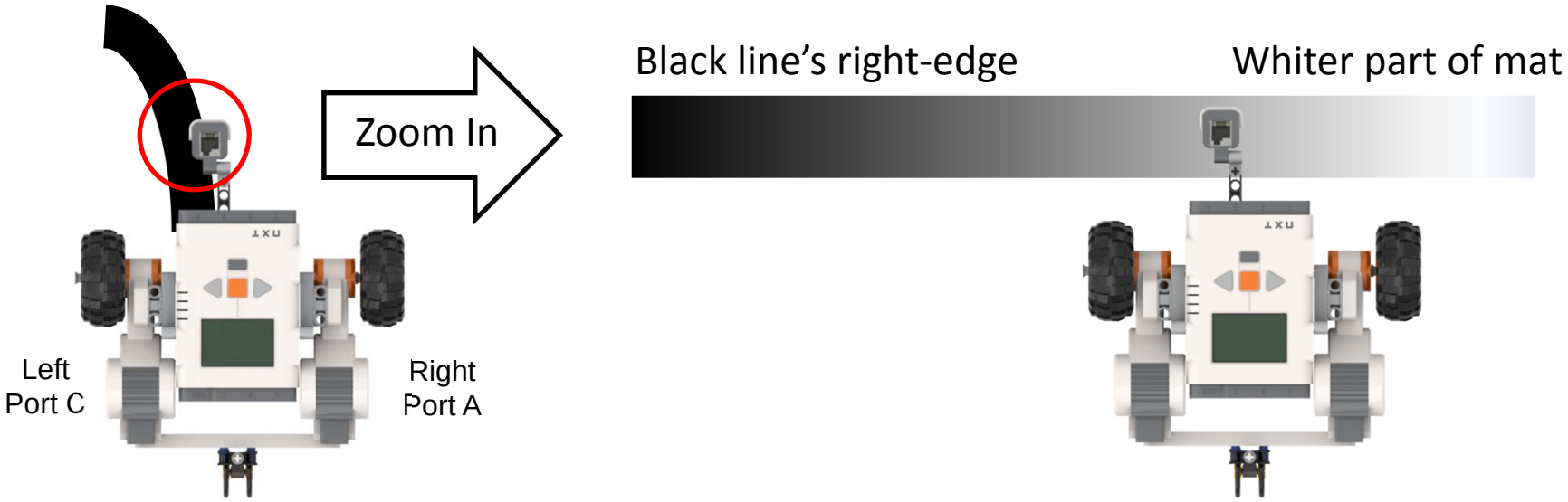
Lego NXT Domabot Line-Following

Bang-Bang and Proportional Control

What's Really Happening



- Light sensor reports intensity (not color)
- Sensor traces outer edge of black line
- Thus, reported values will change as sensor traces line

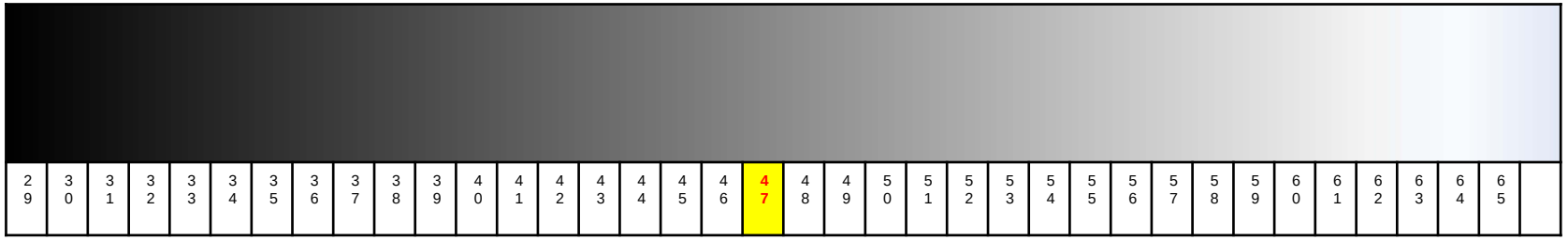
Bang-Bang Control

Approach: Find a threshold value. When irValue > irThresh then yaw CCW (towards black)

Yaw CW (thus towards whiter part) by moving Left Motor (C) and Stopping Right Motor (A)

Yaw CCW (thus towards darker part) by moving Right Motor (A) and Stopping Left Motor (C)

