Back Electromotive Force-Based Speed Sensing for Brushless DC Motors in Unmanned Aerial Vehicles and Hardware Simulation Applications

Jennie Chen • Supervisor: Prof. Meyer Nahon

McGill University, Aerospace Mechatronics Lab • jia.y.chen@mail.mcgill.ca

CHALLENGE

Brushless DC (BLDC) motors are the most common type of motor found in unmanned aerial vehicles (UAVs), which have many uses in research, commercial, and consumer applications. One challenge in controller design for UAVs is the characterization of the motors and propellers. In this project, **a sensor is designed to measure the actual speed of a BLDC motor in revolutions per minute (RPM)** for closed loop control.

Design Constraints

- Implement RPM sensing system for hardware simulation and on-board UAVs
- Non-intrusive to normal flight operations
- Minimal delay (close to real-time data)
- Desired output: analog voltage (0-5v) or digital counter

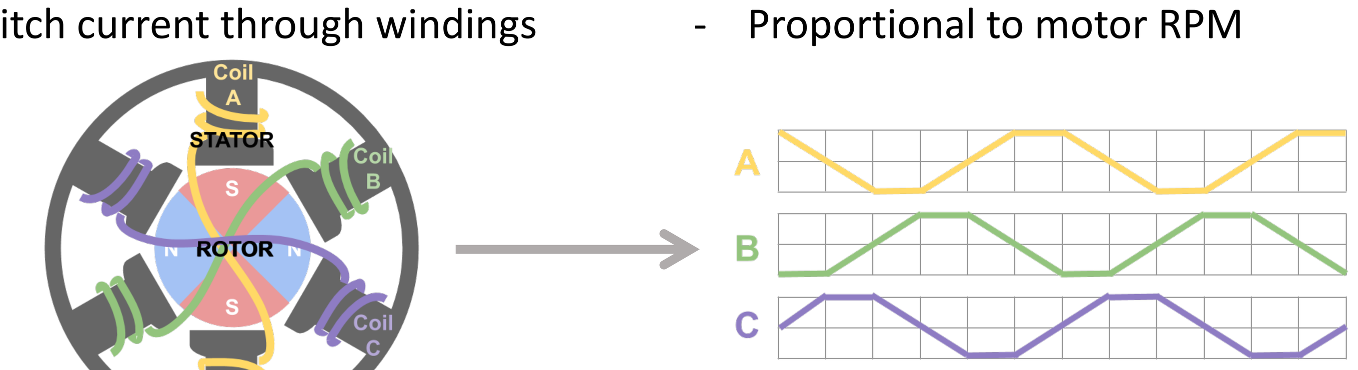
BACKGROUND

BLDC Motor

- Rotor (permanent magnet) + stator (windings)
- Commutation sequence control: Switch current through windings

Back Electromotive Force (BEMF)

- Self-induced voltage that appears in the opposite direction to the current flow due to magnetic induction
- Proportional to motor RPM



BEMF generated by each coil of the BLDC

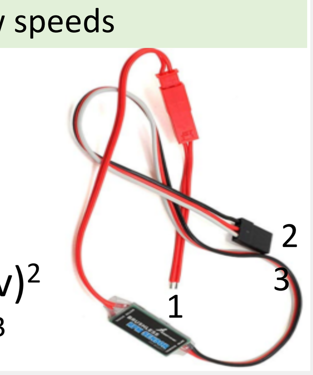
COMMERCIAL SOLUTIONS

Table 1. Comparison of Common RPM Sensing Approaches

	Pros	Cons
Hall Sensor	Wide range of speed	▪ External magnetic field interference ▪ Intrusive
Shaft Encoder	Absolute position information	▪ Non-feasible for flight ▪ Prone to mechanical wear
Back EMF Sensing	Sensorless, non-intrusive	▪ Accuracy issues at low speeds

