Introduction

The power electronics developed for the MBARI wave-energy conversion system, and recently adapted for the UofW/APL Tidal Turbine, perform the role of regulating the motor torque of these devices and converting the resulting voltage to a DC form that is suitable for charging batteries. This is a four-quadrant device that can operate the permanent-magnet brushless motors in these devices as either a generator or a motor, depending on if energy is flowing into or out of the attached batteries, respectively. This document outlines the connections, operation, software control, and specifications for this hardware as it is configured for the UofW/APL tidal turbine. Figure 1 shows the power-electronics and integrated load-dump (right-hand side) connected to the lab-test dyno setup, which can be replaced with the tidal turbine hardware for field use. The power-converter itself is self-contained, and can be connected to an external battery pack and control computer.

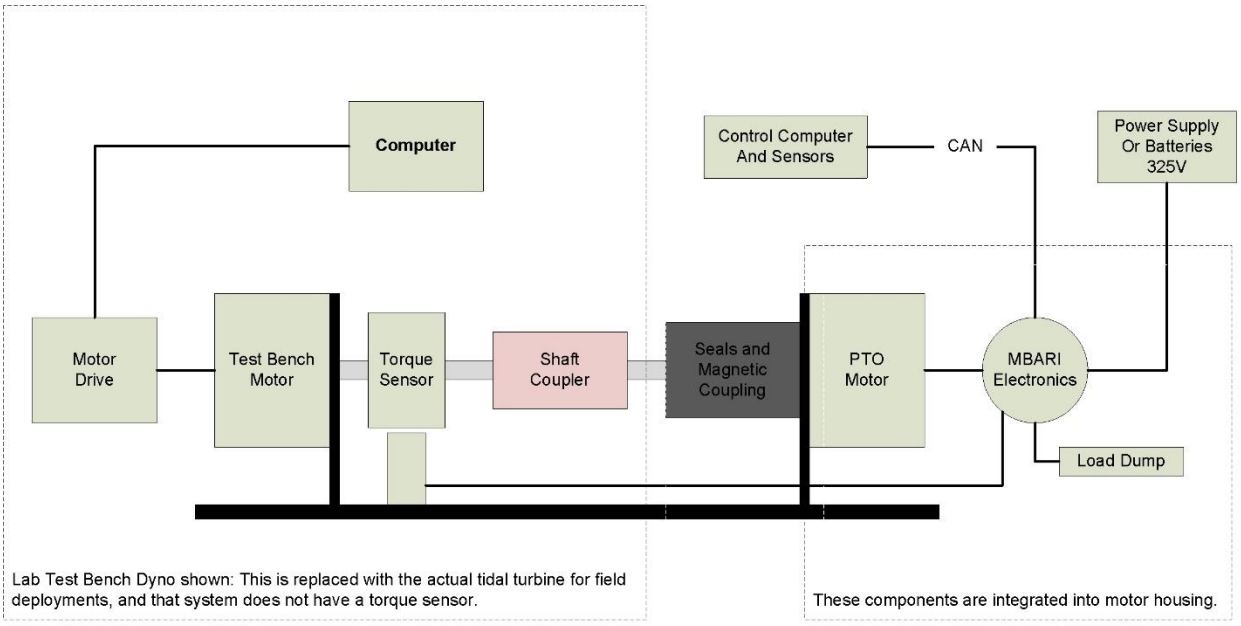


Figure 1: Block diagram of power converter system (on right) attached to lab-test dyno (on left).

