

Teaching Children Proportional Gain Control using ROBOLAB 2.9
By Dr Soh Chio Siong
robodoc@fifth-r.com

Abstract

Proportional gain control is an important concept in real life engineering. Until the advent of ROBOLAB 2.9 it was not possible to teach it without resorting to advanced C programming based languages like BrickOS or Interactive C. With the availability of advanced motor control and other commands in ROBOLAB 2.9, it is now possible for school children to learn and apply proportional control to their robots.

This paper describes an ROBOLAB Investigator activity to introduce school children to the concept of proportional gain control. Armed with this knowledge, they can then use it to program a model car to travel a precise distance and other applications.

Introduction

In standard firmware as used in ROBOLAB 2.5.4, it is difficult, if not impossible, to implement proportional gain control with only 5 available power levels. And they are extremely non-linear.

New firmware in ROBOLAB 2.9 allows setting 100 power levels in both forward and reverse direction. Furthermore, it is possible to set the motor pulse width modulation (PWM) to use brake in between pulses. This provides a very linear relationship between power and resultant motor speed, which is necessary for proportional control to be successful.

Objective

To show how children can implement proportional gain control using ROBOLAB 2.9. They can then apply it to program a model car to travel a precise distance. Other applications would include a proportional control line follower and maybe even a two-wheeled balancing robot.

Method

Test set-up

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 (Fig 1).

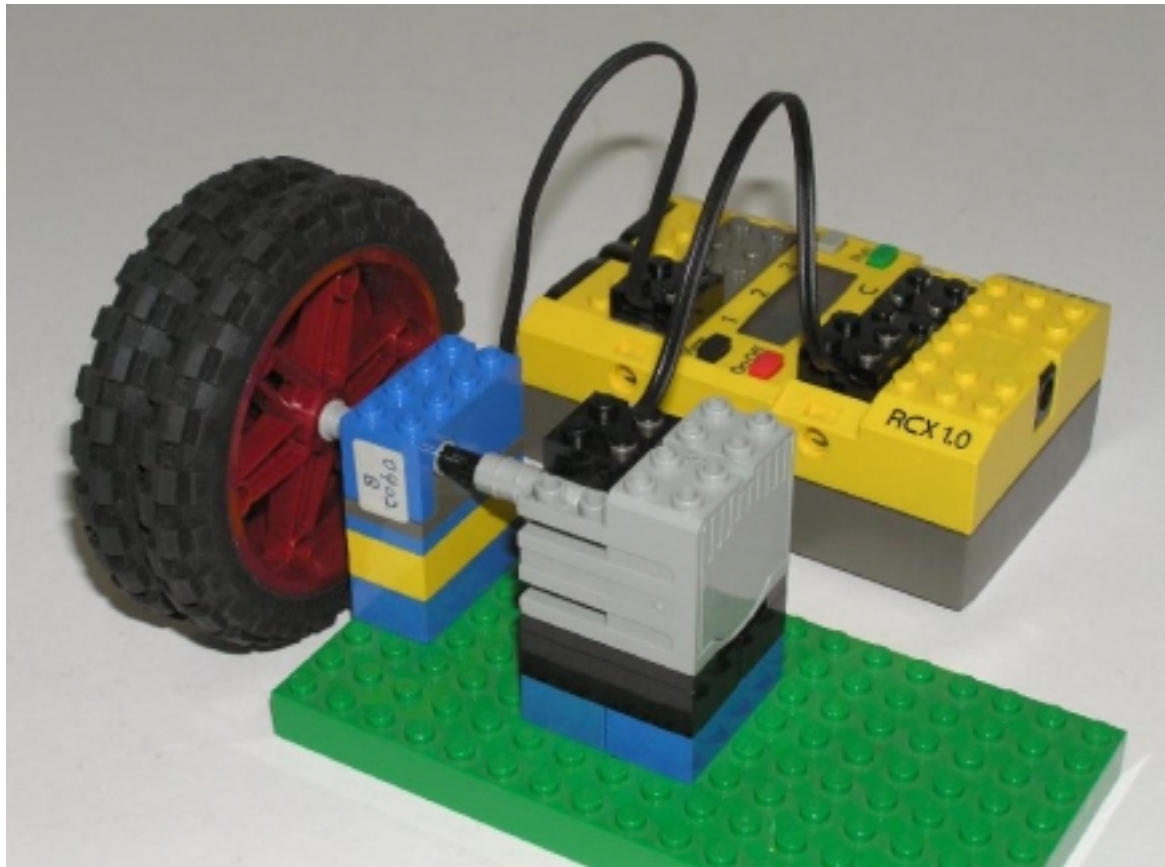


Fig 1. The test setup

Test Procedure

The aim is to get to a rotational position of 160 ticks corresponding to 10 revolutions of the wheels.