Hobbywing RPM Sensor for High Voltage Electronic Speed Controllers

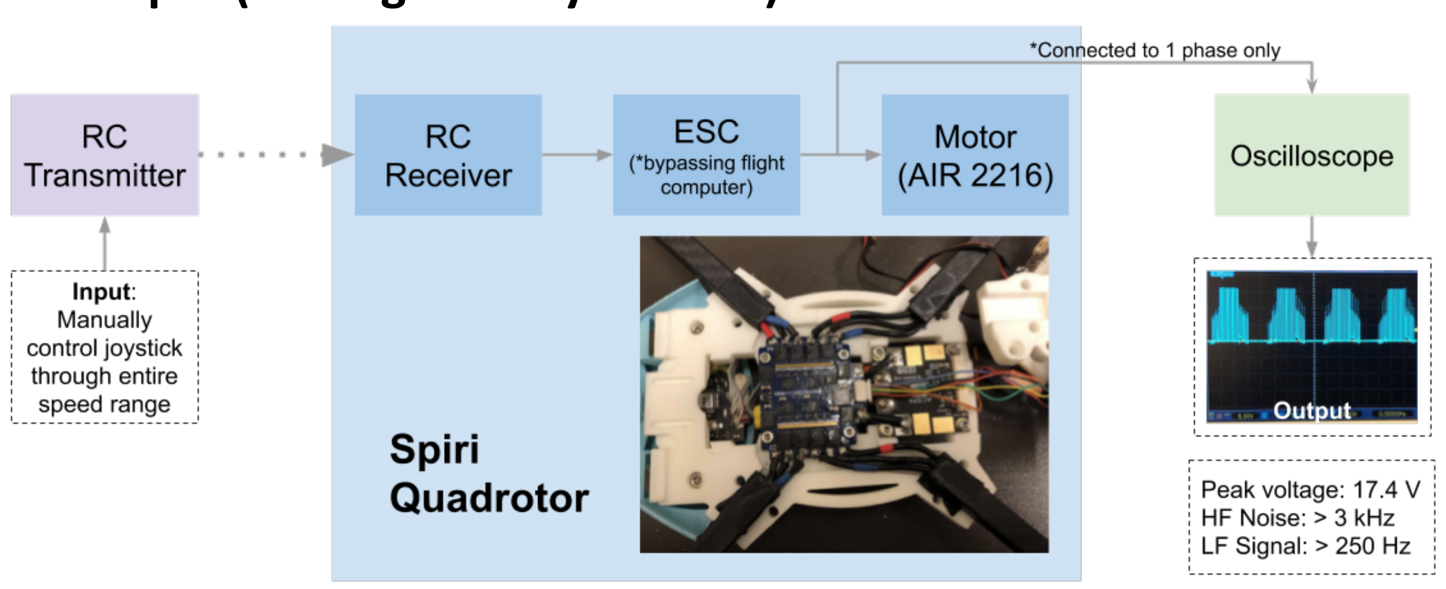
- Weight: 6g
- RPM Range: 1000 – 300000
- Inputs: 2 phases of brushless motor[1], sensor supply power (3.3v or 5v)[2]
- Output: square wave signal waveform with same frequency as BEMF[3]



