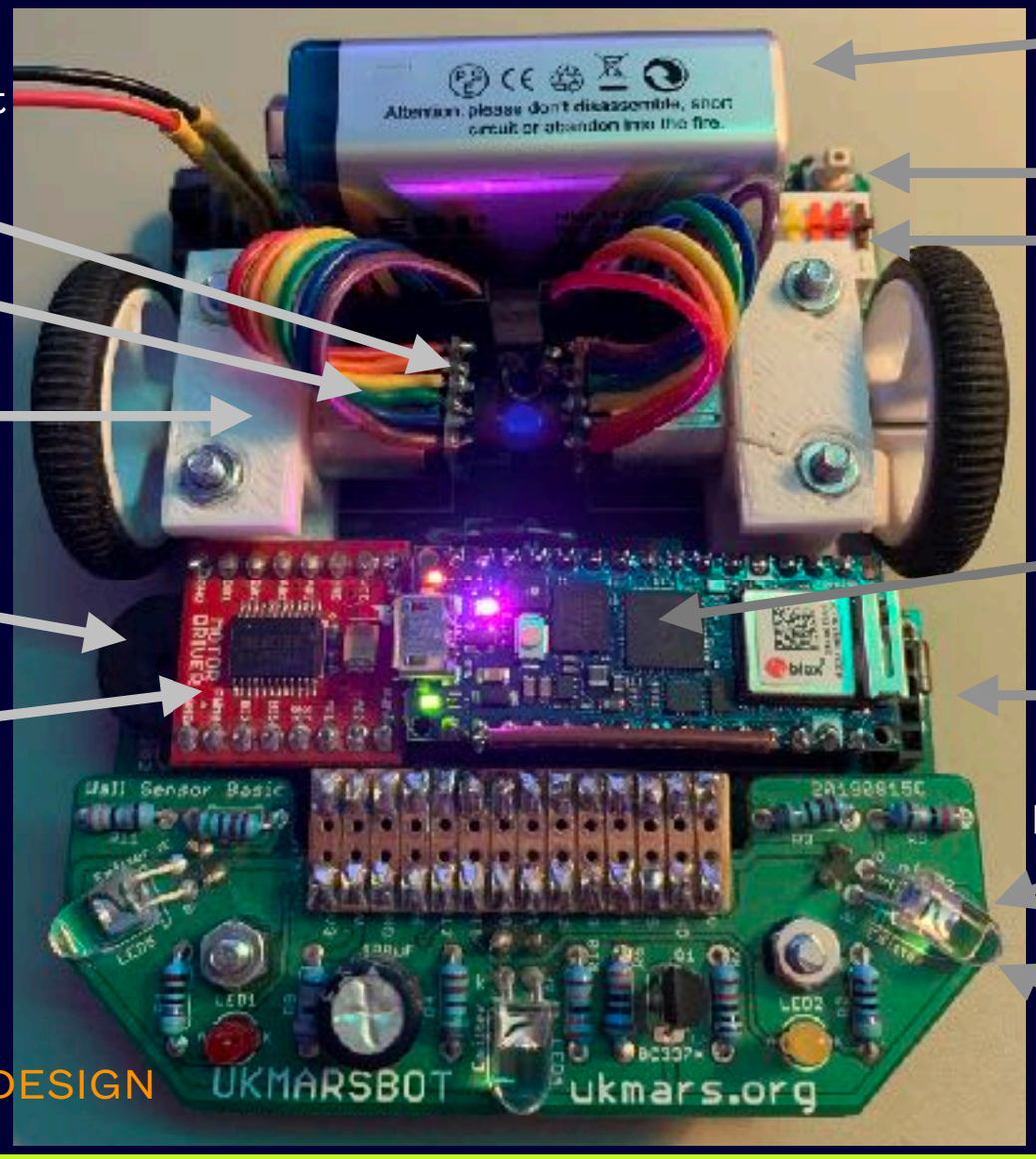


Magnetic Encoder on Motor Shaft

Motor

20:1 Gearbox

Buzzer

Motor Driver

Battery: LiPo or NiMH

Button

DIP Switch

CPU / MCU

