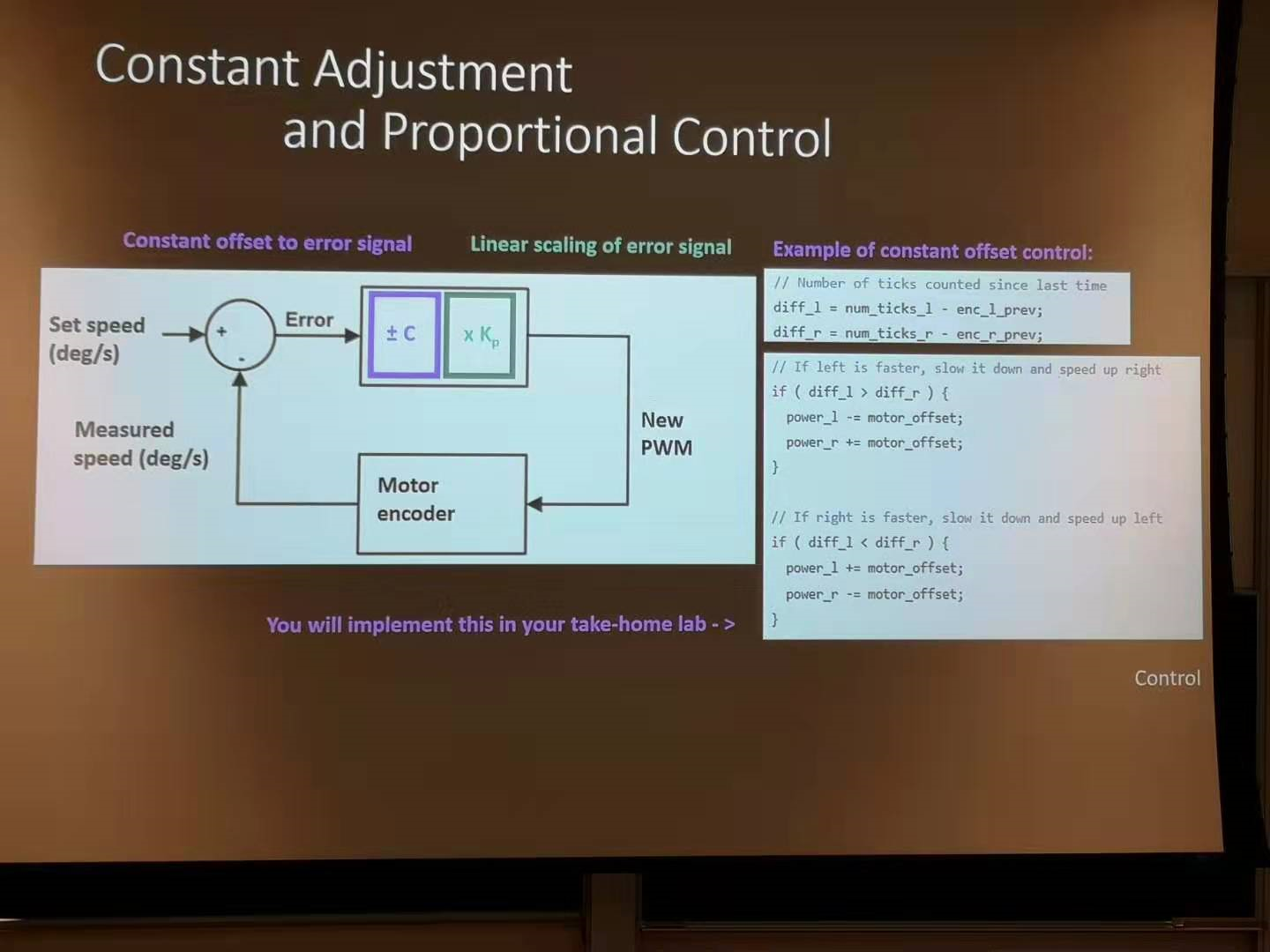
Workshop 4 recap

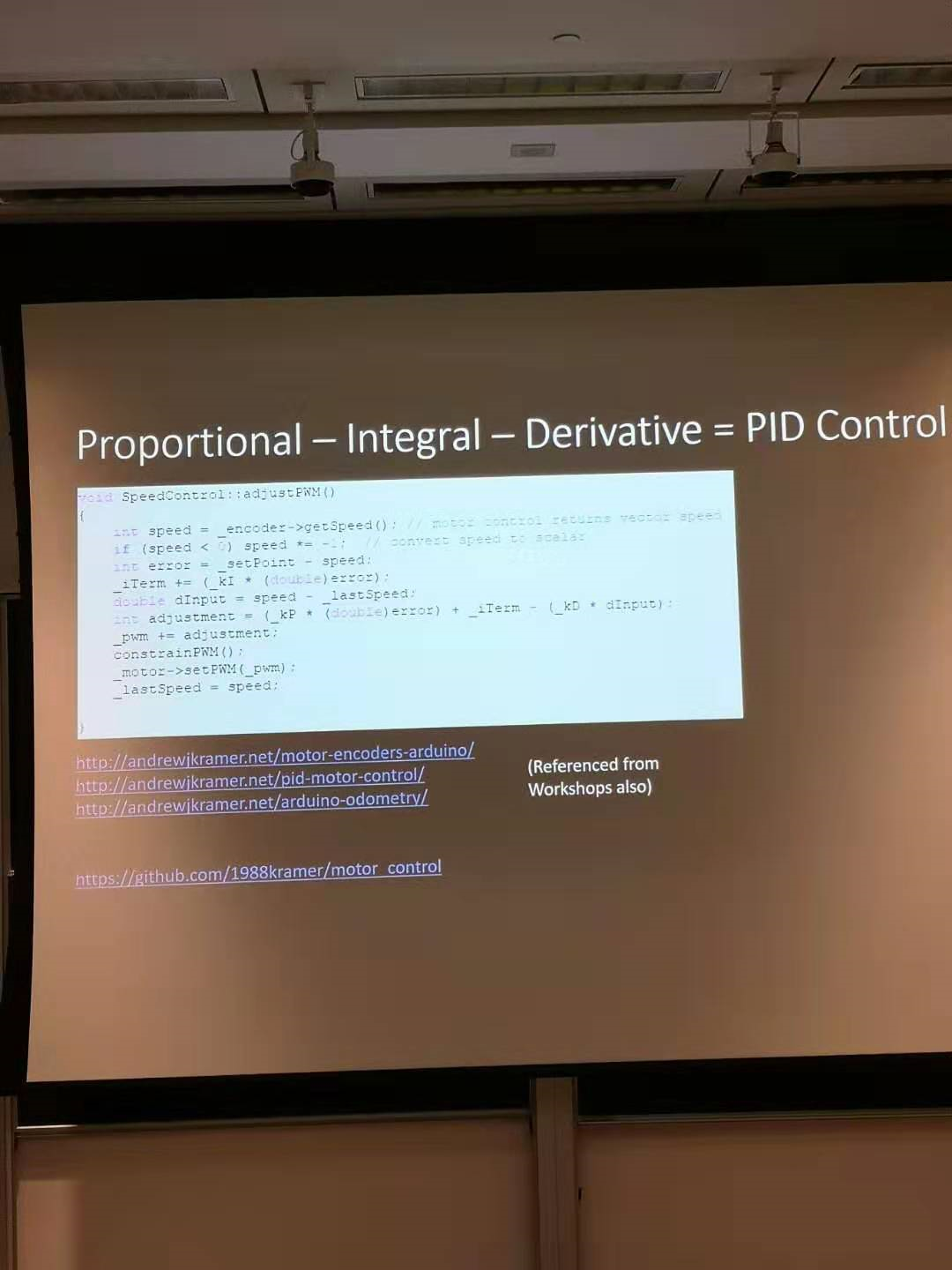
* Arduino Uno → LN298 Motor Driver → GM25-370CA DC Motor + Encoder → Arduino Uno
* Unable to travel in straight line: wheels not aligned on-axis, motors wound slightly differently, weight distribution of robot not exactly equal, friction on each wheel is different, motors are wound slight differently (internal impedance not equal) → unequal motor response

To fix it

* Use hardcoded compensation OR dead reckoning (getting the function of PWM setting vs. # of encoder ticks in 1s)
  + Adjust PWM to either side until encode counts are equal (tune PWM to different motors)
* Closed-loop control (modelling the interaction between actuator and the controller): motor encoder measures error from set speed (deg/s), controller changes to PWM and set a new PWM for the motor
* Constant adjustment and proportional control
  + Need to be careful about the timing of the control system (this should go in the main while loop)
  + With additional tasks executing at the same time, should design a state machine and set priorities of each task
  + Problems: same error may require different correction (fast-moving robot vs slow moving robot causes momentum difference), large error take a long time to correct, can overshoot our desired point

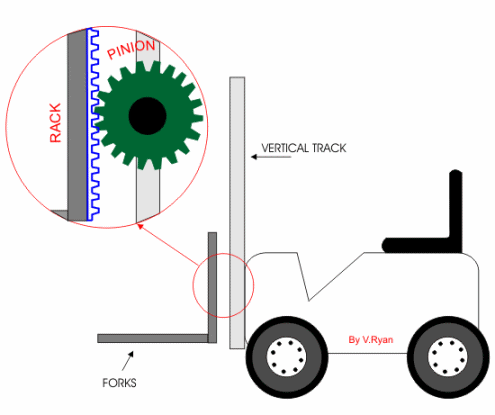


* Proportional - Integral - Derivative = PID Control, summing up past values and taking into account of rate of change, add these terms into the final correction



Arduino code

#include <TimerOne.h> ← use the builtin timer inside the UNO



Push buttons and mechanical switches to stop the component