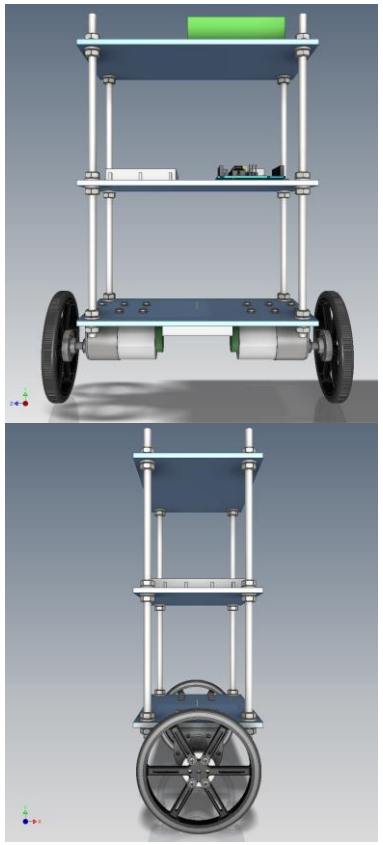
Self Balancing Robot

-by Abhishek Ghogare (153059006)

Problem Statement



- Robot balances itself by moving in 1 dimension
- Accelerometer/Gyro sensor detects the direction of fall
- Microcontroller receives sensor data and acts on motors connected to wheels
- Microcontroller should track the movement of the robot to maintain the its position

Task specifications

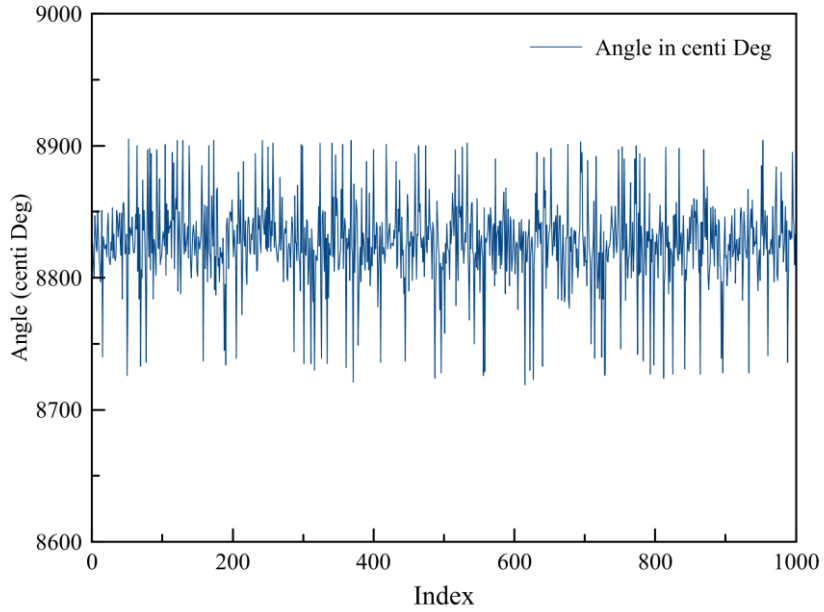
- I2C protocol communication for MPU
- Calculate bot angle with minimum noise
  - Calculate and smooth angle using only accelerometer
    - Use tan angle instead of sin to avoid max length calculation and error minimization
  - Integrate angular velocity over time to get angle
    - Two reasons to drift the angle
      - Non zero sensor output in steady state
      - Accumulation of error due to integration over time
  - Complementary filter, to fuse both angles
- Use PID to balance bot
- Quadrature encoder for position tracking
- Quadrature encoder for speed tracking

MPU-9250 Communication

- Two communication protocols to interact with MPU
  - SPI
    - Simple protocol
    - Pins : SCLK, MOSI, MISO, SS
  - I2C
    - Supports multi master-slave on same bus
    - Pins : SCL, SDA
- Used I2C module and library from Tiva Launchpad