Serial Connection

LED (emitter) - IR or Visible Light

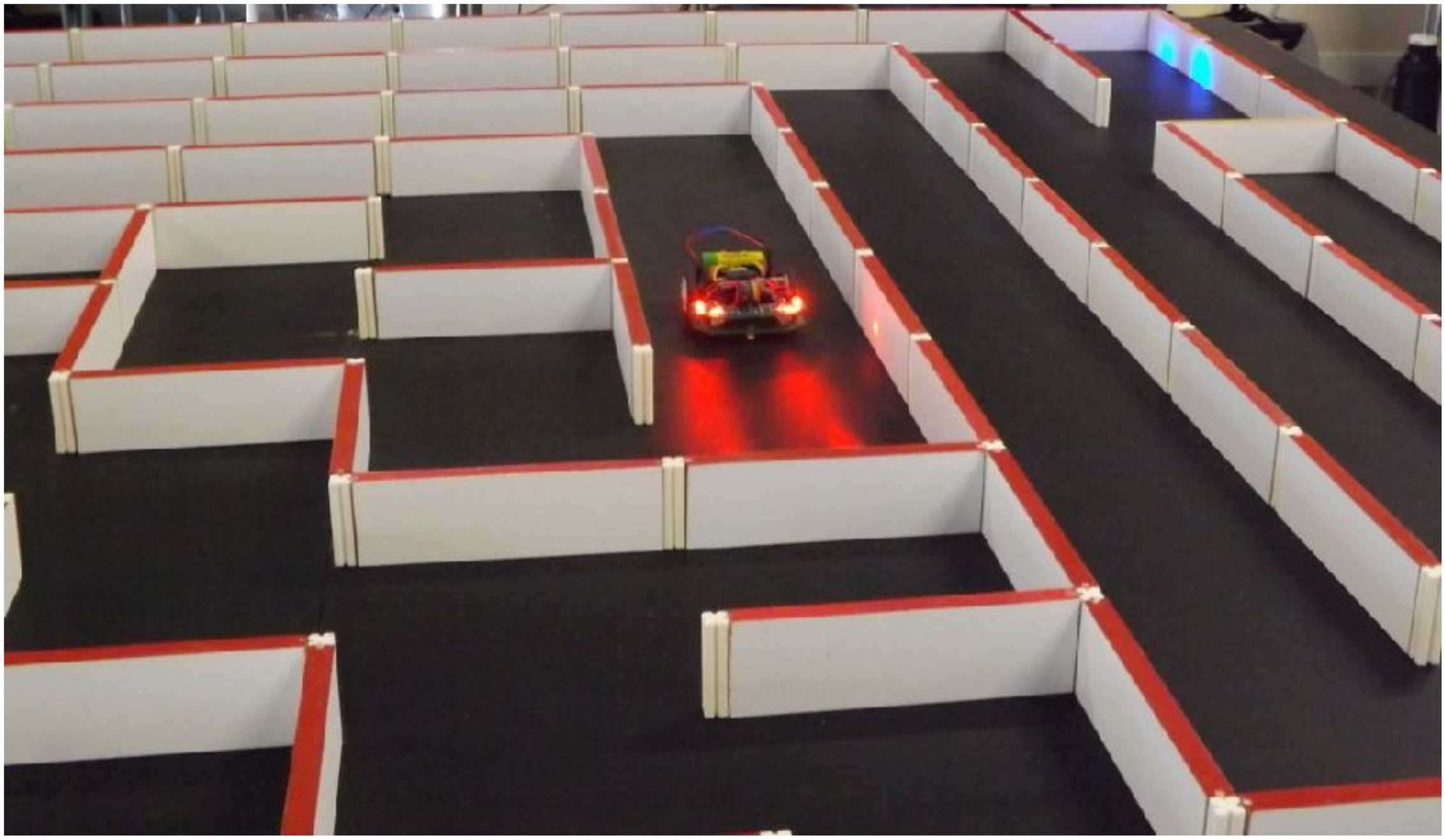
PhotoTransistors or PhotoDiodes

OPEN SOURCE DESIGN

Introduction to Control for Micromouse Maze Robots

