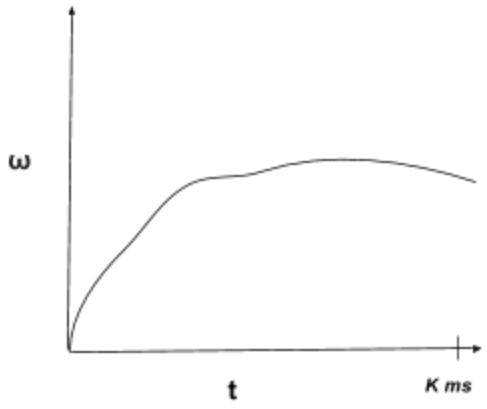
Camera Azimuth Angle Control

This writeup is intended to explore a method for controlling the azimuth angle of the camera system using a continuous rotation servo. This method uses a riemann sum to approximate the change in angle from the last servo update, and then sends the servo a new value to compensate for this angle change.

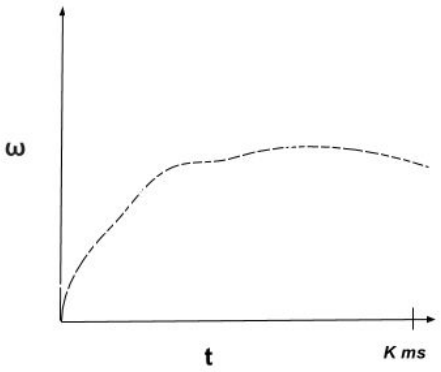
Suppose we sample a gyro sensor on the camera platform as often as possible, and we are updating the servo every ***k*** milliseconds. Over a ***k*** millisecond period, the rate of rotation (**ω**) cannot be assumed to be constant. For example, it may look like the following graph:



$$\Delta\Theta_{total} = \int_0^k w(t)dt$$

Equation 1

ω is in units *degrees / ms* and ***t*** is in units *ms*. Therefore, equation 1 describes the total change in angle over ***k*** ms**.** However, since we are sampling **ω**, we will get the following discrete graph:



We can use the following equation to approximate the area under the function using riemann sums - this computation runs as frequently as possible between servo updates:

$$\Delta\Theta_{approx} \mathrel{+}= \omega_{current} \bullet \Delta t$$

Where Δt is elapsed time since the last gyro read.

This method can be implemented as follows.

```
// NOTE: Using the MSTIMER2 Library,
// int_flag is set to true every k milliseconds

int k = 200; // update the servo every 200 milliseconds
float runningSum = 0; // riemann sum
float oldTime;
float currentTime;

void loop() {
  currentTime = millis();
  deltaT = currentTime - oldTime;
  runningSum += deltaT * readGyro(); // readGyro returns deg/s
  oldTime = currentTime;

  if (int_flag) {
    rotateServoBy(-runningSum);
    runningSum = 0;
    int_flag = false;
  }
}
```


- 5/07/17 : Max A Bowman
- maxabowman@gmail.com