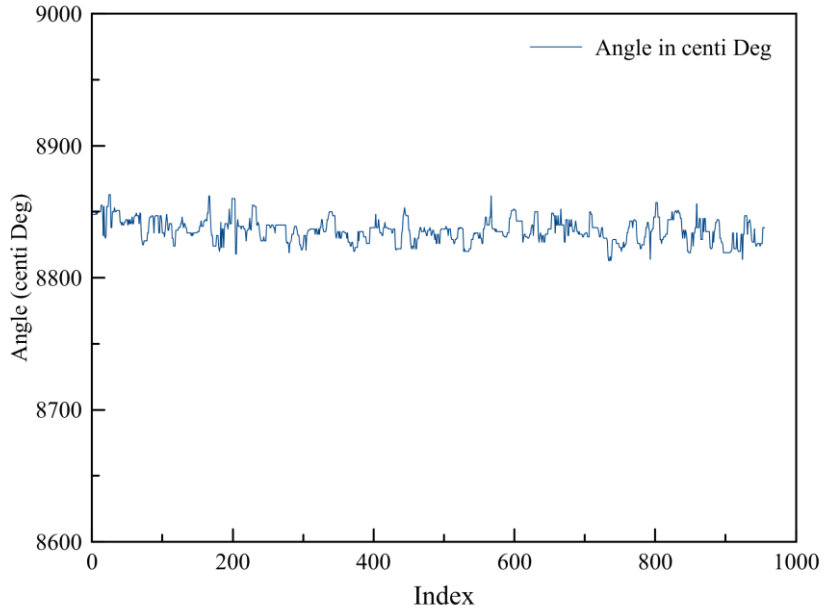
Bot angle calculation Accelerometer

- Very noisy even in steady state



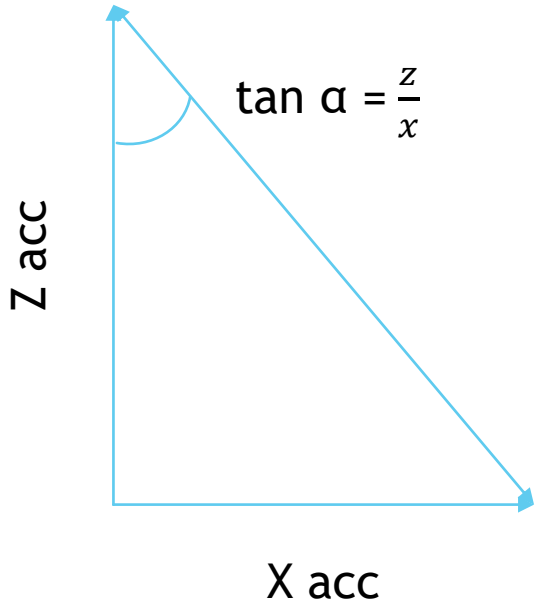
Bot angle calculation Accelerometer

- Using simple median filter decreased much noise
  - Uses 10 last samples



Bot angle calculation Accelerometer

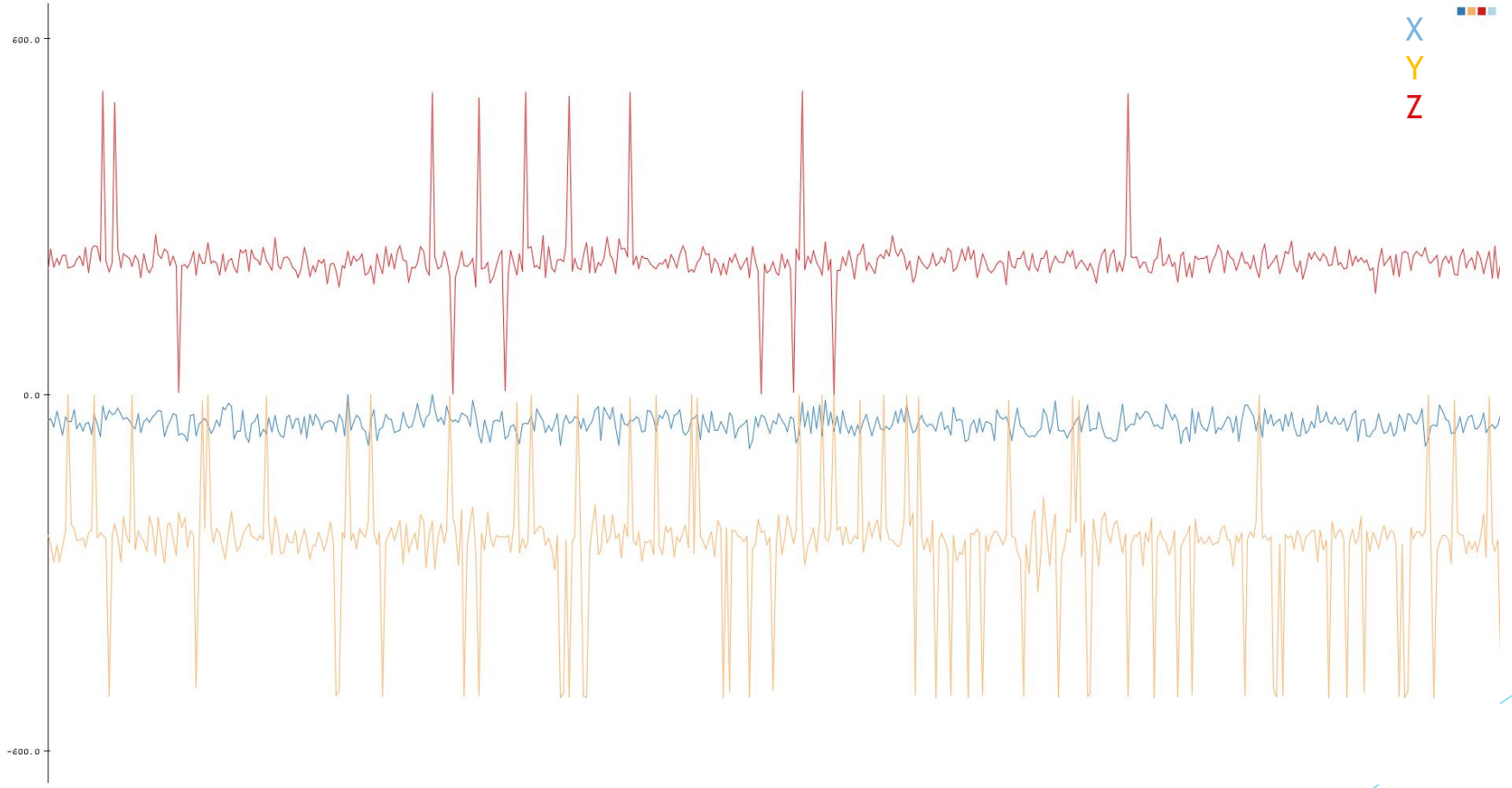
- Using tan(α) instead of cos(α)
- Two advantages
  - Excludes max value of accelerometer from consideration
  - Minimizes error by considering multiple axes
- Angle does not drift, so good reference point



Bot angle calculation Gyroscope

- Integration over time
- Very smooth, less noise
- Calculated angle drifts over time
  - Since error in sensor data also get integrated
  - Non-zero offset of the sensor output
  - Sensor sensitivity is not constant