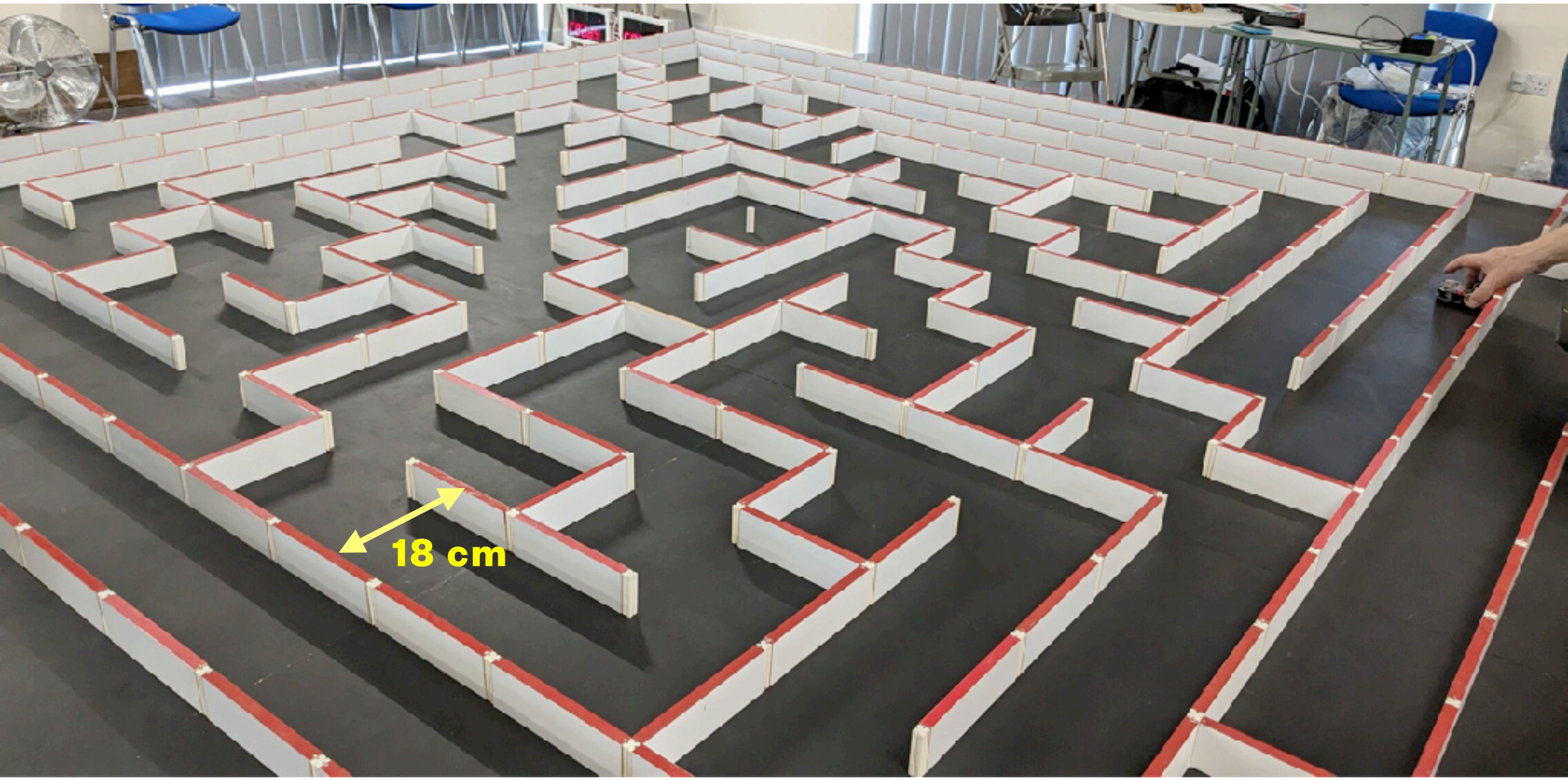
ROB PROBIN, JULY 2023, FORTH2020

Micromouse



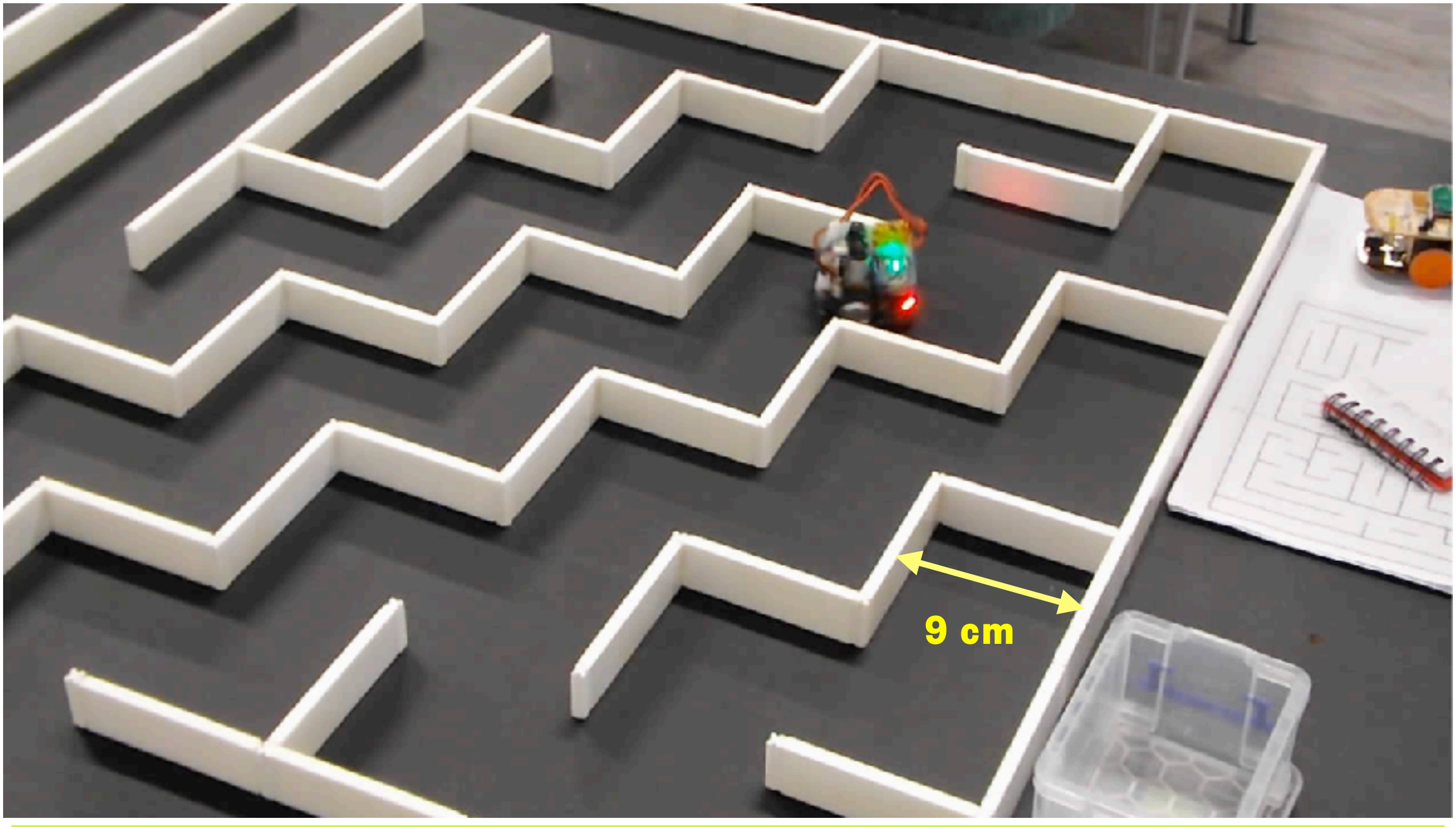
ROB PROBIN, JULY 2023, FORTH2020