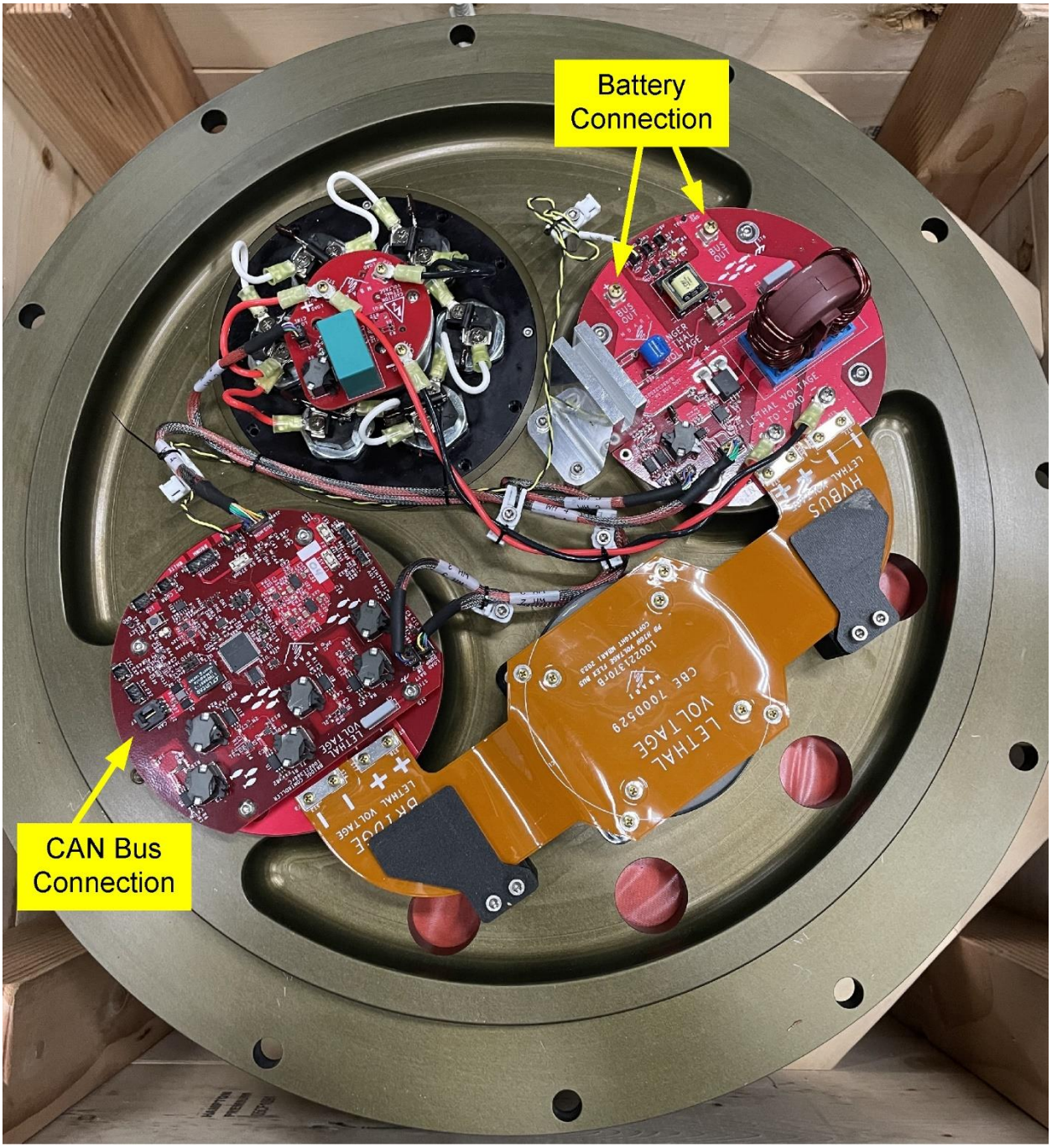


Figure 2 shows the power-converter itself and illustrates where the battery connection and computer CAN-bus connections are made. The specifics of these connections are described in the next secaion and the connection required between the power electronics and the motor are outlined in Appendix A with alignment instructions.