Bot angle calculation
Gyroscope

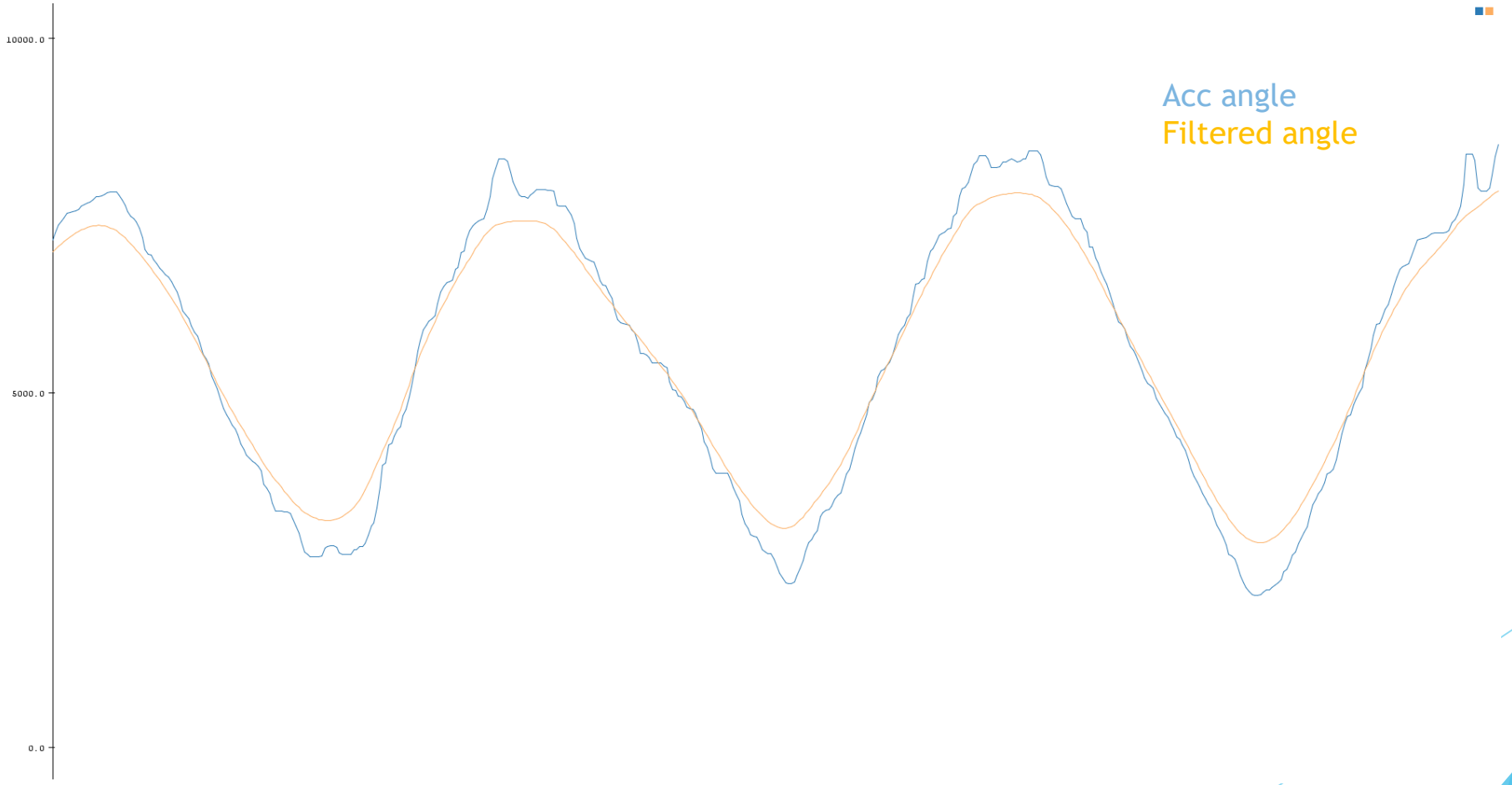


Bot angle calculation Accelerometer + Gyroscope

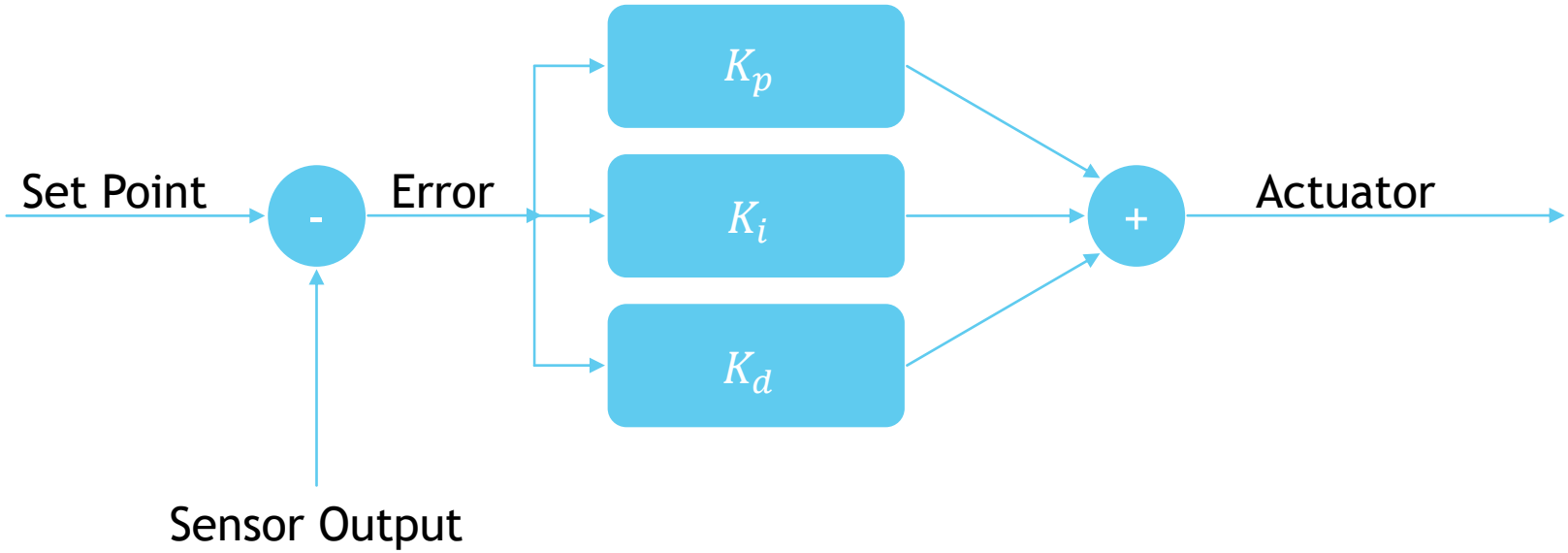
What is the solution?