Full-size Micromouse



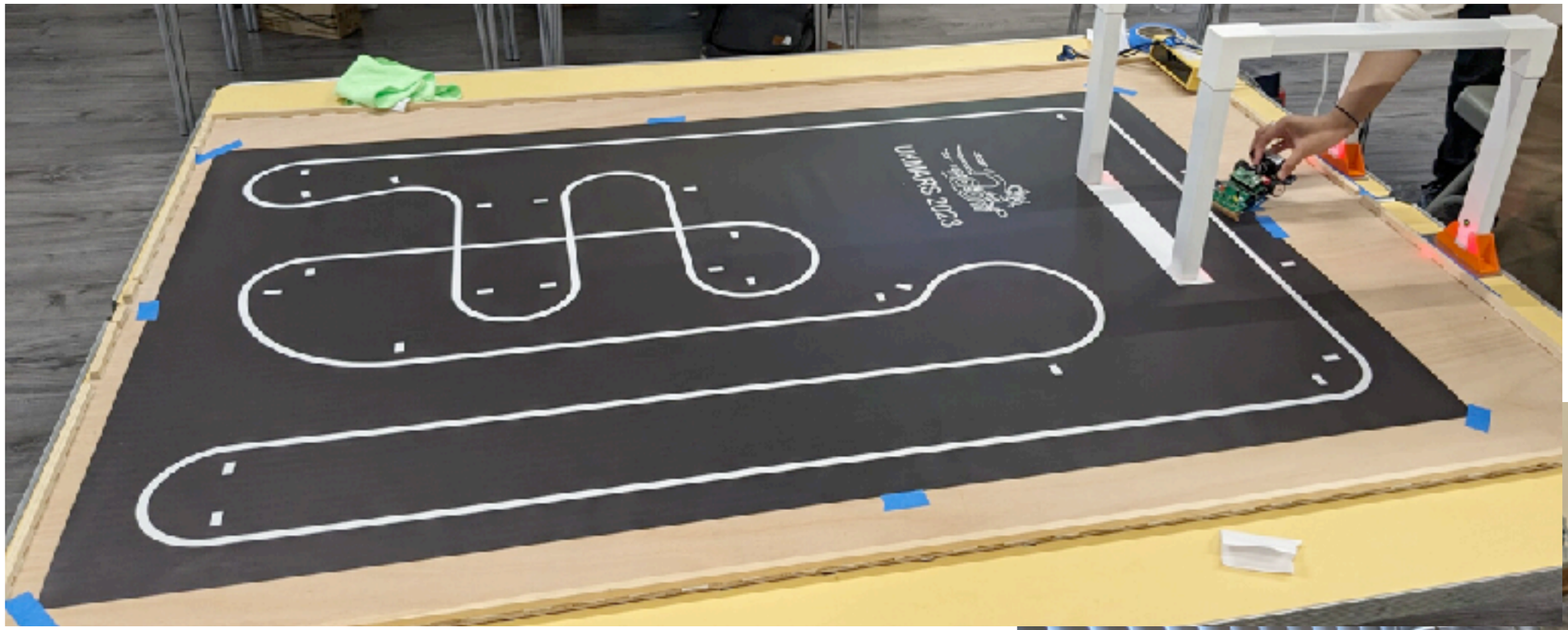
ROB PROBIN, JULY 2023, FORTH2020

Half-size Micromouse



ROB PROBIN, JULY 2023, FORTH2020

Line follower
(Half-size)



