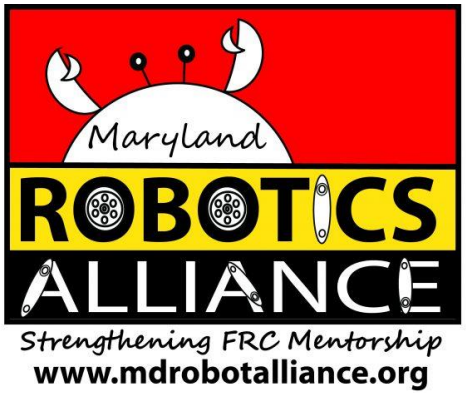


Sensor Integration and Advanced Sensors

Sharon Brackett

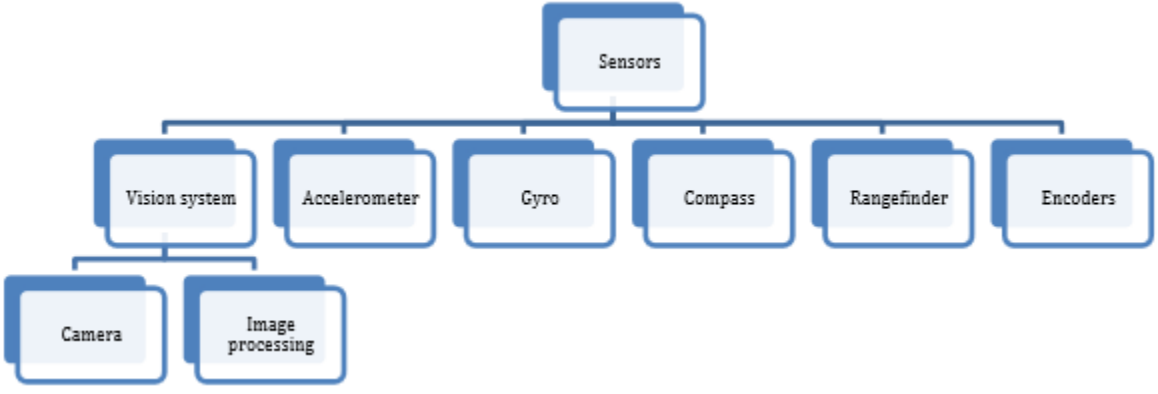
sharon@brackett.com

Me

- Computer Engineer
  - Embedded Systems- Software/Hardware
  - 3D Printing
- Mentor #2537 2008-2014
- Control Systems Advisor 2015-Present

Maryland ROBOTICS ALLIANCE
Strengthening FRC Mentorship
www.mdrobotalliance.org

Sensor Review



Maryland ROBOTICS ALLIANCE
Strengthening FRC Mentorship
www.mdrobotalliance.org

Sensor Inputs

- Digital Inputs
  - Switches
    - Limit Switches
    - Home
    - Counters
  - Encoders
    - Relative
    - Absolute
- Analog sources
  - Pots
  - Rangefinders
- Data Sources
  - RS-232
  - I2C, SPI, CAN