- Complementary filter
  - $(a_a - a_g) \times \tau^2 \times dt^2 + 2(a_a - a_g) \times \tau \times dt \quad + v dt$

Bot angle calculation
Accelerometer + Gyroscope

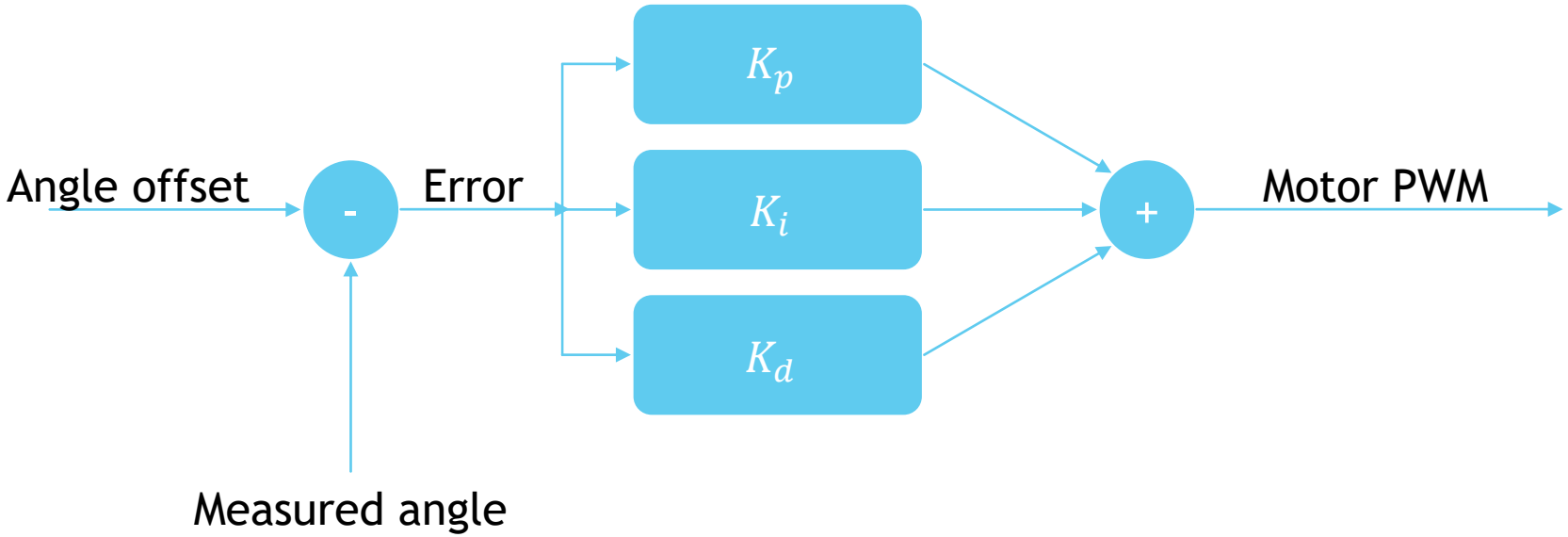


PID Controller



- K_p : Proportional term
- K_i : Integral term
- K_d : Derivative term

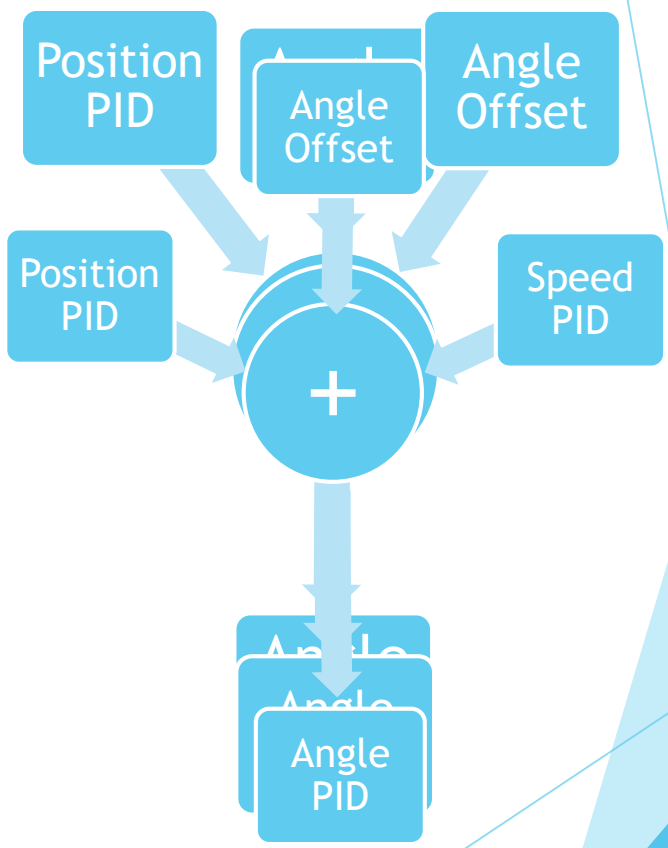
PID Controller



- K_p : Proportional term
- K_i : Integral term
- K_d : Derivative term

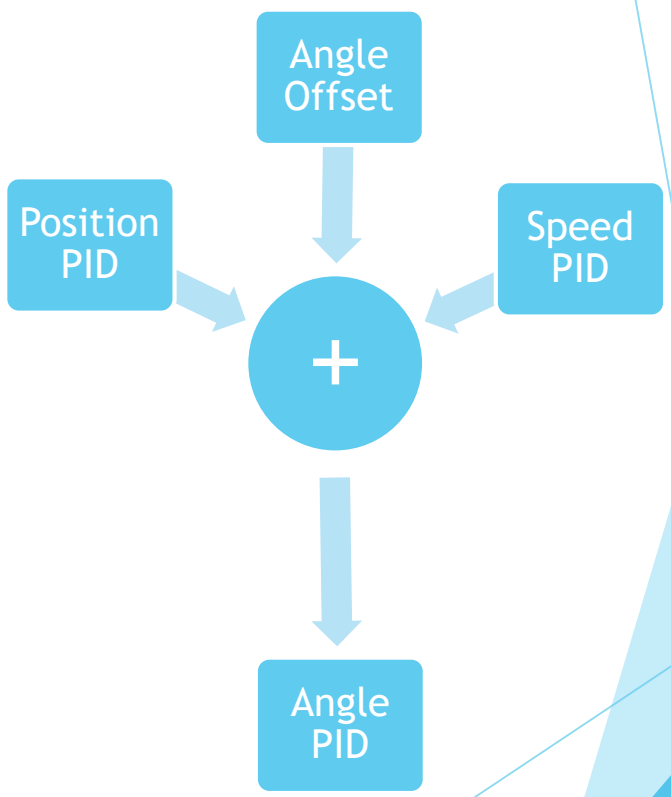
PID Controller

- Problem
  - Unstable bot even after PID tuning
- Solution
  - Added offset to PWM values for each motor by measuring
- Problem
  - Bot angle stabilizes, but not the position
- Solution
  - Cascade position PID, add output of position PID to set point of angle PID
- Another problem
  - Bot tries to stabilize position and angle both, but does not maintain the speed
- Solution
  - Add another PID for speed control, add output of speed PID to set point of angle PID



Cascaded PID

- Add position PID output to set point of angle PID
- Position tries to stabilize but speed not
- So, add speed PID output to set point of angle PID
- One advantage is no need to add angle offset since position PID compensate for that



Future Enhancements

- Directly use target speed PID to achieve desired motor speed
  - So, no need to add PWM offset to motors
- Use onboard DMP to offload MPU value processing
- Both encoder integration
- Remote control
- Height independent PIDs

Metrics

- Single iteration runs in 14ms
- Execution time should not be less than 10ms, it introduces velocity calculation errors