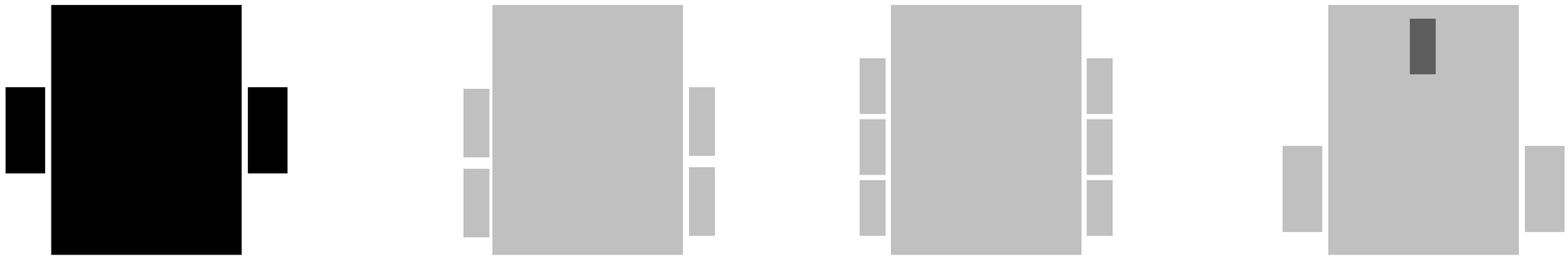
Drag Race

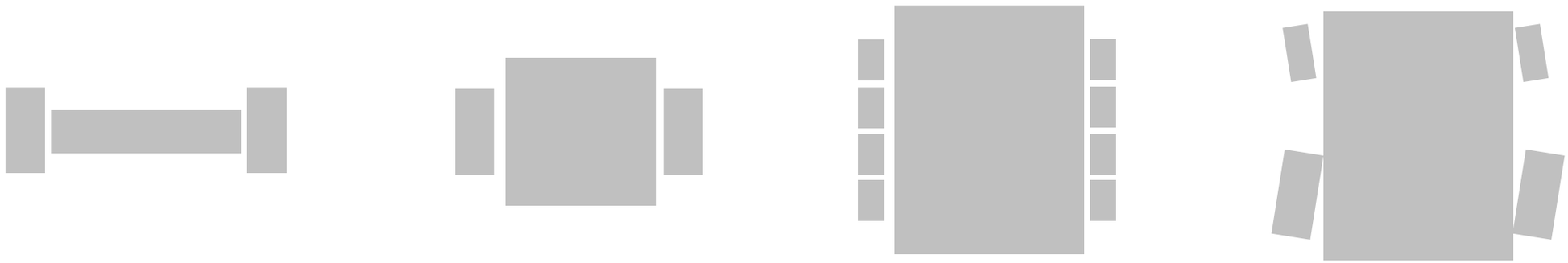
Similar control, but not covered here

Types of Control

- Steering (Wall tracking - left and right)
- Linear speed (mouse speed)
- Angular Speed (Rotation, Turning)
- Linear acceleration
- PID Feedback Control (or more likely PD)
- Feed-forward
- Forth Controller Maths
- Not covered today:
  - Maze solving, Diagonals, Explore vs. fast run, wheel slip, encoders, sensors, odometry error correction, use of accelerometers and gyroscopes, reflectivity of walls, putting it together, getting to position in target cell …