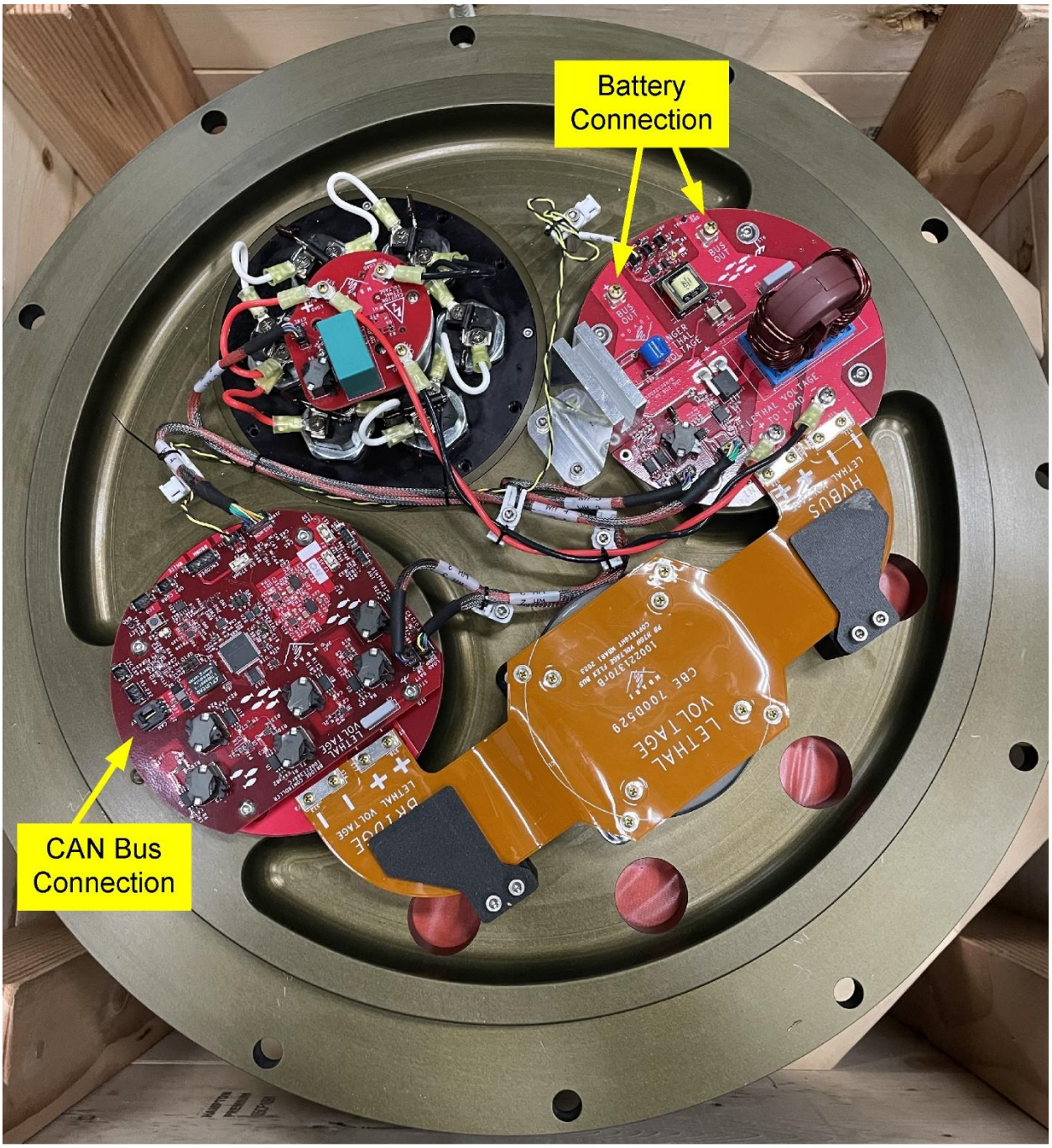


Figure 2: Power electronics mounted inside of the UofW/APL Tidal Turbine motor-housing end-cap.

Electrical Connections

Specifics of the external connections to a battery and computer that are needed to operate the system are outlined here. Additional details related to the motor and encoder connections and required alignment procedures are outlined in Appendix A.

Battery System:

For most operations, a power source needs to be connected to the system, either a battery or for testing a diode-protected power-supply set below 325 Volts can be used. The power-converter is currently set up to sink/source a maximum of 12 Amps, and wiring/connectors should accommodate this. For RFI considerations the motor frame must be bonded to the Bridge PCB's earth ground connection. This should be done with a minimum of 12awg wire from the motor frame to the ST1 screw on the controller PCB, minimizing run length. For safety reasons the motor frame also needs be bonded to earth ground.

Figure 3 illustrates the HV Bus output board to which the batteries are connected. This board performs bus filtering, in rush limiter and Power output cutoff switches along with bus voltage and current sensing.

