LAB #6 TILT SENSING AND PID CONTROL

OBJECTIVE

- Learn how to estimate tilt angles using accelerometers;
- Learn how to perform PID control on a robot.

INTRODUCTION

I. Tilt Sensing

In addition to gyro integration, the robot orientation can also be determined by referencing some vectors of known directions. Three commonly accessible vectors are from the Earth's gravitational field (measured using accelerometers), magnetic fields (measured using magnetometers), and rotation (measured using rate gyros). Among the three vectors, the gravity vector is available everywhere and normally it cannot be interfered. The 3-axis accelerometer measurements (a_x, a_y, a_z) can be used to determine the roll angle ϕ and pitch angle θ for a robot that is stationary or moving at a constant velocity:

$$\phi = \arctan\left(\frac{a_y^B}{a_z^B}\right), \quad \theta = \arctan\left(\frac{-a_x^B}{a_y^B \sin\phi + a_z^B \cos\phi}\right)$$

Keep in mind that this method will not work if the robot is in acceleration!

Note: We typically use the ATAN2 function (with output angle range -180° to 180°) to solve for ϕ, and use the ATAN function (with output angle range -90° to 90°) to compute θ.

II. PID Control

A Proportional-Integral-Derivative (PID) controller, shown in Figure 1, is a negative feedback control method widely used in industrial systems. The proportional, integral, and derivative values can be interpreted in terms of time: P depends on the present error, I on the accumulation of past errors, and D is a prediction of future errors, based on current rate of change.

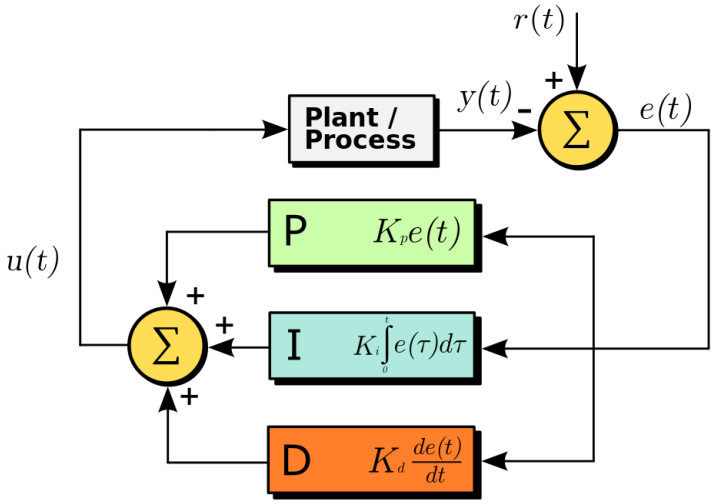

Fig.1. Feedback Control System with a PID Controller

The equation of a PID controller can be written as:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{d}{dt} e(t)$$

Where u is the control input (controller output); $e(t) = r(t) - y(t)$ is the error, which is the difference between the reference input r and the system output measurement y; K_p is the proportional gain; K_i is the integral gain; and K_d is the derivative gain.

Tuning of the PID controller could be a difficult process although there are only three tuning knobs. Many requirements, such as short transient time and high stability, are conflicting and hard to achieve at the same time. There are several methods for PID tuning, which includes manual and computerized methods. Please check the class handout for additional information about PID controller and its tuning.

III. Warnings

1. Put the safety belt on the laptop. Remember: click it or ticket!
2. Do not keep the Create powered on unattended. The robot may drive itself off the table. Only turn on the robot power when you are ready for a test and turn it off right away when you are done with the test.
3. Flip the laptop screen up when you are driving the robot. Do not use the laptop as a bumper!
4. If your laptop needs an update when restarting, let it finish. Do not turn off the laptop in the middle of an update.
5. Make sure to properly shutdown the laptop after each session. Do not put the laptop in sleep! It will just drain the battery after sitting idle for a week!
6. Kinect use LASER for mapping. This is a low power laser that is projected at many directions. Microsoft claim it's harmless to our eyes but we all know how much we can trust Microsoft. My recommendation is not to stare at the laser window (when it's red) for an extended period of time. At least we are not those kids that play video games all day long!

PROCEDURE

I. Tilt Sensing

1. Open up MATLAB code "SMART_RUN" and save as a different name;
2. Modify the code so that you can estimate the pitch and roll angles with the tilt sensing equations discussed earlier;
3. Display the estimated pitch and roll angles in "real-time"; Tilt the robot to different angles and check if the outputs make sense;
4. Drive the robot on the level floor and observe the changes in pitch and roll angles when the robot is moving around;
5. Show the instructor your results.

II. PID Control

1. Modify the "SMART_RUN" code so that the robot will follow a pre-determined heading angle (measured in SD.TotalAngle) as it moves forward at a constant speed;
2. Implement the proportional controller first;
3. Implement the Proportional-Derivative (PD) controller (hint: you do not have to take numerical derivative of the heading angle);
4. Implement the Proportional-Integral (PI) controller;
5. Implement the PID controller;
6. Tune the controllers properly to achieve a fast response, zero-overshoot, and low steady-state error.
7. Show the instructor your results.

DELIVERABLE

An abbreviated lab report of your experiments and answer the following questions:

1. In the tilt sensing equations provided earlier, explain why θ is a function of ϕ instead of another way around?
2. Were the estimated pitch and roll angles valid when the robot was moving?
3. Quantify in a table the performance of the four controllers in terms of settling time (time took to reach ±5% of the steady state value) and steady state error?
4. What made the tuning process difficult?