Motor power was clipped to between -100 and 100.

Sample time of 0.05 sec was used.

Rotational position (ticks) and motor power were data-logged for 100 sample points.

The program was run with proportional gain settings from 1, 5, 10, 20 and 50

Plots of motor power and rotational position were made.

Program in words

Find the error between the target and actual position (in rotation sensor ticks).

Multiply the error term by the gain factor.

Keep the resulting number between -100 and 100.

Set the motor power accordingly.

Program in ROBO LAB 2.9

The program is shown at Fig 2.

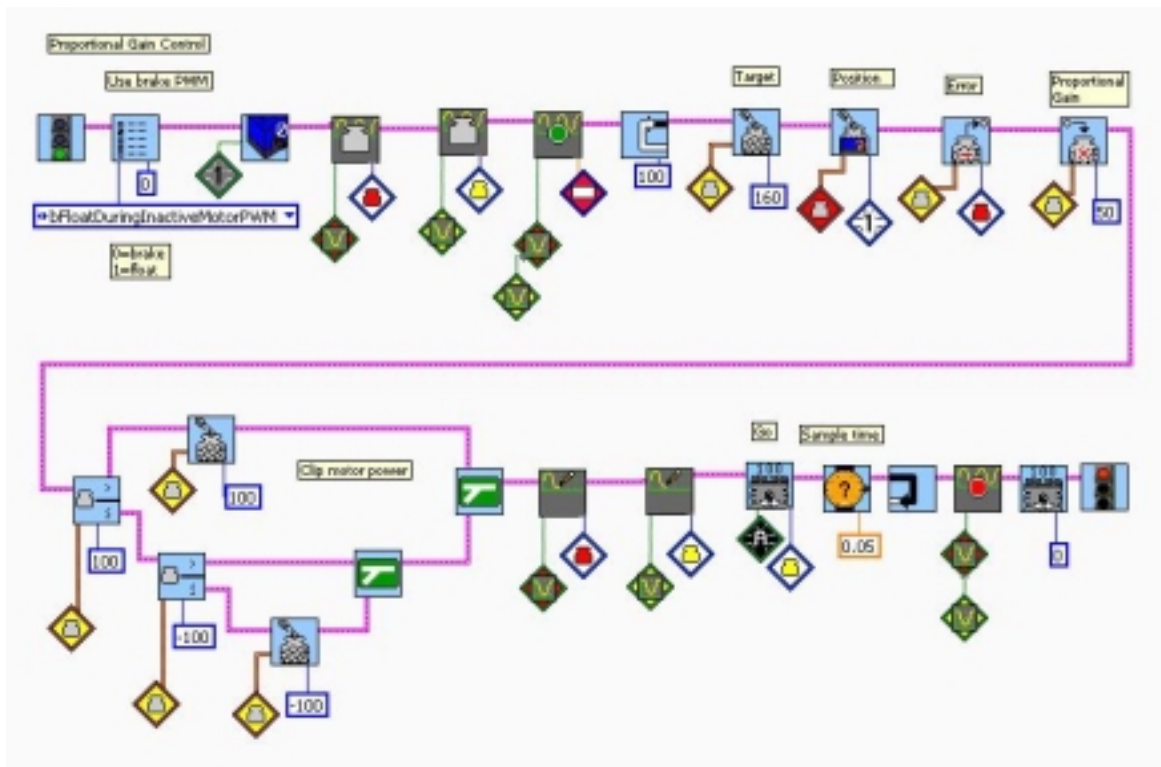


Fig 2. The program in ROBOLAB 2.9

Results

Gain = 1

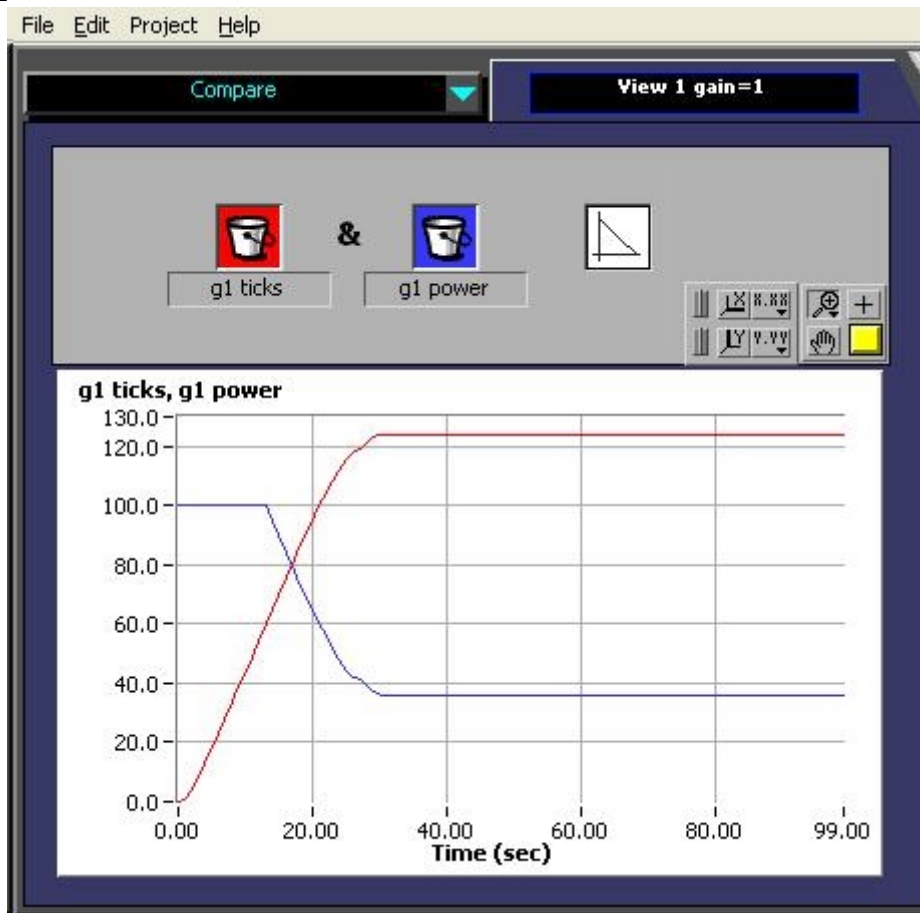


Fig 3. Gain = 1

Note that full power (100) 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 model car, it will never reach its destination! Think of that.

Gain = 5

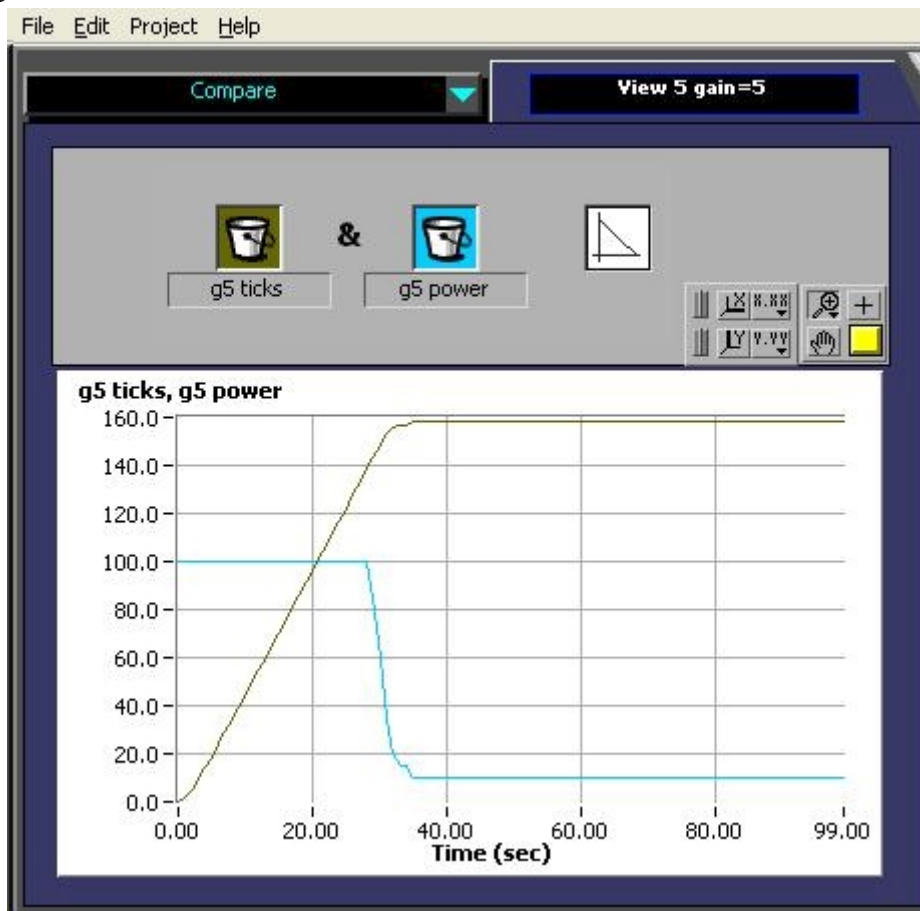


Fig 4. Gain =5

This is much better.