Suppose



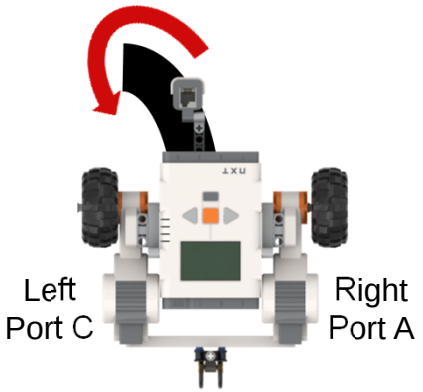
Variable	Value	Meaning
$irMin$	27	Darkest intensity value
$irMax$	65	Brightest intensity value
speedBase	50	Domabot base speed

$$\text{irThresh} \triangleq \frac{\text{irMax}-\text{irMin}}{2} = \frac{65+29}{2} = 47$$

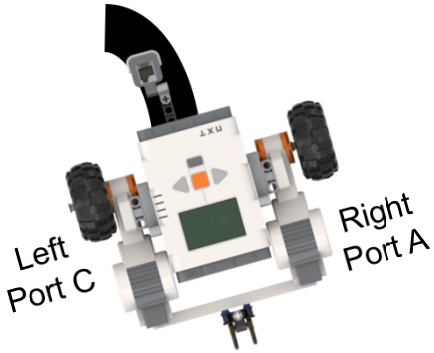
Pseudocode for calibrating and computing `irThresh`

```
Set timer for 2 seconds;
// Step 1 - Yaw CCW slowly
Forward Right Motor slowly;
Reverse Left Motor slowly;
While (current time < 2 seconds){
  Read irValue; // read light sensor value
  if irValue > irMax
     irMax = irValue;
  if irValue < irMin
     irMin = irValue;
} // end while

Set timer for 3 seconds;
// Step 2 - Yaw CW slowly
Forward Left Motor slowly;
Reverse Right Motor slowly;
While (current time < 3 seconds){
  Read irValue; // read light sensor value
  if irValue > irMax
     irMax = irValue;
  if irValue < irMin
     irMin = irValue;
} // end while

// Yawing CW and CCW results in min and max values
// Step 3 – Display values and calculations
irThresh = (irMin + irMax) / 2; // i.e. mid-point
Display min, max, and threshold values
Hit Right Arrow to continue
```

Step 1:
- Sensor on outermost part of black line i.e. (A)
- Start yawing CCW

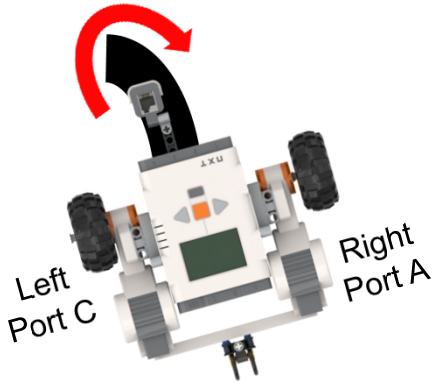


(A) Initial position

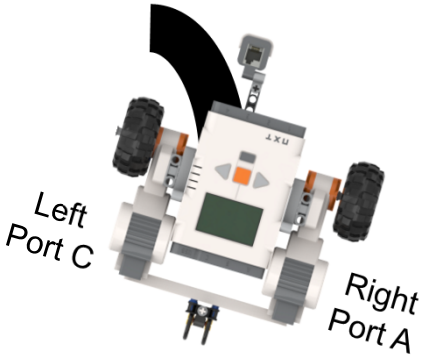


(B) CCW Yaw finished

Step 2:
- Now sensor somewhere on black line i.e. (B)
- Start yawing CW i.e. (C)