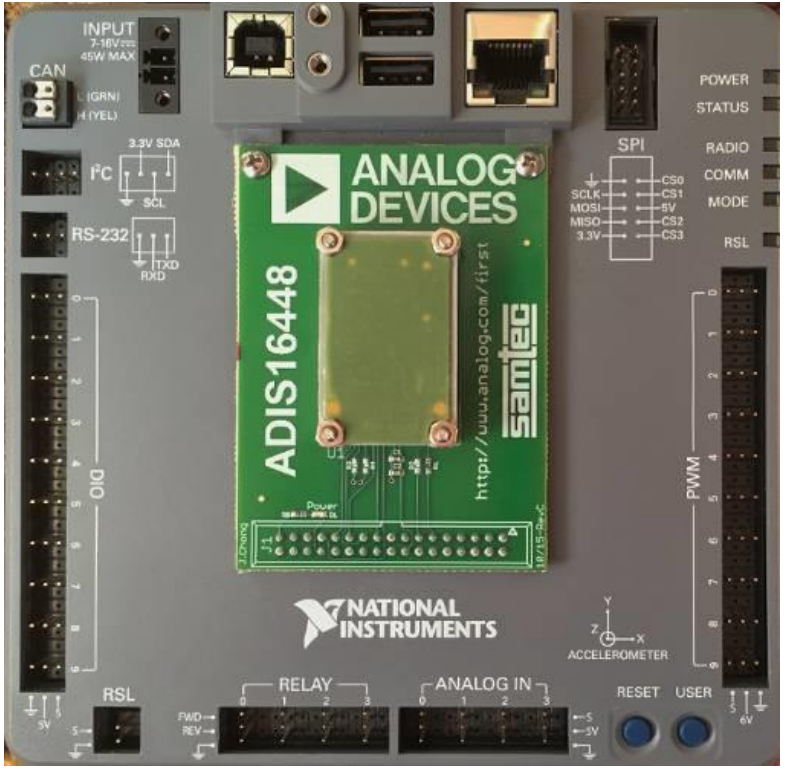
Maryland ROBOTICS ALLIANCE
Strengthening FRC Mentorship
www.mdrobotalliance.org

ADXRS450 Gyro

- Z axis only
- 300˚/sec
- Sits on SPI port on RoboRIO
- SPI-Serial Peripheral Interface

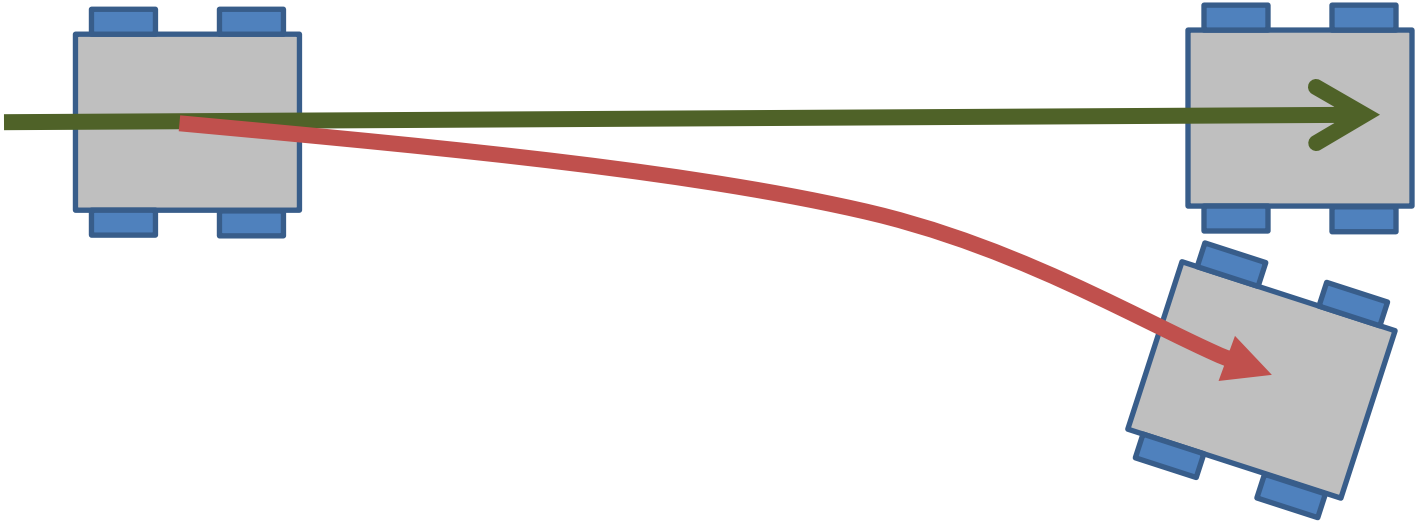
All in one IMU

- ADIS 16448 FRC IMU
  - Gyro
  - Accelerometer
  - Magnetometer
- X, Y, and Z axis
- Barometer
- https://github.com/juchong/ADIS16448-RoboRIO-Driver