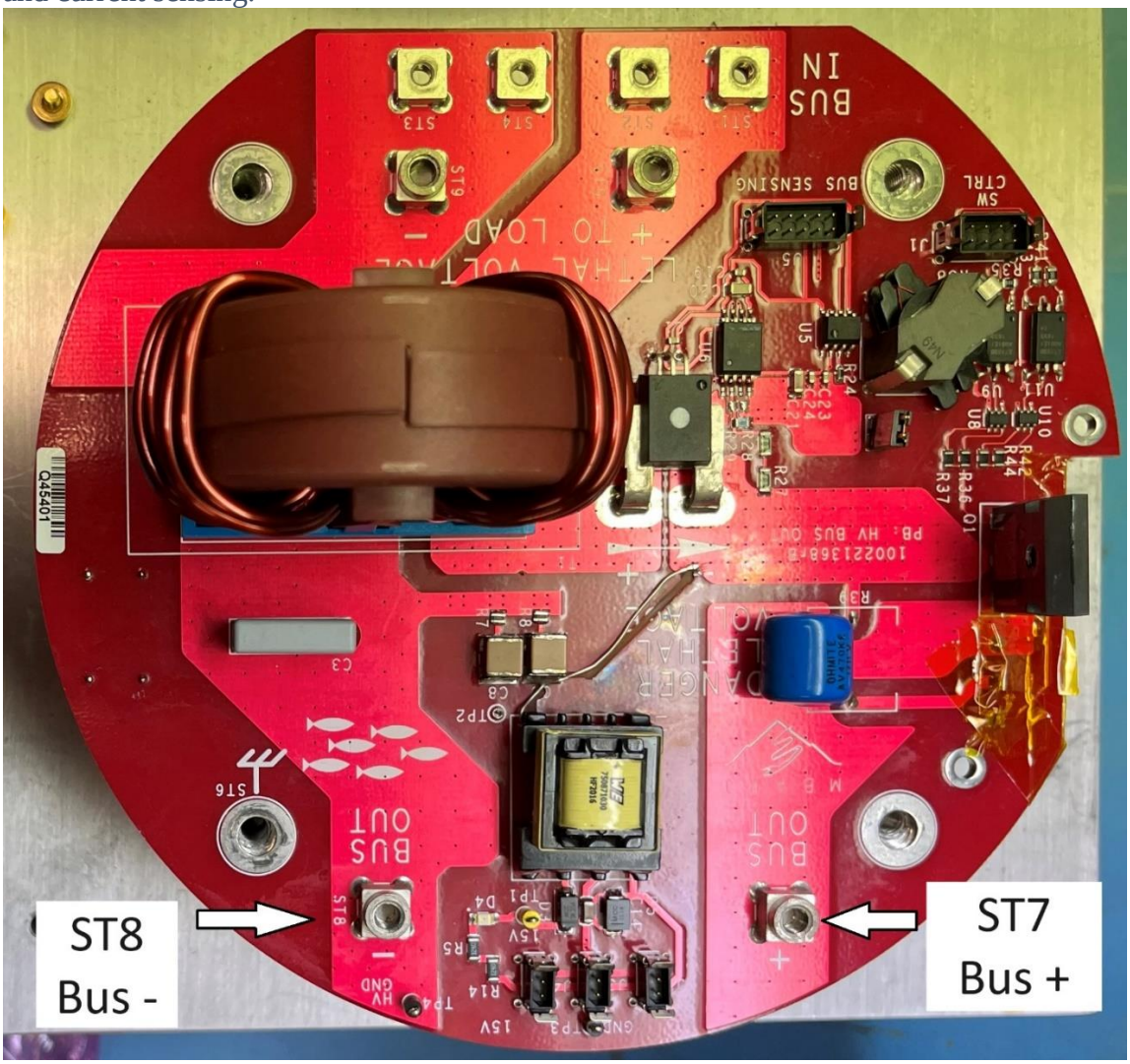


Figure 3 HV Bus Output PCB. ST7 and 8 are M4 screws

CAN Bus:

Computer communications are performed over a CAN bus running at 125000 baud. The transceivers in the power-electronics are isolated and require three connections, CAN-H, CAN-L, and a reference, present on the J2 connector on the controller board.