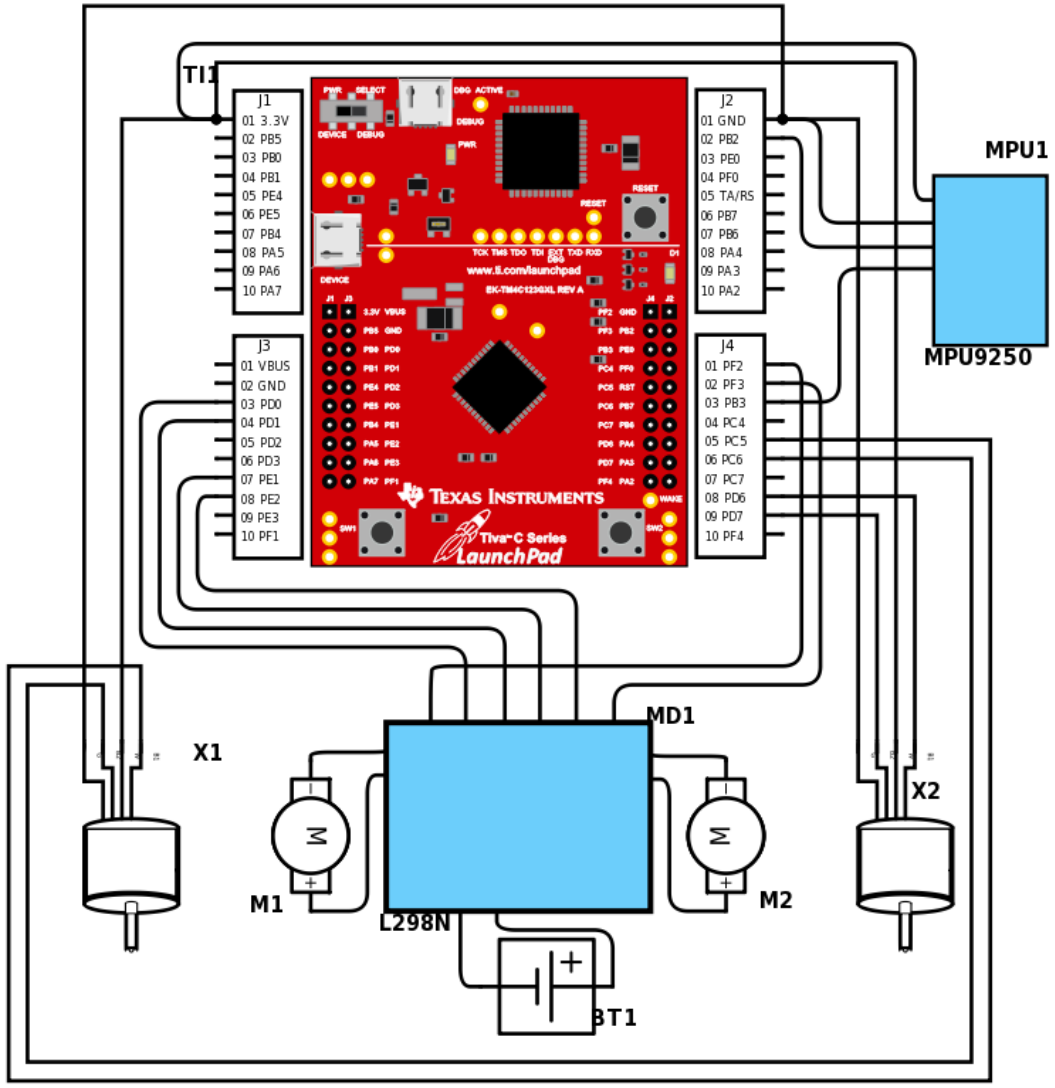
Hardware Modules

- Tiva Launchpad
  - I2C module for MPU communication
  - Quadrature encoder module for motor encoder
    - Two modules
    - Acceptable input frequency is $\frac{1}{4}^{th}$ of the processor frequency
    - Direction change detection

Software Modules

- PID controller library
- Median filter
- Complementary filter

Connections Diagram



Project Timeline

Checkpoint	Date
MPU testing with Tiva Launchpad	07 Oct 2016
Robot connection and basic motor movement	09 Oct 2016
Angle measurement using accelerometer	11 Oct 2016
Angle calculation using gyroscope	11 Oct 2016
Sensor fusion using complementary filter	12 Oct 2016
PID controller integration	14 Oct 2016
Quadrature encoder module to track position	16 Oct 2016
Position PID integration	17 Oct 2016
Motor speed calculation and Speed PID integration	19 Oct 2016
Another layer of speed PID to achieve desired speed	23 Oct 2016