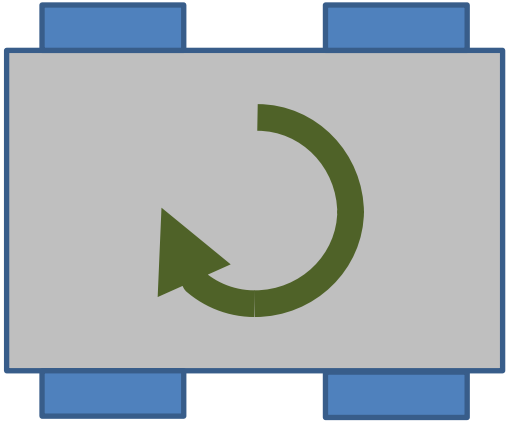
Maryland ROBOTICS ALLIANCE
Strengthening FRC Mentorship
www.mdrobotalliance.org

Drift



The Gyro

- Measures change in angle
- Produces a number
- Has to be zeroed



C++

```
class GyroSample : public SampleRobot
{
    RobotDrive myRobot; // robot drive system
    AnalogGyro gyro;
    static const float kP = 0.03;

public:
    GyroSample():
        myRobot(1, 2), // initialize the sensors
        gyro(1)
    {
        myRobot.SetExpiration(0.1);
    }

    void Autonomous()
    {
        gyro.Reset();
        while (IsAutonomous())
        {
            float angle = gyro.GetAngle();
            myRobot.Drive(-1.0, -angle * kP);
            Wait(0.004);
        }
        myRobot.Drive(0.0, 0.0); // stop robot
    }
};
```

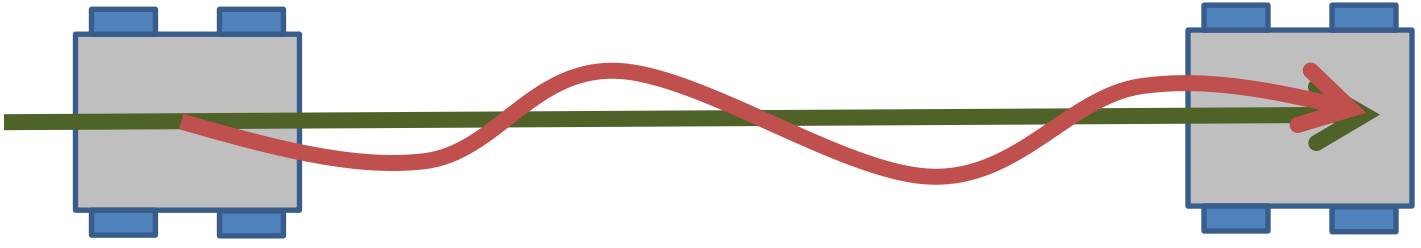
Java

```
package edu.wpi.first.wpilibj.templates;
import edu.wpi.first.wpilibj.AnalogGyro;
import edu.wpi.first.wpilibj.RobotDrive;
import edu.wpi.first.wpilibj.SampleRobot;
import edu.wpi.first.wpilibj.Timer;
public class GyroSample extends SampleRobot {

 \   private RobotDrive myRobot; // robot drive system
    private Gyro gyro;

 \   double Kp = 0.03;

    public GyroSample() {
        gyro = new AnalogGyro(1); \            // Gyro
on Analog Channel 1
        myRobot = new RobotDrive(1,2); \ // Drive
train jaguars on PWM 1 and 2
        myRobot.setExpiration(0.1);
 \   }
```

```
 public void autonomous() {
        gyro.reset();
        while (isAutonomous()) {
           double angle = gyro.getAngle
           myRobot.drive(-1.0, -angle*Kp);
           Timer.delay(0.004);
        }
        myRobot.drive(0.0, 0.0);
 \   }
}
```