(B) CCW Yaw finished



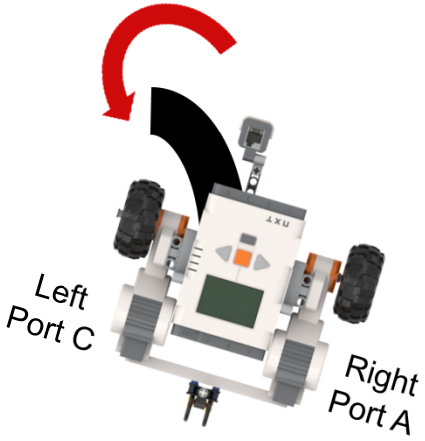
(C) CW Yaw finished

Pseudocode for bang-bang line following

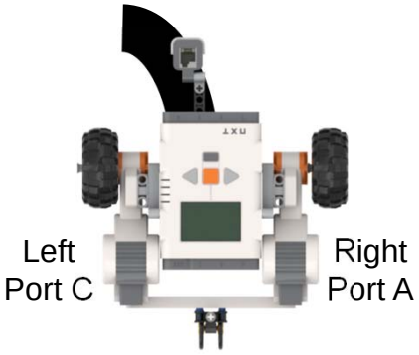
```
Set timer for 2 seconds;
// Step 4 – Domabot may have yawed onto white
// Yaw CCW again, towards outermost black edge
Forward Right Motor slowly;
Reverse Left Motor slowly;
Read irValue;
While (irValue > irThresh){ // still in whitish area
  Read irValue; // read light sensor value
} // end while

// Step 5 – Domabot back on outermost edge of black line
// Start Bang-Bang line following
do {
  Check if user hit left arrow button to quit;
  Read irValue;
  if irValue > irThresh { // then sensor on whitish part
     speed = speedBase;   // so right motor forward
     Forward Right Motor at speed;
     Stop Left Motor;
  };

 if irValue <= irThresh { // then sensor on blackish part
     speed = speedBase;   // so left motor forward
     Forward Left Motor at speed;
     Stop Right Motor;
  };
} while (left arrow button not pushed)

// Step 6 – User hit Left Arrow Button
Turn off Left and Right motors;
Play Sound
Stop all tasks
```

Step 3:
- Display results

Step 4:
- Sensor might have yawed onto white area i.e. (D)
- Start yawing CCW until at outermost black edge i.e. (E)



(D) Start yawing CCW

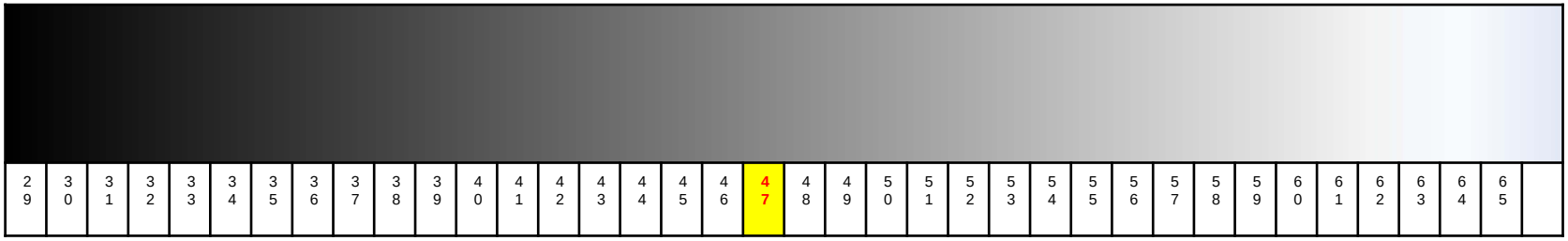


(E) Domabot at edge
Begin Domabot bang-bang

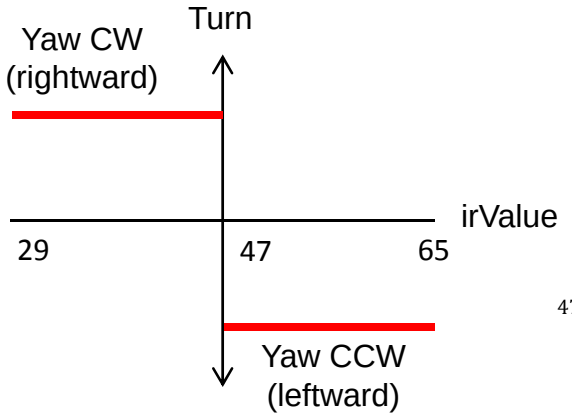
Proportional Control

Yaw CW (thus towards whiter part) by moving Left Motor (C) and Stopping Right Motor (A)

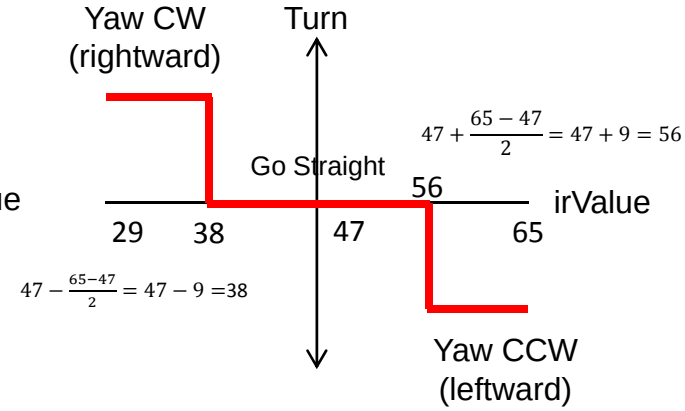
Yaw CCW (thus towards darker part) by moving Right Motor (A) and Stopping Left Motor (C)



