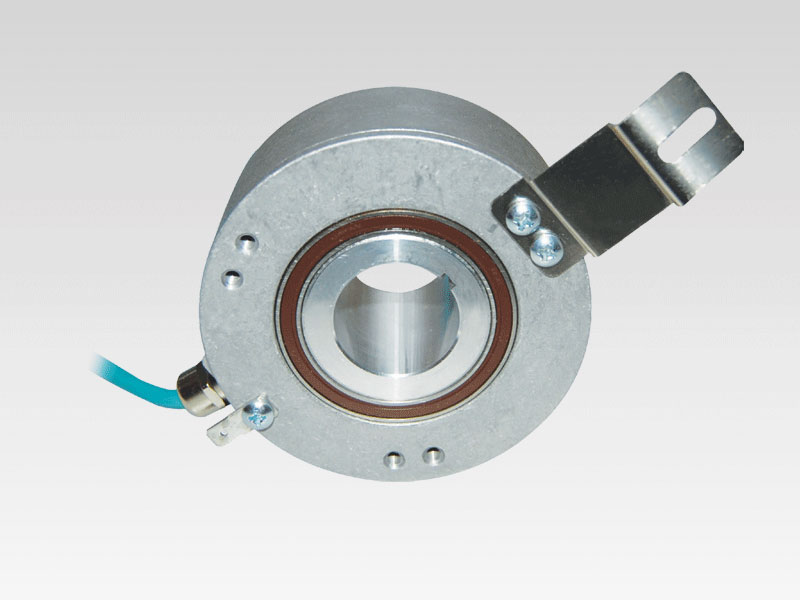
# Control Methodology

In this project we wanted to implement a closed loop control algorithm to keep motor speed constant when variable loads are attached to the DC motor. We will sample motor speed by the means of an encoder. Encoders are electromechanical devices that generates a PWM signal based on rotation speed of the shaft they attached to. In general encoders come with their own shaft and shafted encoder is inserted to the rotationary part. In this project DC motor under study has a large enough shaft so it is better to choose a hollow shaft type encoder because of the listed reasons.

  
**Figure:** Hollow Type Encoder Example.

(Especially) During startup, high current spikes will be observed so for the control loop, we wanted to control both current and speed of the motor. In this way, a simple speed and current control loop is constructed in Figure x and Figure x+1 . In the simulation, machine parameters are inserted directly but rectification plus Buck converter cannot be inserted directly. MATLAB’s PID tuner application is used to determine PID parameters in order to give a rough behavior and before tuning, system is linearized by the application itself. When the controller is inserted to rectifier-Buck converter cascaded system, PID app cannot linearize the system(due to MOSFET- diode model behaviors) and hence the parameters cannot be determined. Instead a constant voltage supply is assumed (220V in this case) and control simulation is shaped around this assumption.

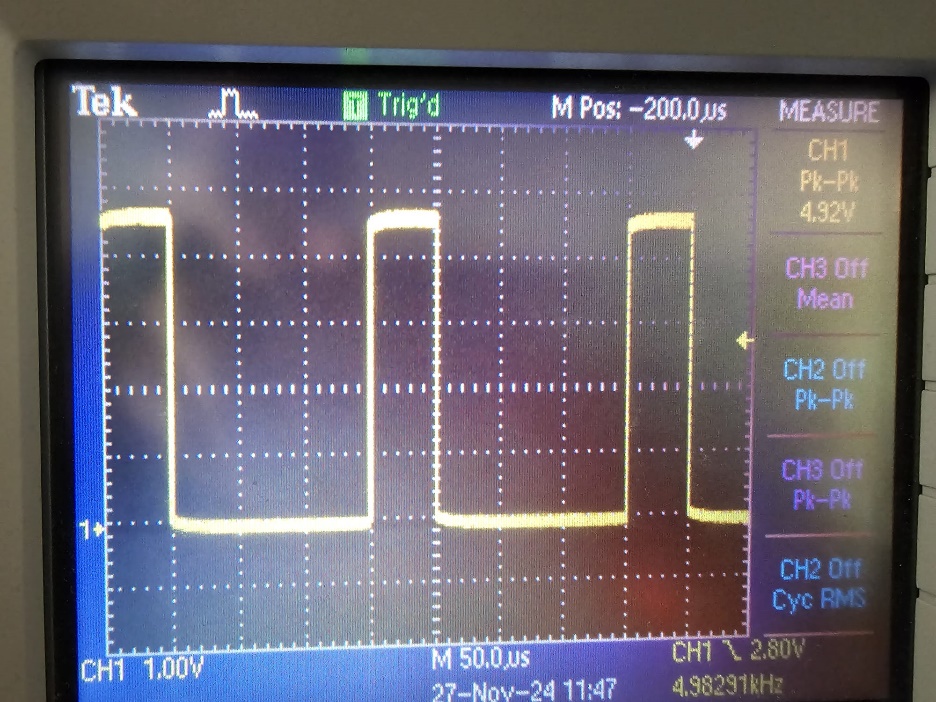
**Figure X:** Closed Loop Control Simulation

In the simulation, two “set point” cases are studied: a ramp input from startup to rated speed and a step input to observe the plant response. Outputs of the both behaviors are present in Figure x+1. In the figure, a robust enough speed control can be observed easily: during ramp input maximum 100rpm error is present (which is expected from a zero order system when a first order ramp input is given). Lesser errors can be achieved by increasing P parameters but that will make the motor behave more aggressively (most probably drawing currents with high amplitude spikes). In the same figure, DC motor currents can be observed with high spikes. This behavior is contradictory with expected behavior (a good enough current control is also desired) but at this point weight is given on speed control parameters and understanding system behavior. Depending on encoder choice and depending on controller choice (analog or digital) simulation parameters need an update and final current smoothening behavior is left to later steps.

**Figure X+1:** Closed Loop Control Simulation Results.

On the choice of controllers, we have two alternatives. For the analog controllers we have TL494 which is an analog PWM generator with two integrated error amplifiers to control duty cycle. [https://www.alldatasheet.com/view.jsp?Searchword=Tl494&gad\_source=1&gclid=Cj0KCQiAo5u6BhDJARIsAAVoDWtLRLNEs\_e6o1kucrSWyM9ZrsqDa7a9zjaRP\_92KMw0wz2SAJ0VUSAaAmuJEALw\_wcB] We could also use simple 555 timers to create the desired frequency signal but it requires additional comparators, op-amps to process feedback signals, hence 555 timer choice is discarded. If a digital controller will be employed, an Arduino Uno or STM32F103(Blue Pill) will be chosen. Arduino Uno is present on our personal inventories and it is a simple card to use but it has problems when generating high frequency signals (above 30kHz).

In order to test capabilities of Arduino Uno, an experimental circuit is constructed as seen on Figure x+2. Recorded output PWM signal, generated by Arduino Uno is present in Figure X+3. Arduino code is written to generate a 40kHz signal with 0.005 steps to duty cycle however observing the oscilloscope screen, generated signal is around 5kHz even though the code is written otherwise. 4.92V signal is expected to drive power MOSFET’s and IGBT’s when gate drivers are employed in between. Further tests will be conducted on the card and it is expected to fit the job however, depending on the situation it may be replaced by an STM card.

  
**Figure X+2:** Arduino Generated Experimental PWM Signal Recorded on Oscilloscope.