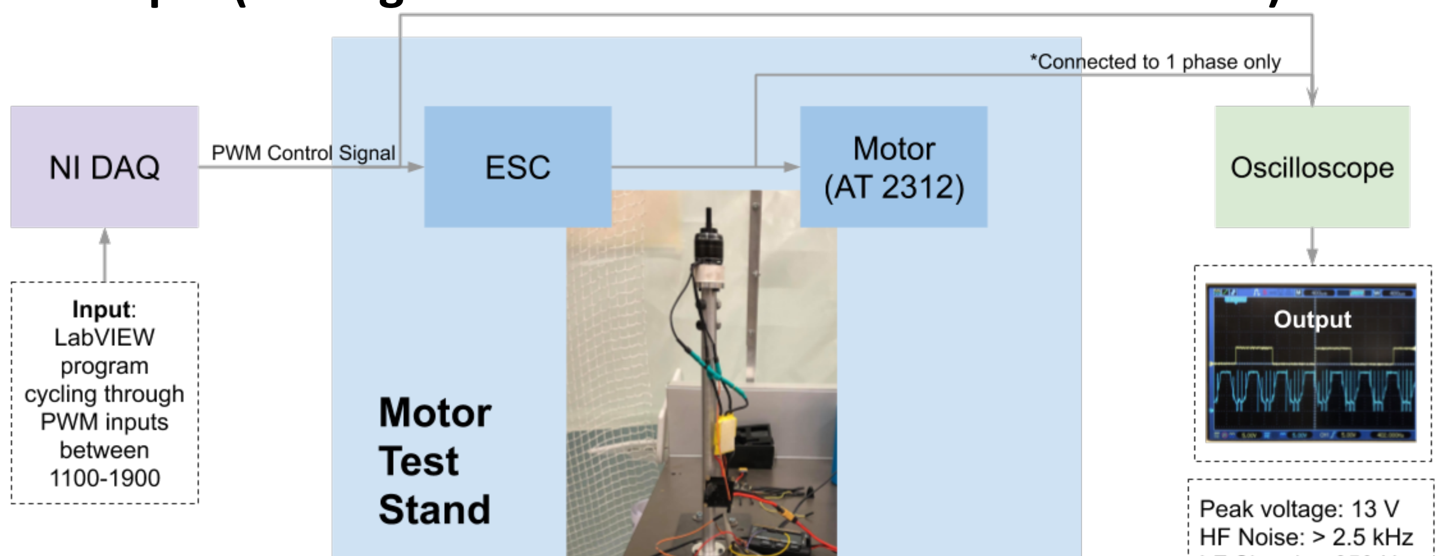
PROCEDURE

1- Characterization of the BEMF signal

Setup A (Testing directly on UAV)



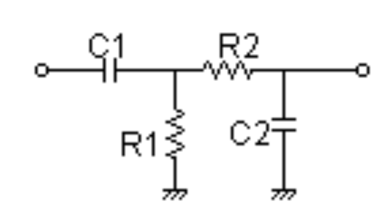
Setup B (Testing with motor characterization hardware)

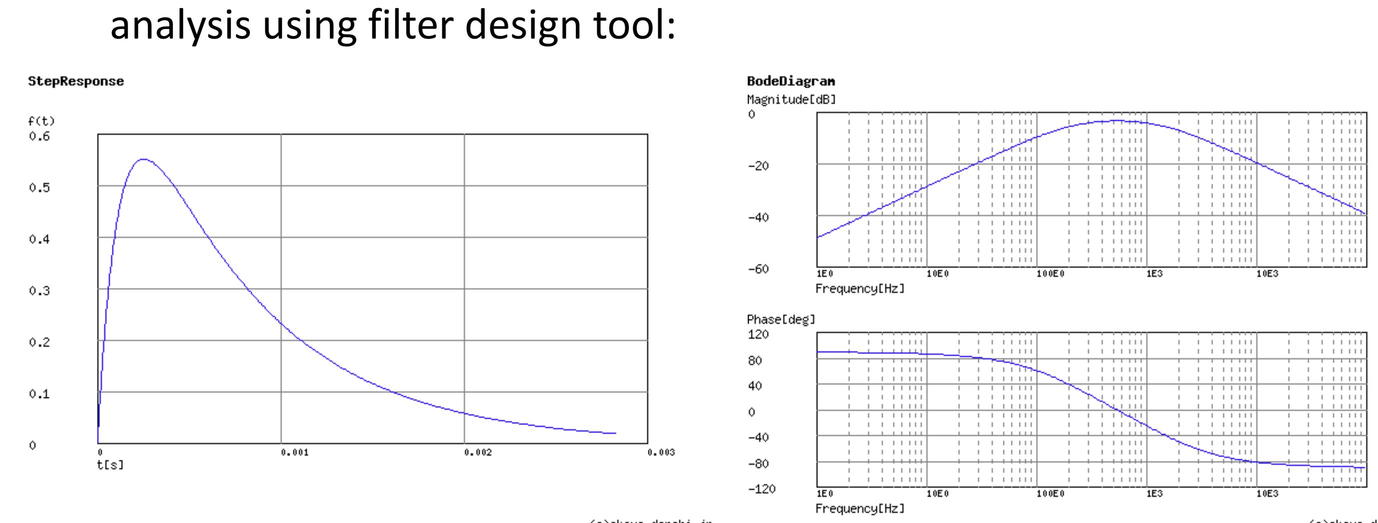


2- BEMF input filter design

Chosen filter: 2nd order passive RC band-pass filter

- **High-pass filter component:**
  - Eliminate DC offset
  - Let useful signal pass through
- **Low-pass filter component:**
  - Attenuate high-frequency noise
- Resistance and capacitance values chosen through frequency and transient analysis using filter design tool:





- C1 = 0.047 uF
- R1 = 12 KΩ
- C2 = 0.01 uF
- R2 = 15 KΩ

- fc = 547 Hz
- Q = 0.349
- ζ = 1.434

3- Output translation

A. Digital output:

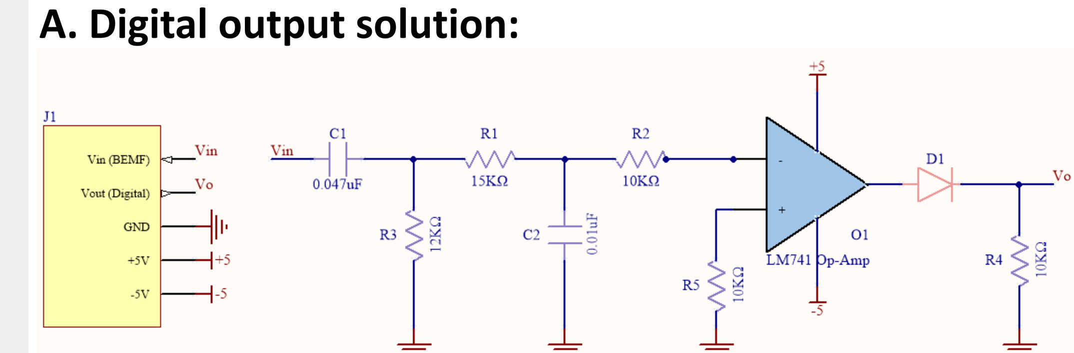
- Detect zero crossings using op-amp as comparator to output square wave
- Rectify signal using diode

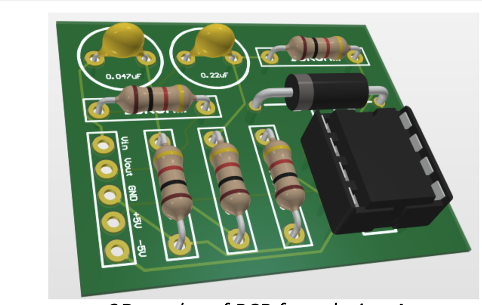
B. Analog output:

- Use of a frequency to voltage converter IC to output a voltage proportional to the input frequency (calculated from choice of passive components used)