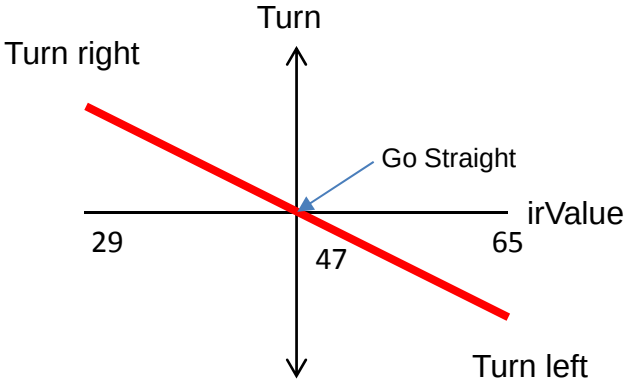
$$\text{irThresh} = \frac{\text{irMax} - \text{irMin}}{2} = \frac{65 + 29}{2} = 47$$



Bang-Bang: 2 regions
(Yaw CW or CCW)

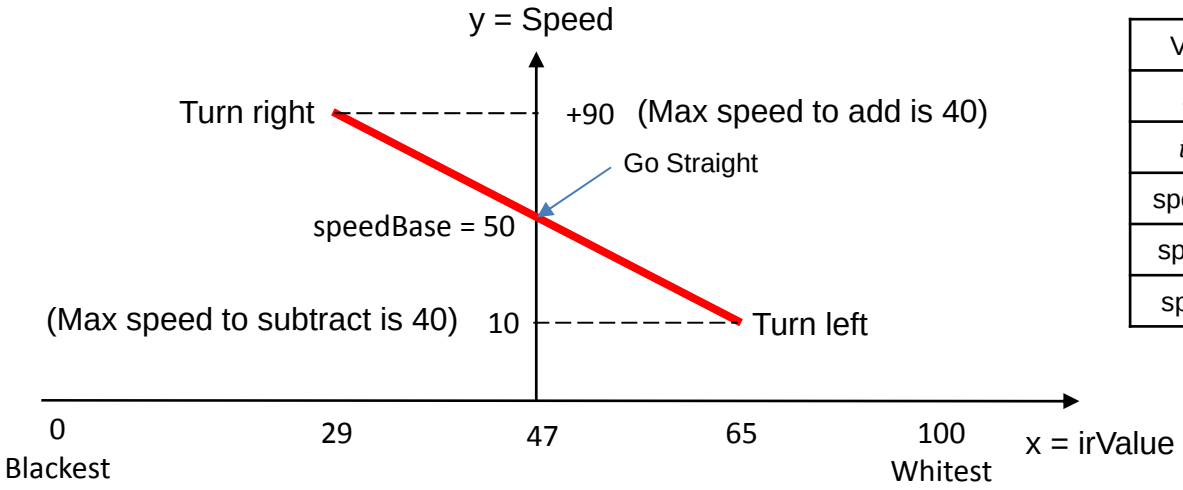


3 regions
(Yaw CW, Go Straight, Yaw CCW)



Proportional: Infinite Regions

Normalize the graph



Variable	Value	Meaning
$irMin$	27	Darkest intensity value
$irMax$	65	Brightest intensity value
speedBase	50	Domabot base speed
speedMax	90	Domabot max speed
speedMin	10	Domabot min speed

$$\text{Slope } m = \frac{10-(90)}{65-29} = \frac{-80}{36} = -2.22$$

Sanity Check:

1. $y(29) = 50 + \frac{-80}{36}(29) = -64.4 + 104.4 = +40$
2. $y(65) = \frac{-80}{36}(65) = -144.4 + 104.4 = -40$
3. $y(47) = \frac{-80}{36}(47) = -104.4 + 104.4 = 0$

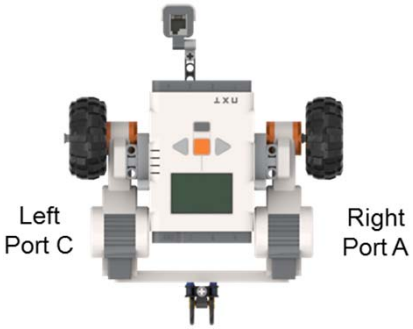
Equation of a Line is $y(x) = mx + b$ thus $b = y(x) - mx$ Hence b $= 50 - (-2.22 \cdot 47) = 50 + 104.34 = 154.34$