Mouse Geometry

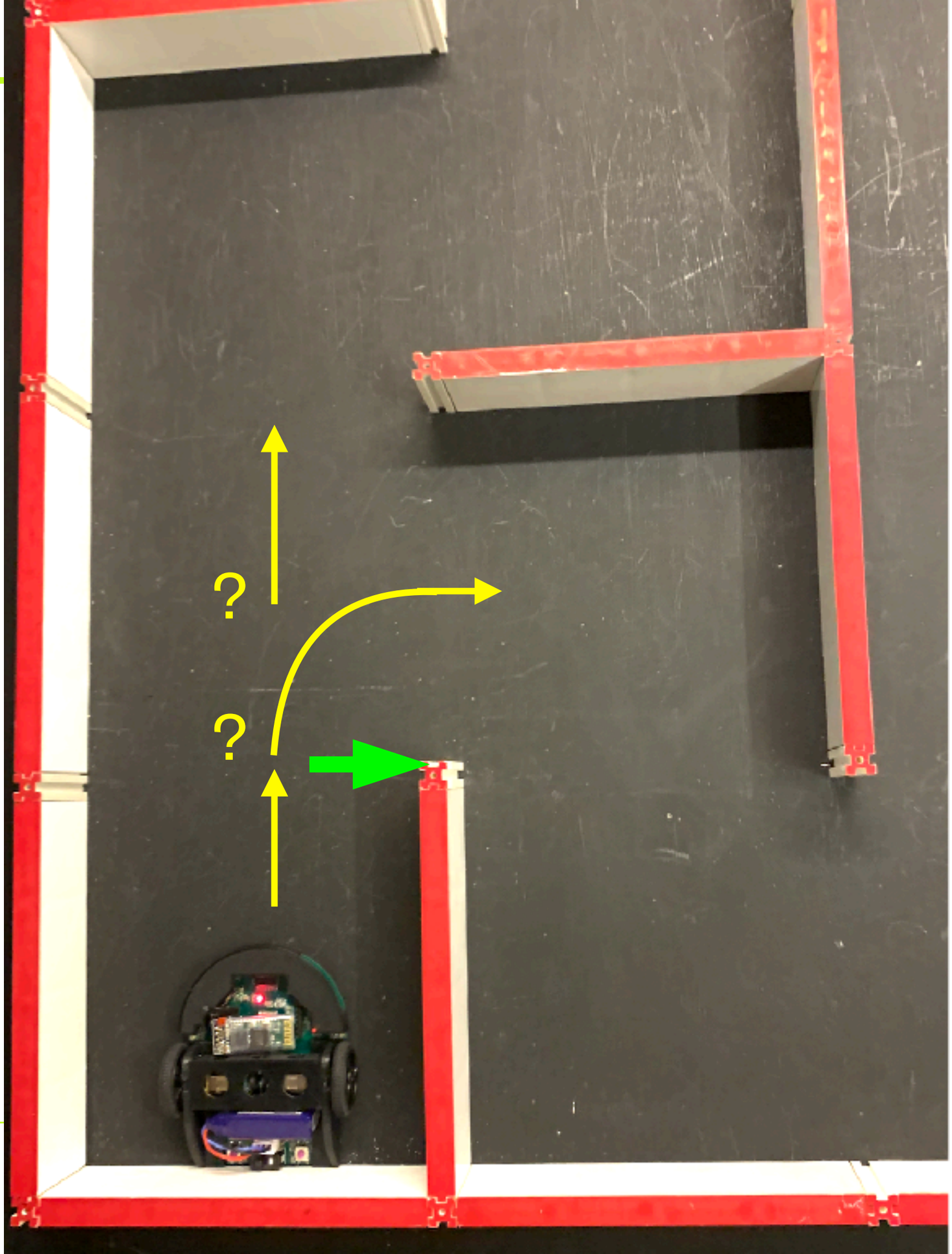




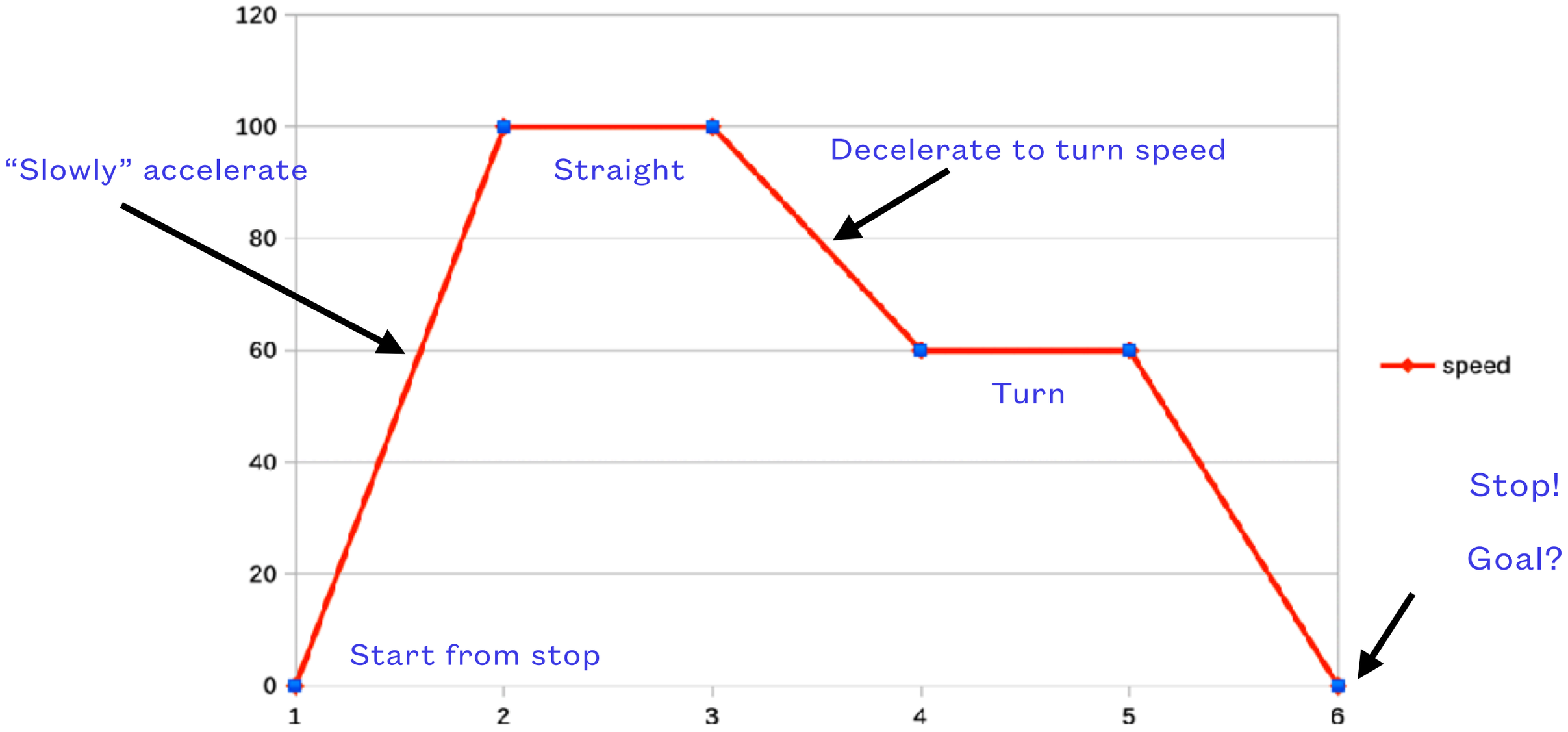
Movement

- Sense wall disappearing left or right, or a wall ahead

- Don't ***really*** want to stop in each square…
  - so motors never stop…



Acceleration Profile



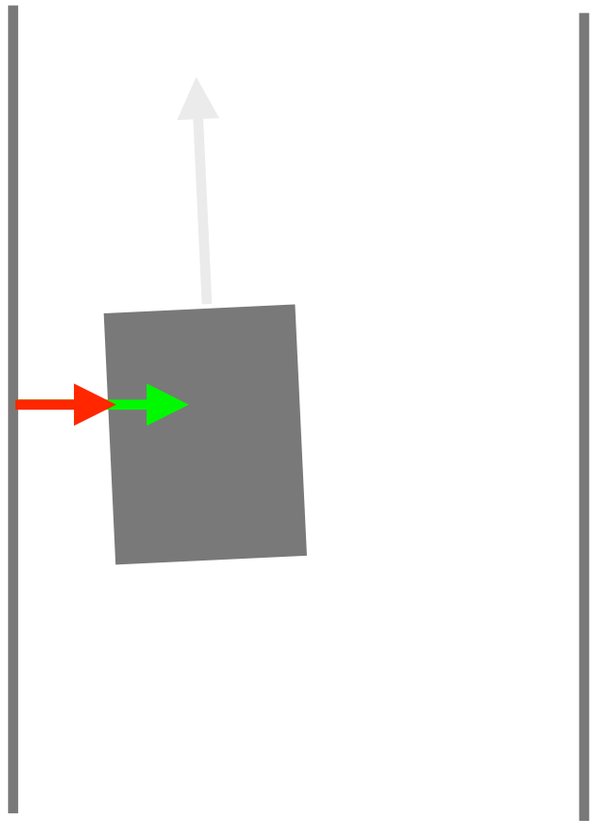
Acceleration (Notes)

- *Applying maximum voltage to motor terminals is not going to work :-)*
- Static friction, Kinetic friction, Inertia or momentum
  - NOTE: Bigger mass requires more energy to accelerate/decelerate and increases inertia = BAD
    - Extra downforce is not compensation!
- For a given surface and down force (gravity), you are likely to only be able to apply so much **torque** to the wheels before wheel slip becomes a problem.
  - Motor torque T $\propto$ Current I (NOT voltage)
- But … controlling the voltage is easier
  - So limit the acceleration (which is caused by the motor torque)
- Applying a constant maximum acceleration to a speed change is easy and results in reasonable performance…