J2 CAN - Molex C-Grid mating connector PN: 0050579403

Pin	Description
1	CANH
2	CANL
3	CAN Ref

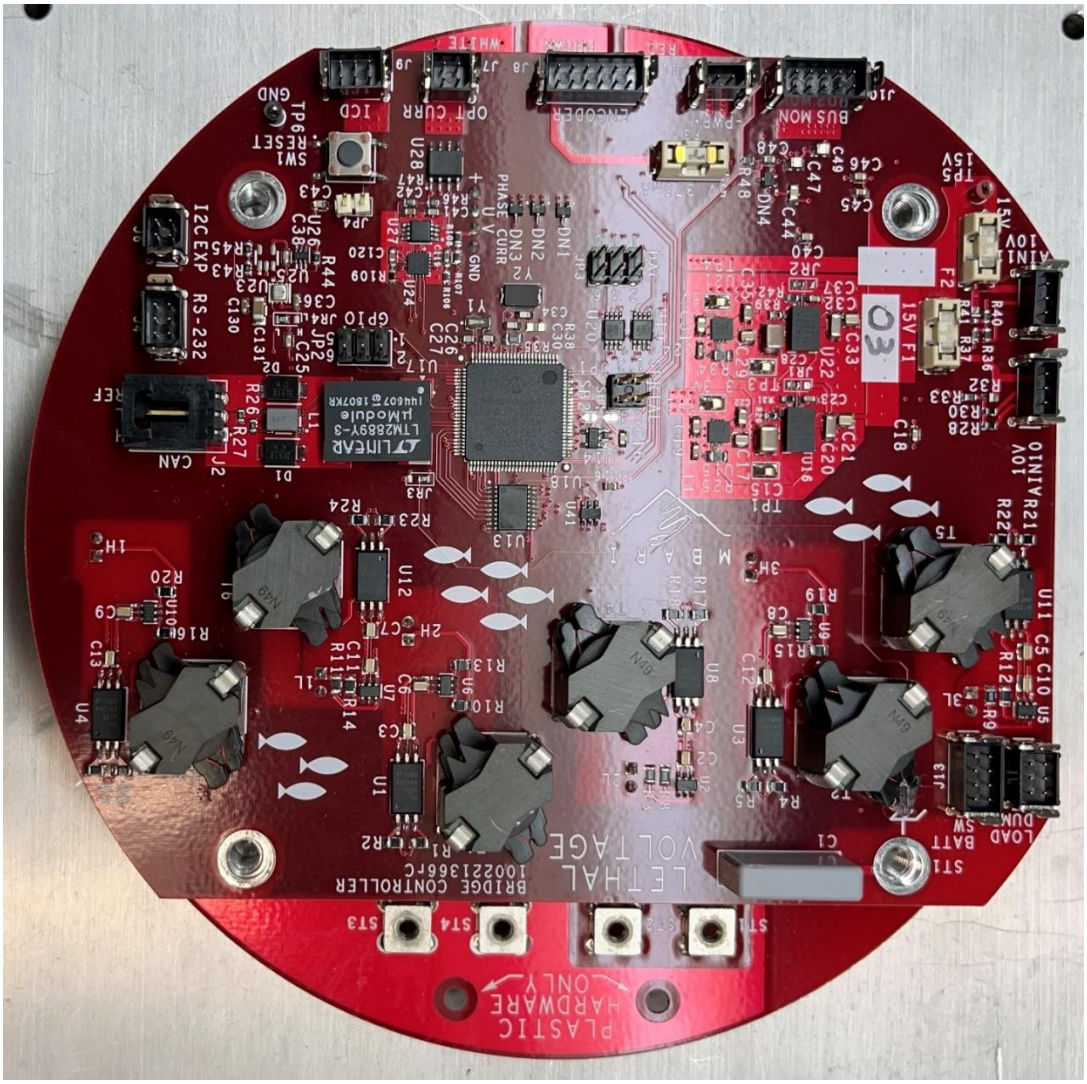


Figure 4: Controller Board, CAN connection is J2.

Operation

The fundamental role of the power converter is to control the current in the motor windings to achieve a specified quadrature current that results in torque of the desired magnitude and direction (CW/CCW). Depending upon the motor speed and direction, this results in either

current being drawn from the batteries, or flowing into the batteries (four quadrant operation). Therefore, any given time there is a “target” winding current and a resulting actual winding current. Most of the time these match closely, but in some cases it is not possible to achieve the target winding current for a variety of reasons, and the converter does it’s best within a set of constraints, described below.

The target winding current can have three origins; a default that is based upon the motor RPM, a user-specified request for a specific target current, or is derived from an internal motor speed control loop that adjusts the target current to maintain a specified rotor speed. In all cases the requested target current is subject to some limitations before being used as a setpoint in the converters internal current control loops. In particular, a maximum winding current target of +/- 30 Amps is enforced. For example, if the user requests a winding current of -40A, that request is immediately changed to -30A. Also, at higher speeds, these target current limits are further modified to drive the converter towards the generator quadrants as shown in Figure 5. This effectively implements an RPM limit on the system, as the system spins faster and faster, it becomes impossible to continue to add torque in that direction (at 130 RPM) and forces the converter to be a generator and resist rotations above that speed. This limitation always applies, regardless of the origin of the target winding current request.

There are additional possible limitations that can cause the target winding current to not be met by the converter. In the motor quadrants, the ability to exert high winding currents at high speed is limited by the power available from the power source. The converter implements a feature that limits how much current can be drawn from the battery system, the default is 8 Amps and this can be set up to 12 Amps by the user. If larger battery currents are required to meet a target winding current, the winding current request is reduced. This is shown as power-limited arcs in the motoring quadrants of Figure 5. Note that there is a similar limitation on the battery charge currents enforced by the converter to protect the battery and wiring, but this does not limit the winding currents because the integrated load-dump has the capacity to absorb all of the available power.

The final case in which the target winding current will not be implemented is in the small “box” at the origin of the speed-current graph. Small winding current requests (< .1A) will be ignored when the motor is not spinning, this allows the switching converter to be shut down when there is no motion, saving significant power when the motor is not spinning. It is only a small inconvenience because larger current requests cause the switching converter to turn on and implement the requested current.