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

Gain = 10

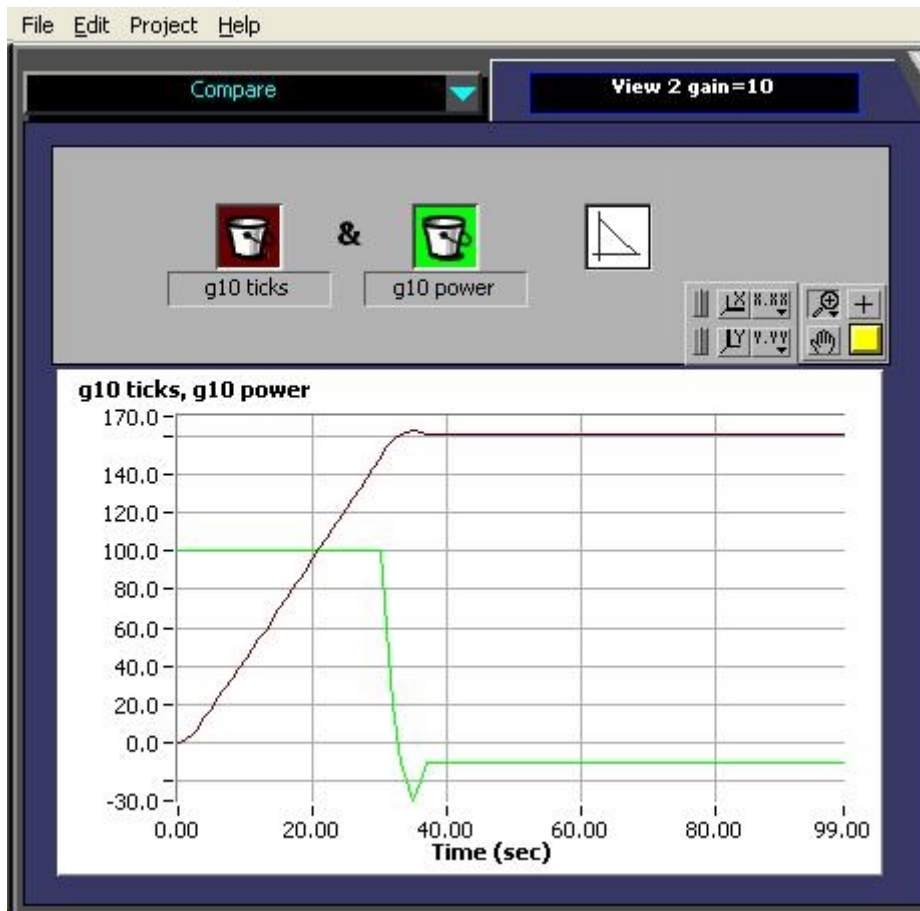


Fig 5. Gain =10

The wheel comes to rest at the target position.
But note that there is a slight overshoot before reaching the target position.
So the motor had actually to reverse direction once, to get back on target.