  - But often surface (e.g. dust) on mazes is a big problem for high speed mice…
  - World-class mice require extra downforce, e.g. suction fans, can give an extra 200g of downforce without increasing inertia

Maintaining Control: Batteries

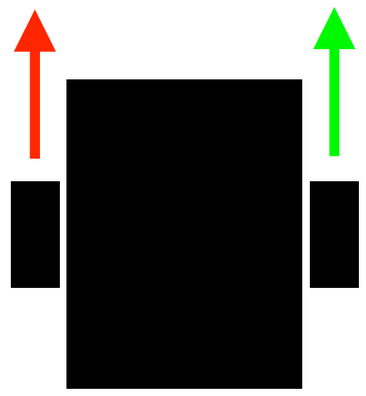
- LiPo: anywhere from about 4.2 V to about 2.7–3.0 V
- Don’t use raw PWM, scale!
  - PWM = V_bat / V_target
  - Target Voltage
  - Set max target voltage → 9v battery → maybe 6v?

PID Steering



- Sensor Inputs - approx. distance from walls (vs. desired)
  - Ignore missing walls, and maybe turn off steering during turns …
- Need to create an angular rotation, or rather an **angular error**
  - *add into rotation error (desired vs. actual)*
- Proportional term - reasonable - further off means more compensation
- Derivative term - probably small - avoiding overshoot, decrease settling
- Integral term - ZERO - no small long term offsets are worth worrying about
- Potentially limit angular error to avoid over correction

Linear vs. Angular?