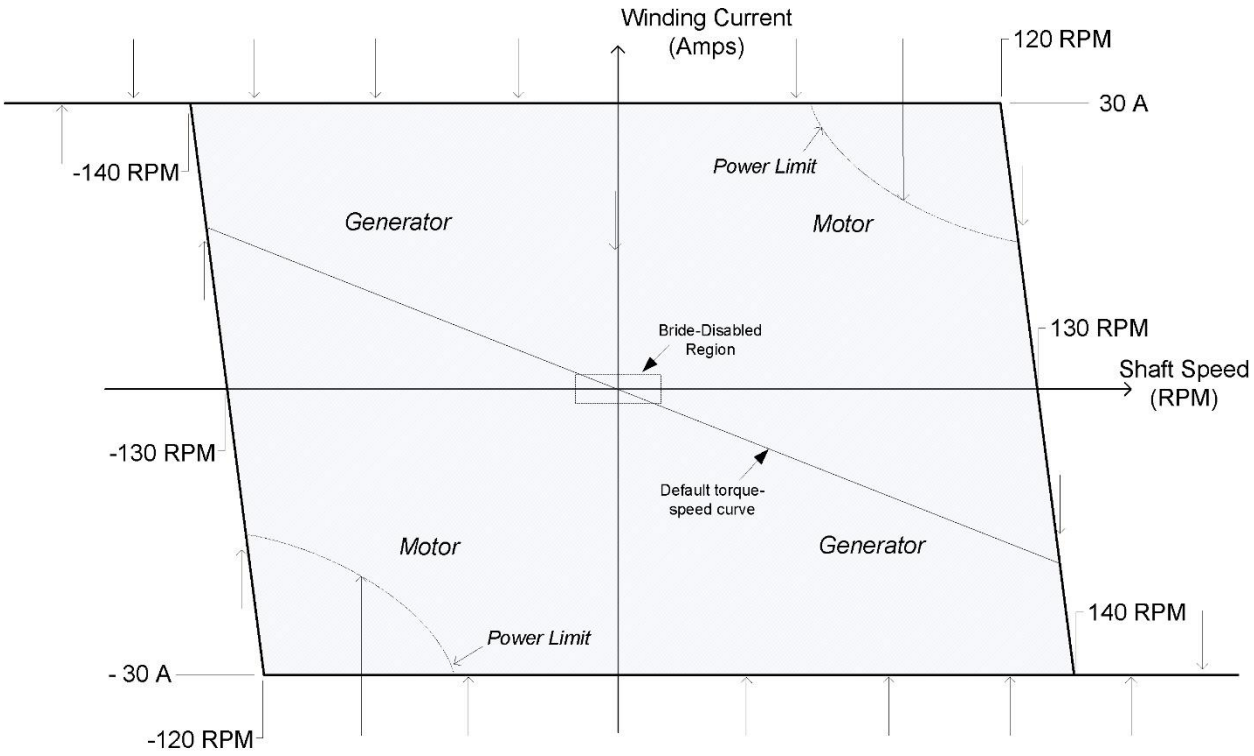


Figure 5

Because the system is designed for safe autonomous operations in which the connection to the external computer can not be always relied upon, there is a 2 second timeout on user-requested winding currents, after which the system will revert to the default torque-speed curve illustrated in Figure 5. This default curve always lies entirely within the generator quadrants of the system and therefore limits the speed of the system in all cases, regardless of if there is external battery voltage available. Note that this default torque/speed curve is subject to the same maximum winding current limits discussed above, and is always enforced unless a user-supplied target winding current command has been recently requested. The effect is that for an external control computer to maintain a specific winding current, it must issues winding current target requests at greater than 0.5Hz.

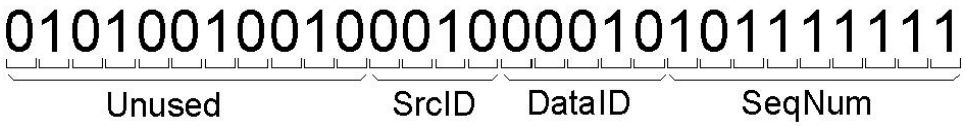
Finally, the winding current target can also be driven by an internal speed control loop that adjusts the winding currents to maintain a desired rotor speed supplied by the external computer. In order to keep the system from being stuck at a certain speed if the external computer is disconnected, this speed request is subject to a 10 second timeout, after which the system resorts to its default speed/torque curve. To maintain a specific rotor speed the external computer must issue speed requests at greater than 0.1Hz.

All communication with the converter is performed over the CAN bus, available commands to control the converter system are described below. The system also synchronously emits telemetry data on the CAN bus that captures the state and performance of the system at all times, this is described in the telemetry section below.

CAN Telemetry

The CAN Bus protocol used here is specific to this device, and uses extended ID CAN packets that have a 29 bit header with up to 8 bytes of data following that. Three pieces of information

are encoded in the least significant 18 bits of the header, with the remaining bits currently unused.



The controller streams eight-byte data packets and that include a variety of telemetry data. Each packet consists of a header and 8 data bytes. At a fixed rate of (10 Hz default), four such packets are generated and sent, each with a different DataID that indicates what data is included in the packet as described below. To allow the receiving software to reassemble the snapshot of telemetry data, each of the four packets from a given time have the same sequence number (SeqNum), that increments at the packet rate (10Hz default) before rolling over back to zero after 2^9 iterations. For data from the power converter, the SrcID is always 2, differentiating these packets from other data that may be present on the CAN Bus such as a battery system or other component.

The data items themselves are described as follows, grouped by DataID:

DataID = 0

(bytes 0-1) PC_RPM: This is the rotor RPM. 1 cnt = 0.5RPM
(bytes 2-3) PC Bus Voltage (V): System Bus Voltage measured by the controller. (1 cnt = 0.01V)
(bytes 4-5) PC_Winding Current (A): Instantaneous measured winding quadrature current.
(bytes 6-7) PC Battery Current (A): Instantaneous current flowing to (+)/from(-) the batteries. (1 cnt = 0.002 Amps)

DataID = 1

(bytes 0-1) PC Status: 16 bit status word (described below).
(bytes 2-3) PC Load Dump Current. (1 cnt = 0.002 Amps)
(bytes 4-5) Target Voltage, internal shunt regulator target voltage (1cnt = 0.01V)
(bytes 6-7) Target Current, Winding Current target voltage (1 cnt = 0.002 Amps)

DataID = 2

(bytes 0-1) Diff Pressure Sensor: On-board pressure sensor (unused in the UofW/APL system)
(bytes 2-3) RPM Std Dev (RPM): Standard deviation of RPM measured over the past 5 minutes.
(bytes 4-5) PC Scale: User set-able scale factor that is applied to default winding-current/RPM curve. (1 cnt = 1%)
(bytes 6-7) PC Retract: Unused in the UofW/APL system (1 cnt = 1%)

DataID = 3

(bytes 0-1) Aux Torque Sensor: On-board torque sensor input (unused in the UofW/APL system)
(bytes 2-3) Bias Current (A): Constant winding current applied to default current/RPM relation.
(bytes 4-5) Charge Current Limit (A): User settable battery charge current limit.
(bytes 6-7) Draw Current Limit (A): User settable battery draw current limit.