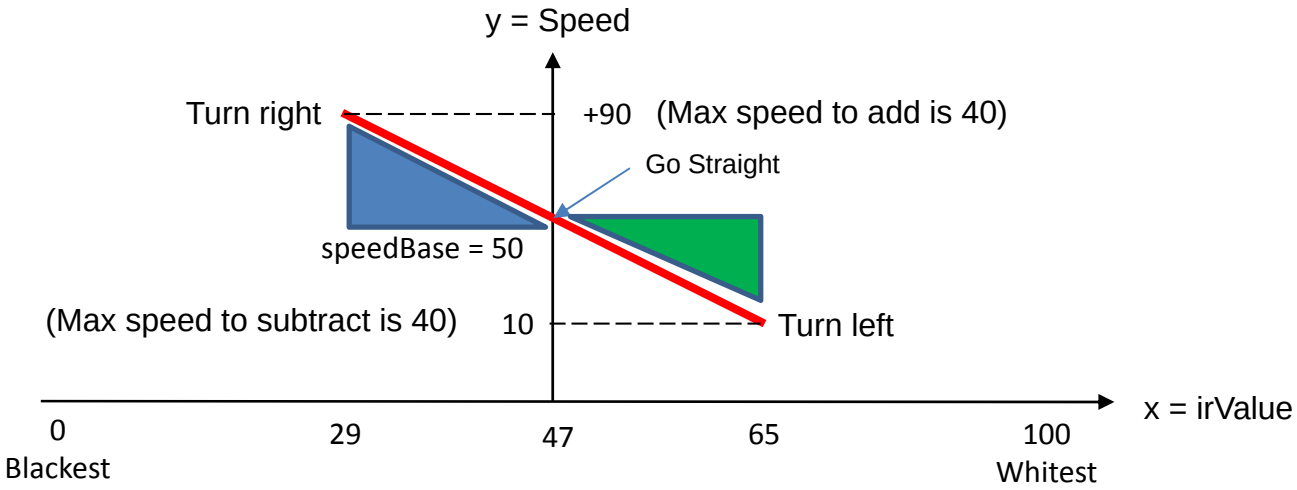
So $y(x) = -2.22x + 154.34$

Sanity Check:

1. $y(29) = -2.22(29) + 154.34 = -64.38 + 153.4 = 89.96 \approx 90$
2. $y(65) = -2.22(65) + 154.34 = -144.3 + 153.4 = 9.1 \approx 10$
3. $y(47) = -2.22(47) + 154.34 = -104.34 + 153.4 = 49.06 \approx 50$

Formulate P-control law





Call this `irError`

(1-1). speed for Motor C (left)= speedBase + gain(irValue − irThresh)=50+2.2(irValue − irThresh)

(1-2). speed for Motor A (right)= speedBase − gain(irValue − irThresh)=50−2.2(irValue − irThresh)

Blue area: yaw CW (away from black): Increase Motor C speed and Decrease Motor A speed

Sanity Check: let irValue=40 then

$$\text{speed C}=50+(-2.22)(40-47)=50+(-2.22)(-7)=50+15.54=65.54$$

$$\text{speed A}=50-(-2.22)(40-47)=50+(2.22)(-7)=50-15.54=34.46$$

Results in yawing CW (QED)

Green area: yaw CCW (toward black): Decrease Motor C speed and Increase Motor A speed

Sanity Check: let irValue=55 then

$$\text{speed A}=50-(-2.22)(55-47)=50+(2.22)(8)=50+17.76=67.76$$

$$\text{speed C}=50+(-2.22)(55-47)=50+(-2.22)(8)=50-17.76=32.24$$

Results in yawing CCW (QED)

Pseudocode for Proportional Control Line Following

```
// Calculate proportional gain from
// calibrated values irMax and irMin
// and defined speedMin and speedMax values
kP = (speedMin - speedMax) / (irMax - irMin);

Do {
  Check if Left Arrow Button pushed;
  Read irValue;
  irError = irValue – irThresh;
  // On whiter part if irError > 0
  // So increase Motor C to yaw towards black
  speed = speedBase + kP * irError;
  // NB: speed now > speedBase
  if speed > speedMax
    speed = speedMax;
  if speed < speedMin
    speed = speedMin;
  Forward Motor C at speed;
  // And decrease Motor A to yaw towards black
  speed = speedBase – kP * irError;
  // NB: speed now < speedBase
  if speed > speedMax
    speed = speedMax;
  if speed < speedMin
    speed = speedMin;
  Forward Motor A at speed;
} while Left Arrow Button not pushed

// Left Arrow Button was pushed
// Exit gracefully
Turn off both motors;
Play Exit sound
Stop All Tasks
```

Recall Equations (1.1) and (1.2) from before
Also kP is the slope calculated from calibration