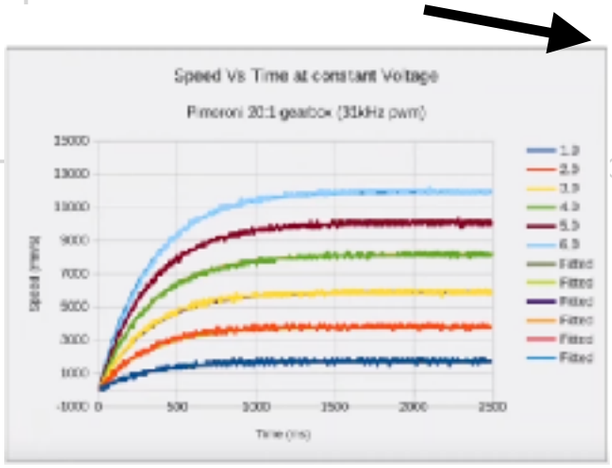
- Speed motor A, speed motor B ???

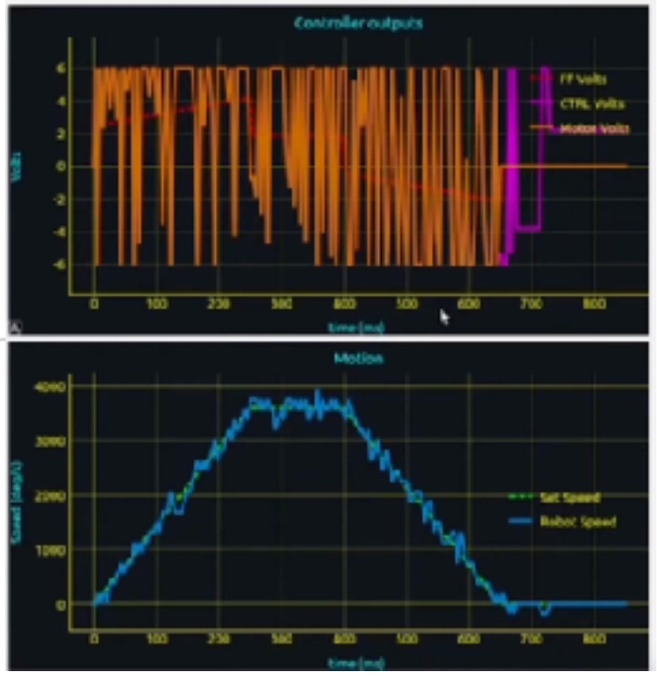
- But turns and linear motion (mouse speed) have:
  - Different requirements, constraints, forces
  - Easier to think?

PID tuning

$$U(t) = Kp\ e(t) + \frac{Kd\ (e(t) - e(t-1))}{dt}$$

- Experimentally is hard to design a good controller…
  - Domain specific, trial and error can give a ok speed performance but bad
- What can we measure?
  - Measure distance, speed, time, motor drive input → per
  - Open loop speed per V
    - Open loop speed / time → 63% point → Tm
    - Final Speed → System Gain Km





- **Mouse and the feedback system → 2nd order system appro**x. (motors are a 1st order system)
  - Damping Ratio - maybe 0.707 - slightly faster than critical damping get you within 5% most quickly)
  - Natural frequency (perhaps we want to use the same order as the time constant - try more or less)
- We can approximate Kp and Kd from these mathematically - which gives us reasonable results.
  - Remember Kd can to be scaled by sampling frequency in real robot, i.e. 1/dt … avoid a divide
- ***Taken from great presentation on this topic by Pete Harrison:***
  - ***https://www.youtube.com/watch?v=qKoPRacXk9Q***

Feed forward

- We know the mass, the motor, the drive chain, the load, the requested and current speed!
  - Can we create a reasonable feed forward system?

- Reduce feed-back to correcting for errors ***only***
  - Not possible to avoid error correction entirely
  - But reduces the effect off a poor PID controller
    - Or rather gives a much better overall controller

Forth PID Maths in Micromouse

- **Fixed point maths - possible.**
  - 32 bit is probably smallest Fixed point you want because of range
    - Still hits edges for some PID values - range management
    - For Flashforth want to build a set of primitive's.
  - For Zeptoforth on 32-bit fixed point, probably ok (I haven't done this yet)
- **Floating point maths - 32 bit**
  - Even ATMega328P (16MHz)
    - Running encoder interrupts at 12/cycle, 20:1, 2 m/s, 30mm wheels = kilohertz range, plus ADC interupts light and dark
      - 2ms / 500Hz controller loop, PID - Avoid division
      - Not much support in Forth's however …so you are probably building your own support.
  - Nice with Cortex-M4F (e.g. STM32F411, black pill)
  - Even without floating point units most new processors don't have a problem (e.g. Raspberry Pi Pico Cortex-M0+)
  - ***Most understandable option!***
- **Integer**
  - Theoretically possible, but harder to think about

Extend with specific numbers & example Forth code

Possible Future Topics
Any interest?

- Examples of Forth PID controller maths,
- Maze solving,
- Speed control by position
- Putting control together,
- Getting to position in target cell,
- Diagonals,
- Computer Vision on robots,
- Competition strategies - Explore vs. fast run (including effect of competition rules).
- Wheel slip
- Encoders - what sorts, performance, requirements
- Sensors - options, performance, requirements,
- Odometry error correction (walls finishing, starting, posts)
- Use of accelerometers and gyroscopes,
- Reflectivity of walls,