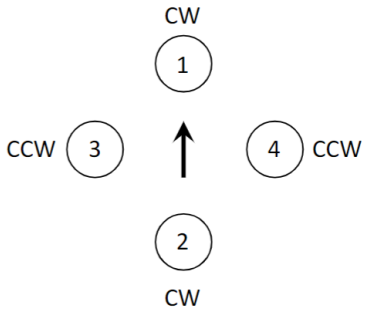
1) Describe in detail how a PID controller works **conceptually** in the context of stabilizing the quadcopter yaw **rate**. What do each of the gains do?

The y component(down) data from the gyroscope must be ran through a second order Butterworth filter. The PID yaw rate controller will have three components, proportional, integral, and derivative, and use the filtered yaw rate data and a desired yaw rate variable. The error is the difference between the desired yaw rate and actual filtered yaw rate. The error will be used in the controller. To yaw a plus configuration quadcopter motor commands must be sent to motors 1(north), motors 2(south), motors 3(west), and motor 4(east).



Motor Commands:

$$Motor\ 1 = throttle - (P\ controller + I\ controller + D\ controller)_{Yaw}$$
$$Motor\ 2 = throttle - (P\ controller + I\ controller + D\ controller)_{Yaw}$$
$$Motor\ 3 = throttle + (P\ controller + I\ controller + D\ controller)_{Yaw}$$
$$Motor\ 4 = throttle + (P\ controller + I\ controller + D\ controller)_{Yaw}$$

Each controller component has a scalar component, K_P, K_I, and K_D (gains). The proportional controller is the product of K_P and the error. The integral controller is the product of K_I and the integral of the error. The derivative controller is the product of K_D and the derivative of the error. The purpose of the proportional controller is to increase/decrease the rise time. Increasing K_P will increase % overshoot and decreasing K_P will decrease % overshoot. The purpose of the integral controller is to reduce steady state error. Increasing K_I will increase % overshoot and decreasing K_I will decrease % overshoot. The purpose of the derivative controller is to reduce % overshoot. Increasing K_D will decrease rise time and decreasing K_D will increase rise time.

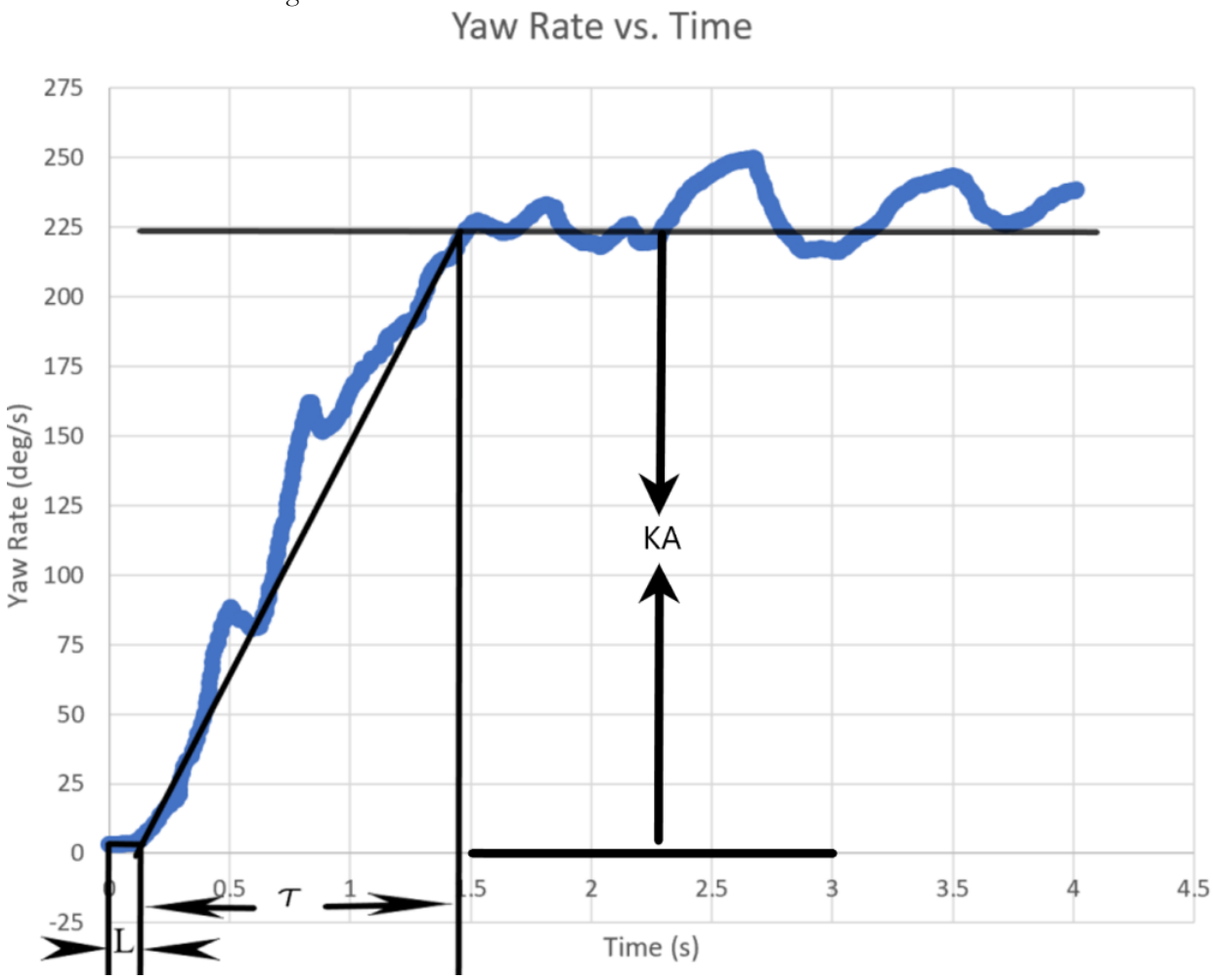
2) Describe in detail how to experimentally tune the yaw rate stabilization controller using open loop gain tuning (tangent method).