CAN Commands

Commands are sent to the power converter as individual CAN packets with the SRC_ID field representing the power converter (02), the data ID set to 10, and an arbitrary sequence number. The first data byte (of eight possible in a CAN Packet) is the Command ID and subsequent data bytes are the command data payload.

The commands relevant to the UofW/APL Power converter are described here, the next section includes some relevant value definitions.

Set CAN Packet Rate: CmdID = 35 (unsigned 8 bit integer as argument)

This sets the frequency at which the CAN data telemetry is streamed in Hz, defaults to 10 at power-up which is 10 4-packet telemetry ensembles per second (each ensemble with a different sequence number as described above).

```
/*SetCanPacketRate:  Cmd ID = 35
 * Input: takes single unsigned  byte from CANbus.
 * Scaling: 1 Count per Hz.
 * Upper Limit (Engineering Units):
 * Lower Limit (Engineering Units):
 * Out of Range Behavior:  Set to Min or Max allowable if out of range
 */
```

Set Enable: CmdID = 39 (signed 16 bit integer as argument)

This command enables the drive and must be issued before the system will control any current in the windings. The argument is a winding current that causes the system to rotate slowly until the index pulse of the encoder is seen. Only after this index pulse is seen can will the converter operate. In the case the forcing is sufficient to rotate the motor until the index-pulse is seen, this command is not necessary and the system will go into the default generator winding-current/RPM mode, but this command is a way to rotate the motor without external forcing.

```
/*SetEnable: CmdID = 39
 * Input: takes signed int from CANbus (two bytes).
 * Scaling: CRSF_CURRENT cnts per Amp.
 * Upper Limit (Engineering Units):  MAX_SLOWROTATE_CURRENT Amps
 * Lower Limit (Engineering Units):  MIN_LOWROTATE_CURRENT Amps
 * Out of Range Behavior:  Set to Min or Max allowable if out of range
 * */
```

Set Winding Current: CmdID = 31 (signed 16 bit integer as argument)

This sets the target winding current that the controller tries to achieve, subject to the limitations described above. This mode has a 10 second timeout and the converter reverts to Generator mode if a new velocity command isn't received in that time. More detail in the following code comment.

```
/*SetWindCurrent:  CmdID = 31
 * Input: takes signed int from CANbus (two bytes).
 * Scaling: CRSF_CURRENT cnts per Amp.
 * Upper Limit (Engineering Units):  MAX_WINDCURRENTLIMIT Amps
 * Lower Limit (Engineering Units):  -MAX_WINDCURRENTLIMIT Amps
 * Out of Range Behavior:  Set to Min or Max allowable if out of
range
 */
```

Set RPM: CmdID = 37 (signed 16 bit integer as argument)

This sets the target RPM and places the converter in Velocity Mode. This mode has a 30 second timeout and the converter reverts to Generator mode if a new velocity command isn't received in that time. More detail in the following code comment.

```
/*SetRPM:  CmdID = 37
 * Input: takes signed int from CANbus (two bytes).
 * Scaling: CRSF_RPM cnts per RPM.
 * Upper Limit (Engineering Units):  MAX_USERCOMMANDEDRPM
 * Lower Limit (Engineering Units):  -MAX_USERCOMMANDEDRPM
 * Out of Range Behavior:  Set to Min or Max allowable if out of range
 */
```

Set Scale Factor: CmdID = 21 (unsigned 8 bit integer as argument)

This sets the scale factor that is assigned to the default Winding-current/RPM relationship. More detail in the following code comment.

```
/*SetScaleFactor:  CmdID = 21
 * Input: takes single unsigned byte as argument from CANbus.
 * Scaling: CRSF_PERCENTAGES cnts per unit factor.
 * Upper Limit (Engineering Units):  MAX_SCALE_FACTOR
 * Lower Limit (Engineering Units):  MIN_SCALE_FACTOR
 * Out of Range Behavior:  Set to Min or Max allowable if out of range
 */
```

Set Battery Switch Mode: CmdID = 25 (unsigned 8 bit integer as argument)

This turns on and off the transistor that allows current to flow to the batteries from the converter. This switch also requires 160V to be present on the battery connection of the convert to allow this switch to turn on. This is to ensure current doesn't flow out of the converter in the case of a cut cable (or disconnected batteries).

```
/*SetBattSwitchMode: CmdID = 25
 * Input: takes unsigned single byte from CANbus.
 * Scaling: None
 * Valid Values:  0 = Off, 1 = On
 * Out of Range Behavior:  Only sets if argument is 0 or 1
 */
```

