Control Systems

6.270
January 2013

Pointer review

- Variable: uint8_t var = 8;
- Pointer: uint8_t *ptr;
- Address-of: ptr = &var;
- Dereference: printf(“Var is: %d\n”, *ptr);
- Dereference and change: *ptr++;

One way to drive forwards...

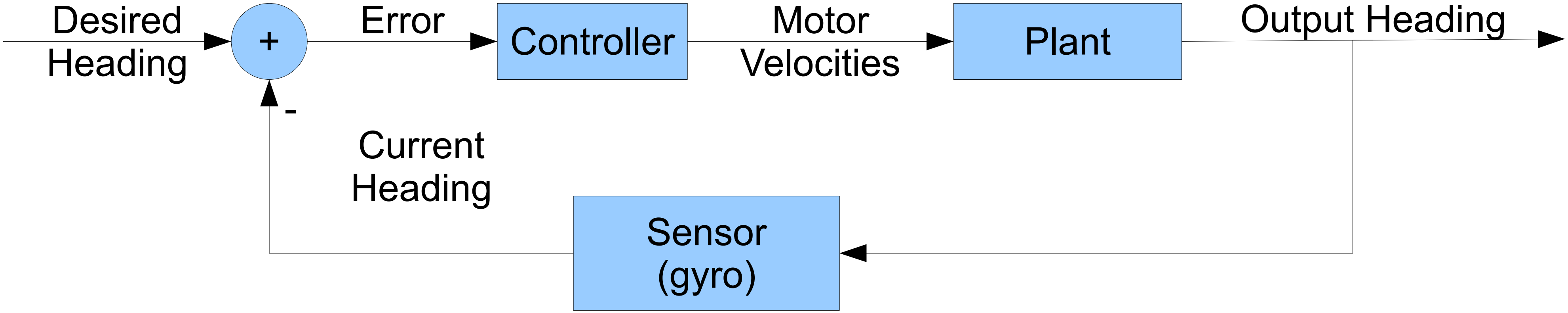
```
int usetup() {  //set up our robot
  gyro_init(11, 1400000, 1000);  //start the gyroscope
   return 0;
}

int umain() {  //main program
  while (1) { //do this over and over
    float deg = gyro_get_degrees();//what angle are we facing?
    if (deg < 0) {
      motor_set_vel(0,40);  //turn left if we are to the right
      motor_set_vel(1,90);
    } else {
      motor_set_vel(0,90);  //turn right if we are to the left
      motor_set_vel(1,40);
    }
  }
  return 0;
}
```

We can do better than that

- With binary feedback:
  - Oversteering
  - Jerky
  - Not how you would drive a car
- What if we adjust based on amount of error?
  - Larger error → larger adjustment
  - Proportional Control!

Proportional Control



Proportional Control

- Let's write a proportional controller!
- Demo!

Some notes about the gyro

- gyro_get_degrees() gives absolute heading with reference to starting position
- i.e. if you rotate CCW twice, gyro_get_degrees() returns 720
- Probably want helper function to calculate heading error better
  - e.g. take heading mod 360
  - e.g. error should never be > 180 or < -180
- Calibrate it before using!

Problems with Proportional Control

- Bias – never reach desired value
- Oscillations

PID Control

- Proportional
  - Handles majority of correction
- Integral
  - Adjusts output based on magnitude and **duration** of error
  - Can reduce bias
- Derivative
  - Adjusts output based on rate of change of error
  - Slows down controller output changes (damping)
  - Can reduce amount of overshooting

Tuning PID Controller

- More complicated than proportional: 3 parameters
- See http://en.wikipedia.org/wiki/PID_controller#Loop_tuning for several tuning methods

Some ideas for driving

- Consider using multiple controllers
  - Heading controller (rotational velocity)
  - Distance controller (forward velocity)
- Update the desired heading as you drive
  - This will be covered tomorrow
- Can robot drive backwards? → maximum heading error is 90 degrees