Once the controllers are built a gain tuning method (tangent method) must be implemented to establish starting K_P, K_I, and K_D values. To perform the Ziegler-Nichols tangent method throttle is set to 50% for all motors, then 20% is subtracted to motors 1 and 2, and 20% is added to motors 3 and 4. From a graph of the response variables L, τ, and K can be determined. L is the dead time, τ is the time constant, and K is the gain. With these variables and a few sets of equations starting K_P, K_I, and K_D values can be determined. From here set the desired rate to 60 $\frac{Deg}{s}$ and increased and/or decreased K_P, K_I, and K_D according to the K_P, K_I, and K_D relations stated above until the following target parameters are met.

Parameters:

- Percent Overshoot < 20%
- Settling Time < 3 second
- Rise Time < 2 seconds

3) Show the estimated PID gains from the open loop gain tuning. Make sure to show a plot of the dat used to calculate the gains.



The graph above is from a step input of 20 = A. Variables L, τ, and K can be determined from the data.

225 deg/s = KA

K = 11.25

$$L = .109983\ s$$

$$M = \frac{226.9226855 - 3.75816}{1.528268814 - .109983} = 157.3480594$$

$$y = 157.3480594x - 13.54744662$$

$$x = \frac{225 + 13.54744662}{157.3480594} = 1.516049499$$

$$\tau = 1.516049499 - .109983 = \tau = 1.406066499\ s$$

The Ziegler-Nichols open loop method table can be determined from variables L, τ, and K:

	Kp	Ti	Td	tau	L	K
P	$\frac{\tau}{KL}$	-	-	1.406066	0.109983	11.25
PI	$(.9)\frac{\tau}{KL}$	(3)L	-			
PID	$(1.2)\frac{\tau}{KL}$	(2)L	(.5)L			

	Kp	Ti	Td	τ	L	K
P	1.136390977	-	-	1.406066	0.109983	11.25
PI	1.022751879	0.329939	-			
PID	1.363669172	0.219966	0.054992			

$$K_D = K_P \times T_D \quad , \quad K_I = \frac{K_P}{T_I}$$

From the table and equation above starting K_P, K_I, K_D are determined for P, PI, PID controllers:

P controller:

K_P
1.1364

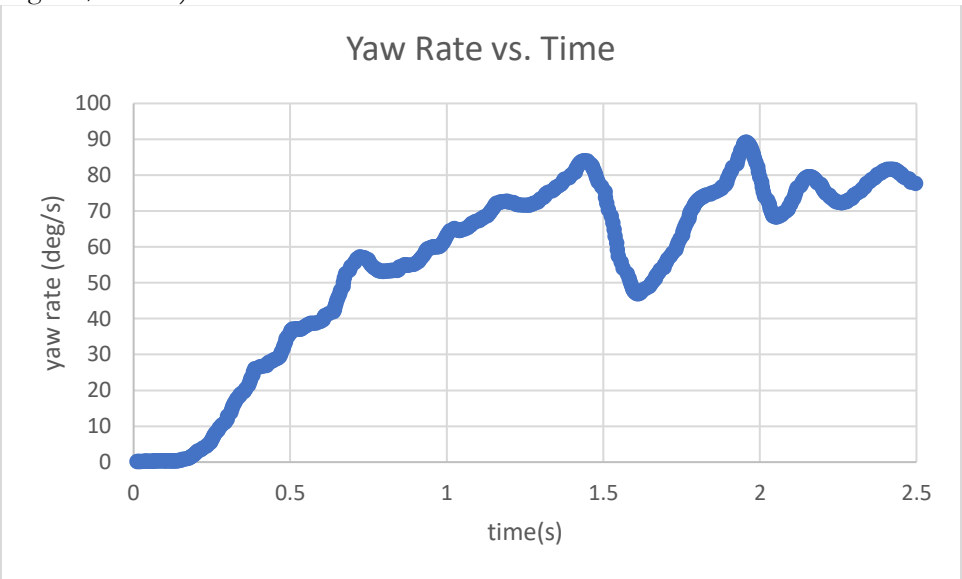
PI controller:

K_P	K_I
1.0228	3.0998

PID controller:

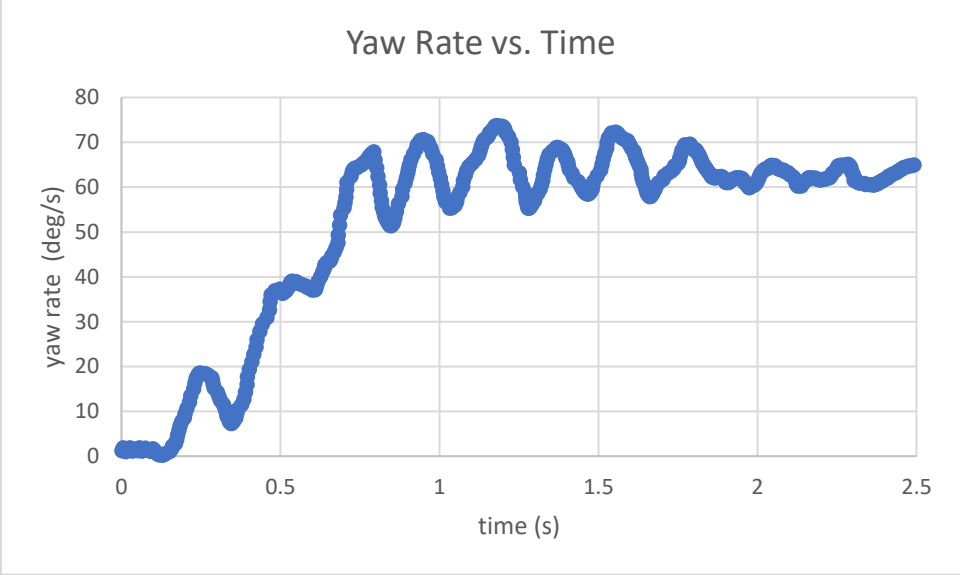
K_P	K_I	K_D
1.3637	1.5000	0.0750

4) Conduct a step response where the quadcopter goes from a desired yaw rate of zero to 60 degrees/second. Show the results (clearly indicate where the step input is started) and estimate the percent overshoot, settling time, and rise time (time to go from 0 to 60 degrees/second).



The Yaw Rate vs. Time graph above is from a PD controller with K_P and K_D values of 2.7 and 1.8 respectively. The highest peak is the second peak which is about 90 deg/s which is about 50% overshoot. Rise time is about 1 seconds. The settling time is approximately 2 seconds. The yaw rate has a large steady state error of almost 15 deg/s.

5) In order to safely fly a quadcopter, your attitude stabilization controller needs to be fast and accurate. Modify your gains until you have met the required performance for a 60 degree/second yaw rate step response. **Percent overshoot < 20%, settling time < 3.0s**, and **rise time < 2.0s**. Show the results, with the step input time clearly indicated. Make sure to provide your estimated percent overshoot, settling time, and rise time. Remember, you want your quadcopter to be as stable as possible (you will be performing free-flight testing very soon and the more stable your controller is the easier it will be to fly your quadcopter).



The Yaw Rate vs. Time graph above is from a PD controller with K_P and K_D values of 3.4 and 2.72 respectively. Although this does not meet all the required parameters, it nearly meets them and is the closest I've got on any of my pitch/roll/yaw controllers. The highest peak is the third peak which is about 75 deg/s which is about 25% overshoot. Rise time is about .7 seconds. The settling time is approximately 2 seconds. It's hard to tell if there is any steady state error. It appears to have a steady state error of 2 deg/s.