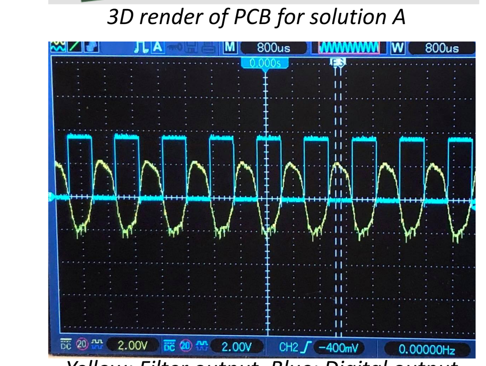
RESULTS

A. Digital output solution:





3D render of PCB for solution A

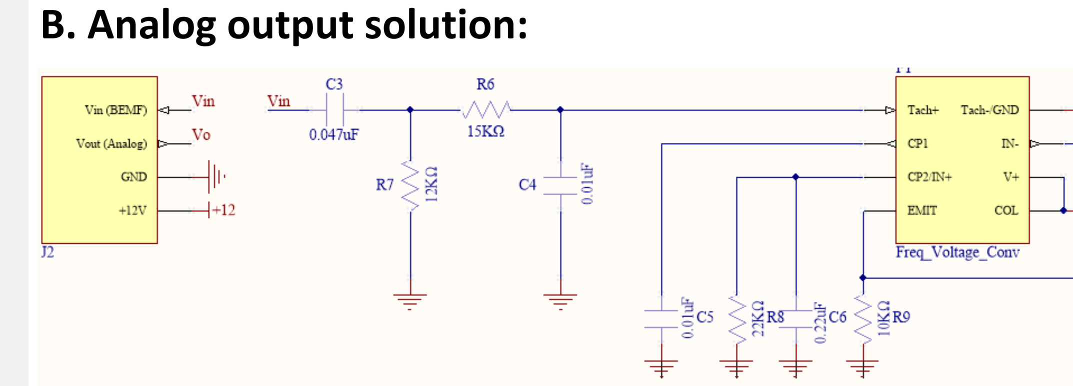


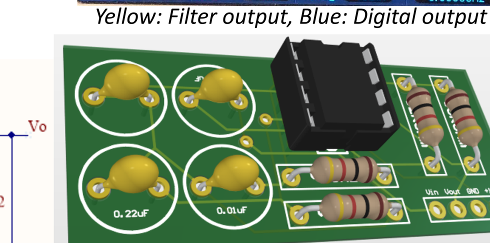
Yellow: Filter output, Blue: Digital output

RPM calculation from digital output (theoretical):

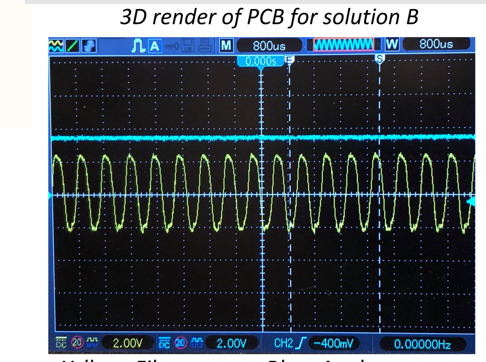
$$RPM = \frac{f_{BEMF}}{\left(\frac{1\,min}{60\,s}\right) * \#_{stator\_pole\_pairs}} = f_{BEFM} * 5$$

B. Analog output solution:





3D render of PCB for solution B



Yellow: Filter output, Blue: Analog output

RPM calculation from analog output (theoretical):

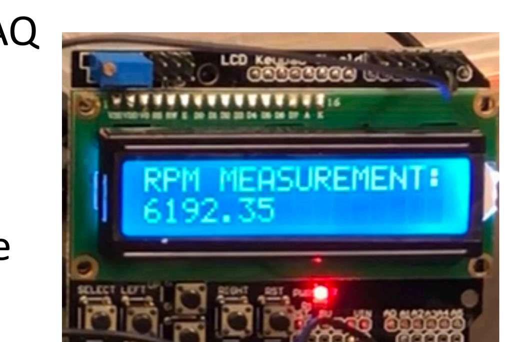
$$f_{BEMF} = \frac{V_o}{(V_{cc} * R_1 * C_1)} = \frac{V_o}{(12\,V * 22\,k\Omega * 0.01\,uF)} = V_o * 378.79$$

$$RPM = \frac{f_{BEMF}}{\left(\frac{1\,min}{60\,s}\right) * \#_{stator\_pole\_pairs}} = V_o * 1894$$

Verification of output range:

- Using Arduino Uno to simulate a flight computer or a DAQ reading the outputs
- Input driven by LabVIEW and NI DAQ

- **Solution A**: Digital counter using interrupts to determine f_{BEMF} and calculate RPM
  - Output range: **~2083 to 6978 RPM**
- **Solution B**: Using A/D converter and map() function to calculate RPM
  - Output range: **~940 to 6630 RPM**



Arduino Uno board displaying the RPM measurement obtained from solution A

FUTURE WORK

Validation of RPM measurement accuracy

Procedure:

- Use Arduino and LCD to measure, calculate and display the two RPM sensing solutions' theoretical output
- Use stroboscope to obtain actual RPM measurement
- Compare results
- Adjust RPM calculation if needed

Future Improvements and Uses

- Create circuitry for supplying +/-5 V using a battery source (for solution A)
- Print and assemble PCBs
- Integrate RPM sensor in motor characterization hardware and LabVIEW program to measure the speed of different motor + propeller combinations
- Integrate RPM sensor in UAVs for in-flight data logging and control feedback

McGill
AML
NSERC CRSNG
McGill Centre for Intelligent Machines