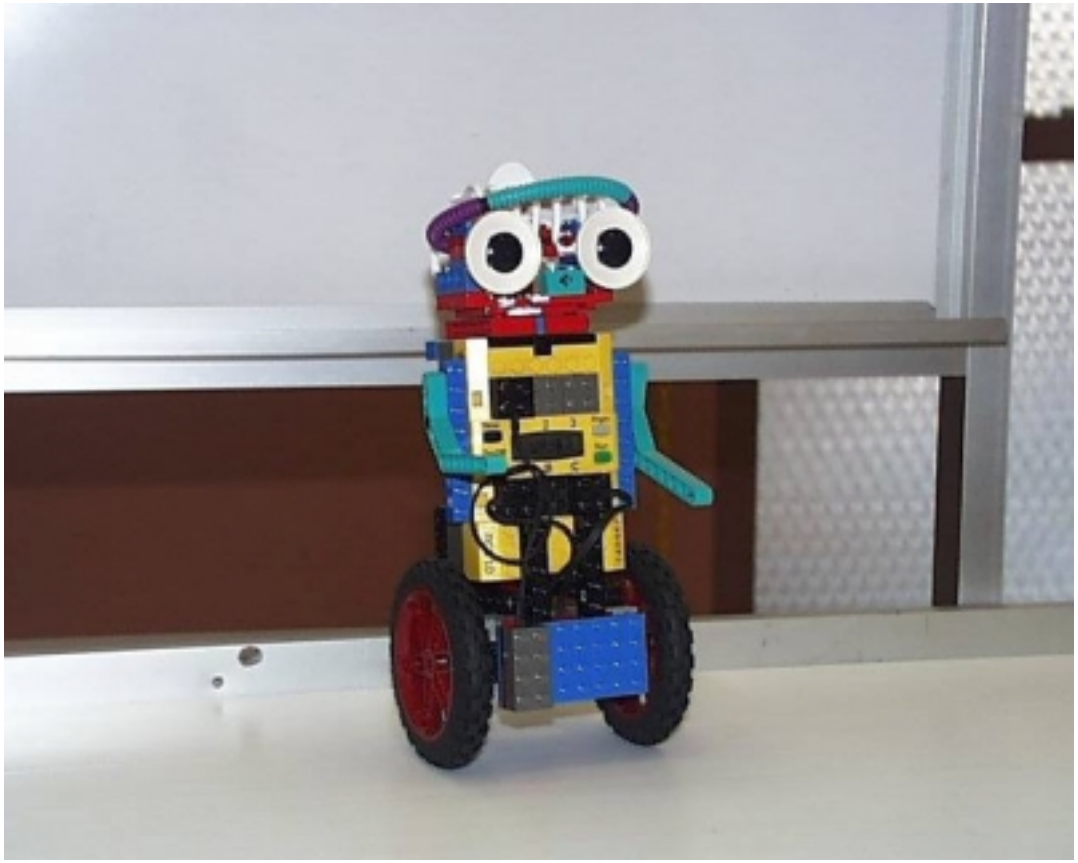




Proportional Line Follower



Balancing Bot



Proportional Control - Other Applications

- Wall following
- Robot arm
- Inverted pendulum
- etc.

Q & A

By Dr C S Soh

robodoc@fifth-r.com