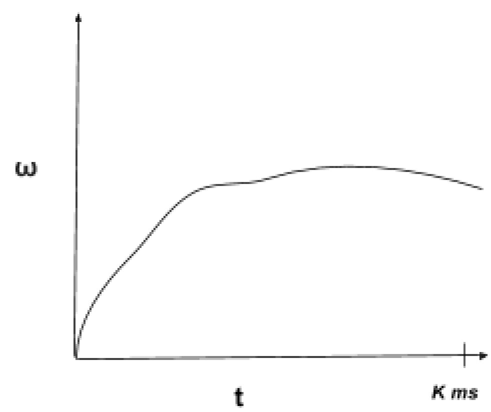
Camera Azimuth Angle Control

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

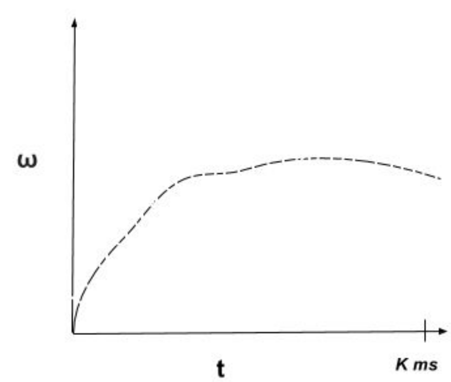
Suppose we sample a gyro sensor on the camera platform as often as possible, and we are updating the servo every ***k*** milliseconds. Over a ***k*** millisecond period, the rate of rotation (**ω**) cannot be assumed to be constant. For example, it may look like the following graph:



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

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 motor 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: update_flag is set to true every 50 ms)

```
  int currentTime = millis();
  int deltaT = currentTime - timer;
  positionChange += deltaT * readGyro() / 1000; // Gyro data is given in 1000ths of a degree, so we divide by 1000
  timer = currentTime;

  // Update motor position
  if(update_flag) {
    // Update stepper position (Z/AZM)
    float degreesActuallyCorrected = rotateStepperBy(-positionChange);
    positionChange += degreesActuallyCorrected;
    // Reset flag
    update_flag = false;
  }
}
```


- 5/28/17 : Max A Bowman
- maxabowman@gmail.com