Gain = 20

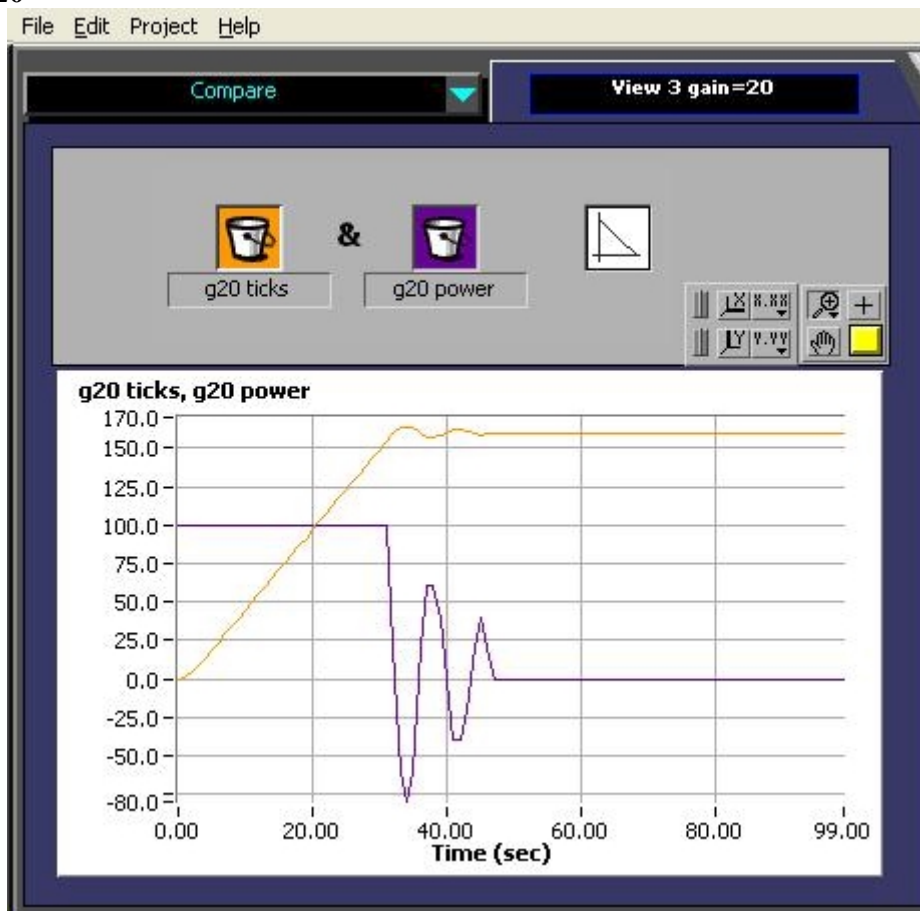


Fig 6. Gain =20

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

Gain = 50

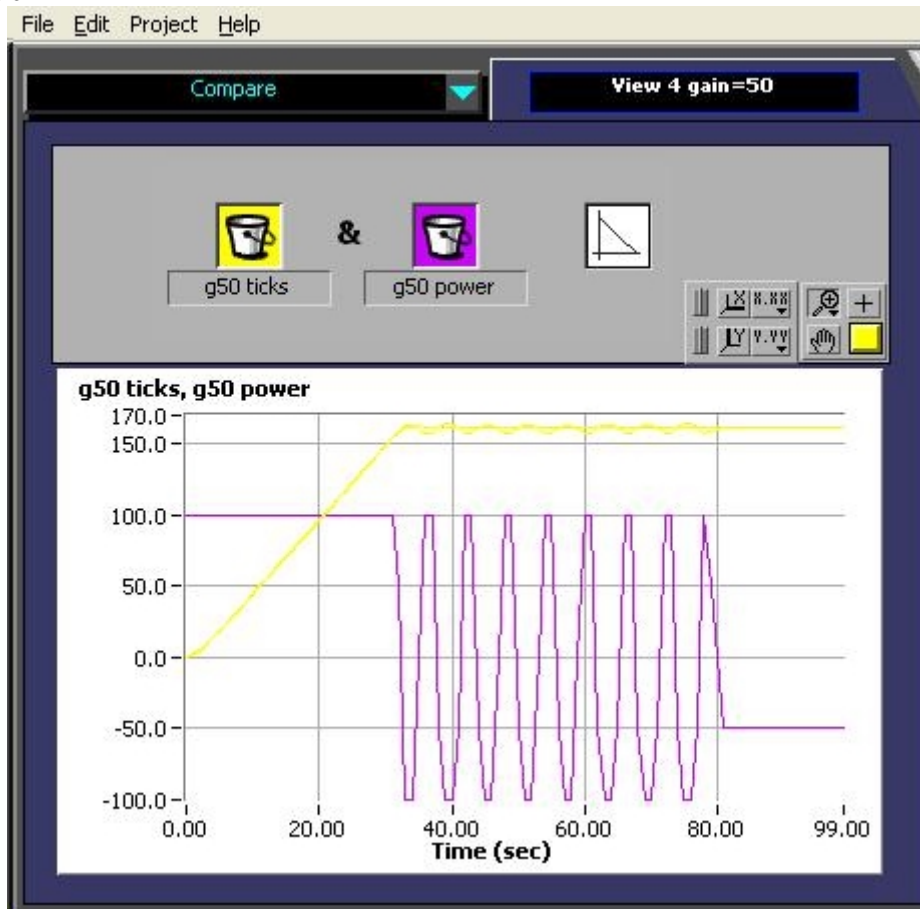


Fig 7. Gain =50

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

Extensions

1. After the wheel has come to rest but while data is still being logged, try to turn the wheel by hand. You should feel the motor responding.
2. What do you notice as the proportional gain is increased? The response should be snappier.