Maryland ROBOTICS ALLIANCE
Strengthening FRC Mentorship
www.mdrobotalliance.org

Better But Not Great



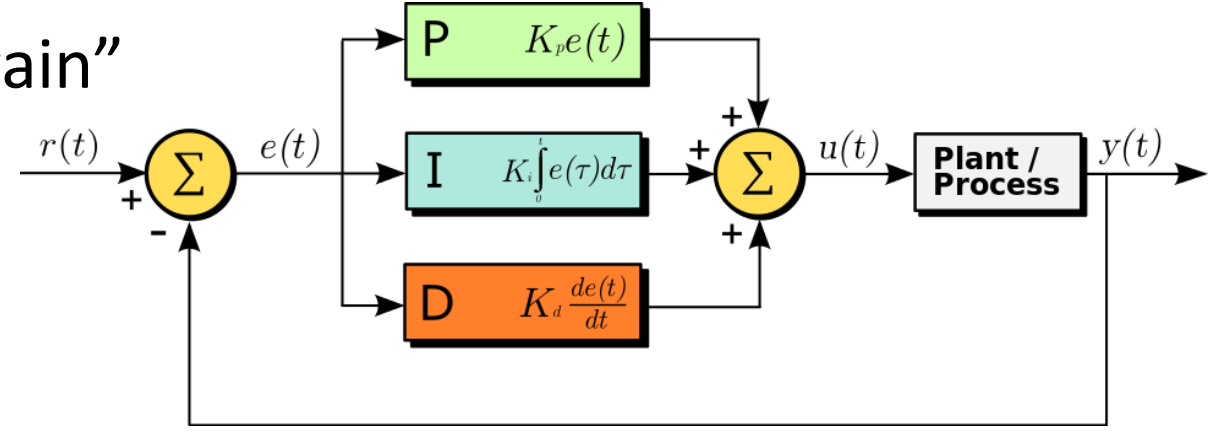
Feedback

- Sensors can provide a signal to allow you to correct the current output value
- This corrective value is called “error”
  - As in the difference between what you got vs what you wanted
- A first order system just takes that error and applies it to the output with a proportional constant – K_p
  - That is what we just did with the gyro

Maryland ROBOTICS ALLIANCE
Strengthening FRC Mentorship
www.mdrobotalliance.org

PID Loops

- Proportional
  - Constant “gain”
- Integral
  - Past error
- Derivative
  - Rate of change



PID Applications

- Command a motor to a specific position
  - Rotate a turret
  - Move an arm
  - Extend an elevator
- Command a motor to turn at a specific speed
- Feedback sources
  - Potentiometer
  - Encoder