Follow-up

The children could then go on to build and program a LEGO model car that will travel a precise distance using proportional control (Fig 8). Students in higher school levels could be introduced to the other elements of PID control, namely the derivative and integral components. They would then realise why they have to study all that differentiation and integration stuff in school.

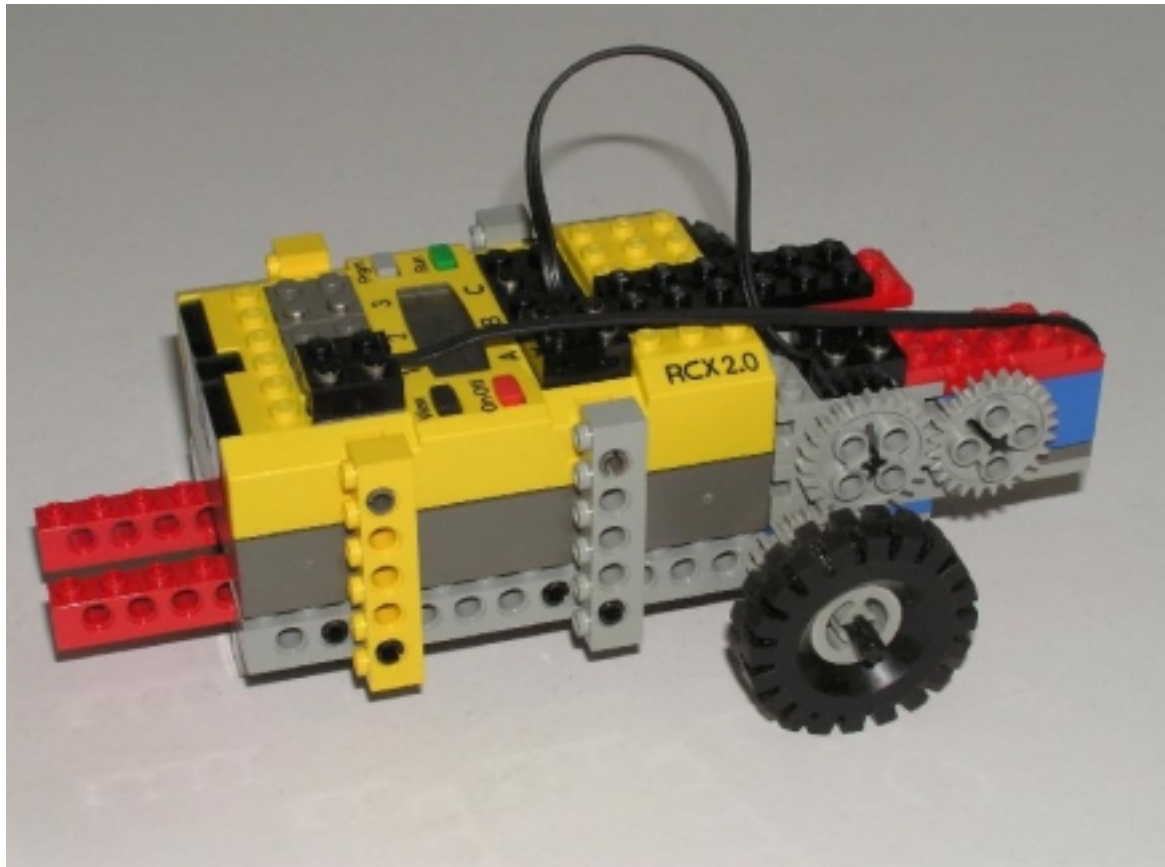


Fig 8. LEGO Model Car using proportional control

Discussion and Conclusions

It is now feasible to teach school children proportional control without them having to learn a new and complex software like BrickOS or Interactive C. Using ROBO LAB 2.9, children can experiment with proportional control the same way as undergraduates in MIT and other learned institutions are doing.

They can then apply it to program a model car to travel a precise distance. Other applications would include a proportional control line follower and maybe even a two-wheeled balancing robot.