Maryland ROBOTICS ALLIANCE
Strengthening FRC Mentorship
www.mdrobotalliance.org

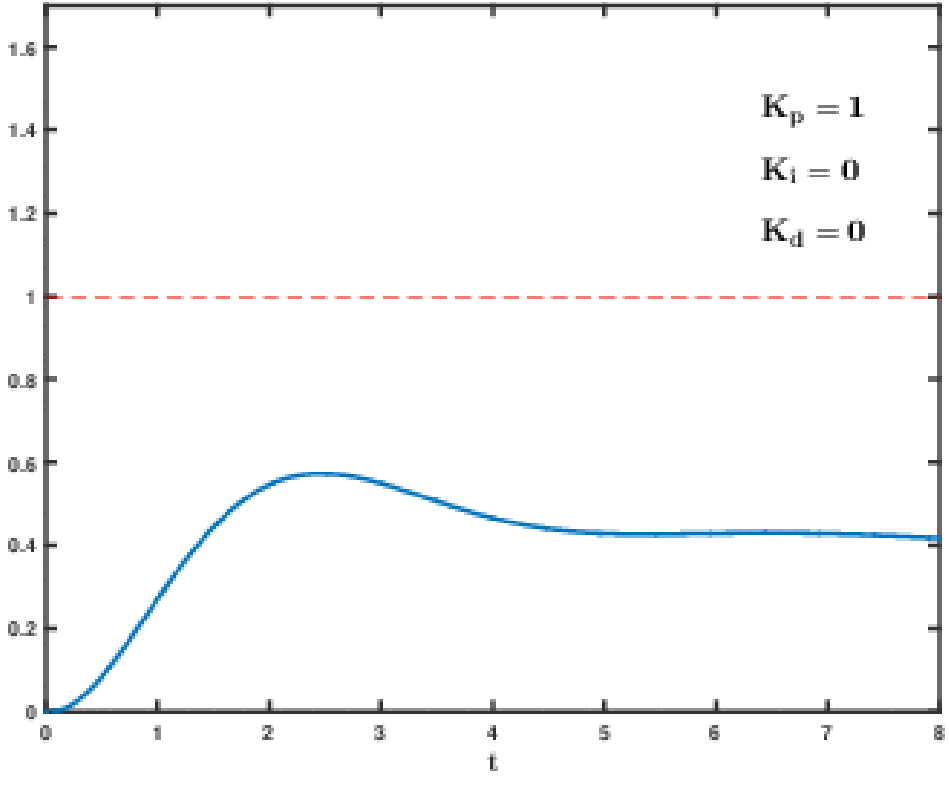
Example

```
Joystick turretStick(1);
Jaguar turretMotor(1);
AnalogChannel turretPot(1);
PIDController turretControl(0.1, 0.001, 0.0, &turretPot, &turretMotor);

turretControl.Enable();  // start calculating PIDOutput values

while(IsOperator())
{
   turretControl.SetSetpoint((turretStick.GetX() + 1.0) * 2.5);
   Wait(.02);      // wait for new joystick values
}
```

1

Maryland ROBOTICS ALLIANCE
Strengthening FRC Mentorship
www.mdrobotalliance.org

Tuning the PID



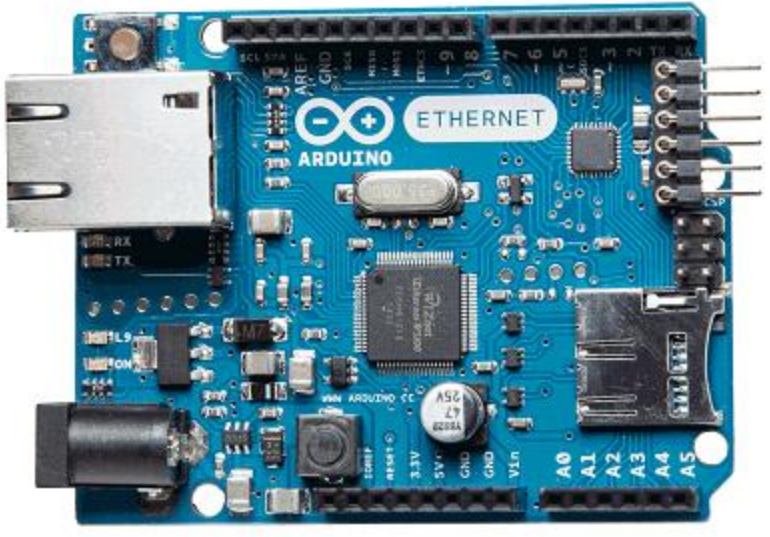
Noise! Limited inputs!

- Only 4 ports
- Analog signals are prone to noise over long wire runs
- Always a good idea to digitize as close to the source as possible
- More than a foot or two is asking for trouble

Maryland ROBOTICS ALLIANCE
Strengthening FRC Mentorship
www.mdrobotalliance.org

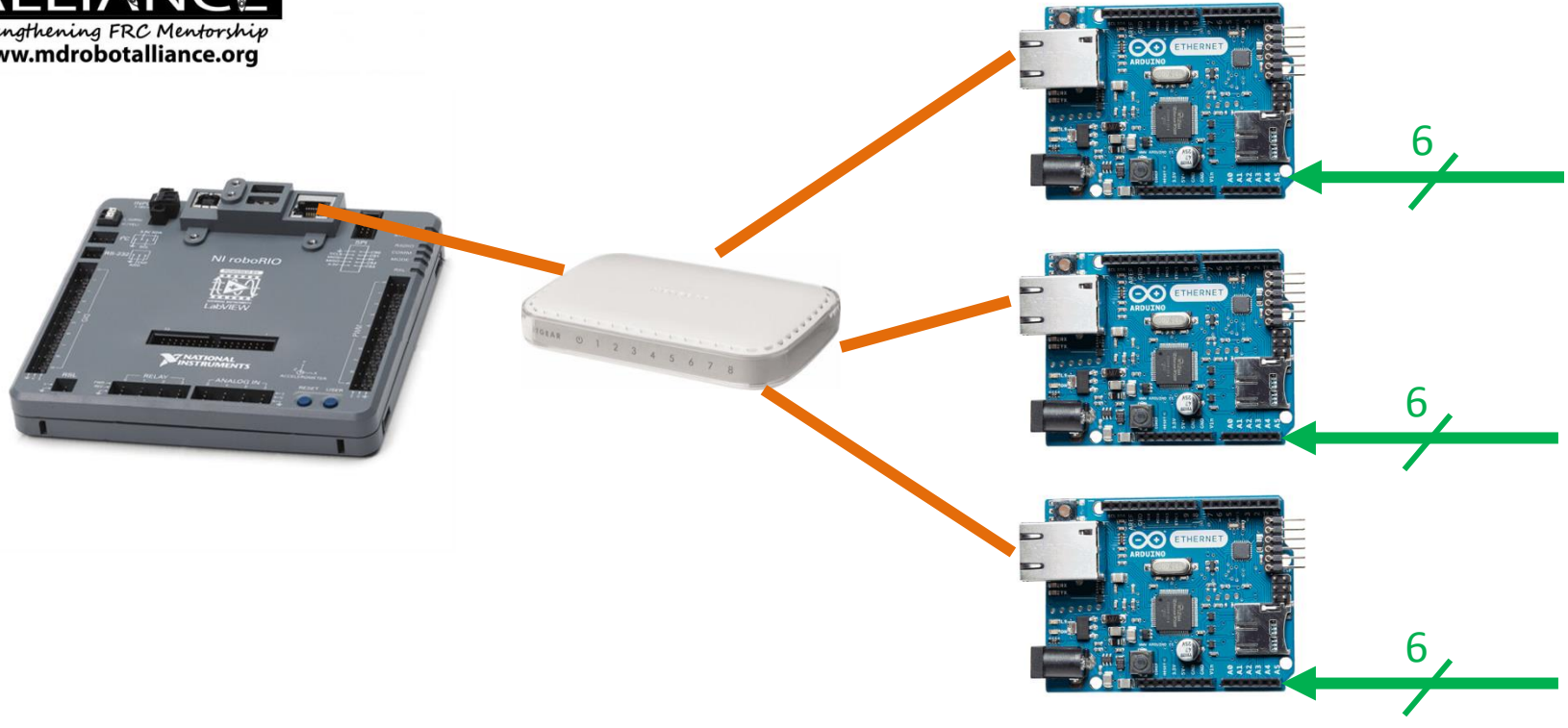
Try A Sensor Net

- Multiple Processors
  - RoboRIO has only 4 analog inputs
  - Arduino has 6!
- Ethernet/Enet Shield
- Many Analog Inputs
- Short runs to sensors
- Read a webpage on port 80



Maryland ROBOTICS ALLIANCE
Strengthening FRC Mentorship
www.mdrobotalliance.org

Arduino Sensor Net w/18 Analog Inputs



Maryland ROBOTICS ALLIANCE
Strengthening FRC Mentorship
www.mdrobotalliance.org

Other Approaches

- Some folks add a camera processor using another SBC (eg RasPi)
- This read over the network with the Pi doing all of the heavy computation
  - Frees the RoboRIO to drive the bot
- But it is still a distributed processing environment

Maryland ROBOTICS ALLIANCE
Strengthening FRC Mentorship
www.mdrobotalliance.org

Questions