Set Target Voltage: CmdID = 22 (unsigned 16 bit integer as argument)

This sets the maximum voltage the converter will use to charge the batteries. Defaults to 325V, but adjustable here depending upon the batteries used. More detail in the following code comment.

```
/*SetTargetVoltage:  CmdID = 23
 * Input: takes unsigned int from CANbus (two bytes).
 * Scaling: CRSF_VOLTAGE cnts per Volt.
```

```
 * Upper Limit (Engineering Units):  MAX_TARGETVOLTAGE Volts
 * Lower Limit (Engineering Units):  MIN_TARGETVOLTAGE Volts
 * Out of Range Behavior:  Set to Min or Max allowable if out of range
 */
```

Set Maximum Battery Charge Current: CmdID = 24 (unsigned 16 bit integer as argument)

This sets the battery charge current limit, excess energy will be diverted to the internal load-dump to achieve this limit. Defaults to 8A, but adjustable here depending upon the batteries and wiring used. More detail in the following code comment.

```
/*SetMaxBattChargeCurrent:  CmdID = 24
 * Input: takes unsigned int from CANbus (two bytes).
 * Scaling: CRSF_CURRENT cnts per Amp.
 * Upper Limit (Engineering Units):  MAX_BATTCHARGECURRENTLIMIT Amps
 * Lower Limit (Engineering Units):  MIN_BATTCHARGECURRENTLIMIT Amps
 * Out of Range Behavior:  Set to Min or Max allowable if out of range
 */
```

Set Maximum Battery Draw Current: CmdID = 29 (unsigned 16 bit integer as argument)

This sets the battery draw current limit. This and the battery voltage establishes a power limit that may limit the winding current achievable in the motoring quadrants. Defaults to 8A, but adjustable here depending upon the batteries and wiring used. More detail in the following code comment.

```
/*SetMaxBattDrawCurrent:  CmdID = 29
 * Input: takes unsigned int from CANbus (two bytes).
 * Scaling: CRSF_CURRENT cnts per Amp.
 * Upper Limit (Engineering Units):  MAX_BATTDRAWCURRENTLIMIT Amps
 * Lower Limit (Engineering Units):  MIN_BATTDRAWCURRENTLIMIT Amps
 * Out of Range Behavior:  Set to Min or Max allowable if out of range
 */
```

Helpful C-code

```
#define PREINDEXPULSE_MODE 0 // Slow rotation mode until index pulse is seen.
#define GENERATOR_MODE 1 // Torque is selected to follow prescribed Speed-Power relation.
#define TORQUE_MODE 2 // User-Specified Motor Torque
#define VELOCITY_MODE 3 // User-Specified Motor Velocity (RPM)

#define DEFAULT_CAN_PACKET_RATE 10
#define MIN_CAN_PACKET_RATE 10
#define MAX_CAN_PACKET_RATE 50

#define DEFAULT_SCALE_FACTOR 1.0
#define MAX_SCALE_FACTOR 1.4
#define MIN_SCALE_FACTOR 0.5

#define DEFAULT_TARGETVOLTAGE 325  //Volts, power-up setting
#define MAX_TARGETVOLTAGE 330  // Volts, limit on user setting.
#define MIN_TARGETVOLTAGE 220   // Volts, limit on user setting.

#define DEFAULT_BATTDRAWCURRENTLIMIT 8.0
#define MAX_BATTDRAWCURRENTLIMIT 12.0  // Limit on user setting
#define MIN_BATTDRAWCURRENTLIMIT 2.0  // Limit on user setting

#define DEFAULT_BATTCHARGECURRENTLIMIT 8.0
#define MAX_BATTCHARGECURRENTLIMIT 12.0 //Limit on User Setting.
#define MIN_BATTCHARGECURRENTLIMIT 2.0 //Limit on User Setting

#define MAX_WINDCURRENTLIMIT 30
#define MAX_USERCOMMANDEDRPM 125 //Max/Min Command-able RPM

#define MAX_SLOWROTATE_CURRENT 12.0 //Current applied to slowly rotate rotor at enable until Index Pulse is found.

//Command and Reporting Scale Factors (cnts/engineering unit)
#define CRSF_RPM 2        //cnts per RPM
#define CRSF_VOLTAGE 100  //cnts per Volt
#define CRSF_CURRENT 500  //cnts per Amp
#define CRSF_PERCENTAGES 100    //cnts per 100 percent

// CAN ID field width definitions, from least sig to most sig.
#define SEQ_WIDTH 9
#define ID_WIDTH 5
#define SRC_WIDTH 4
```

```
//Assemble 29 bit id from components, 4 bit SRC, 5 bit dataID, and 9bit
seq_num
unsigned long Assemble_packetID(unsigned int src, unsigned int dataID,
unsigned int seq_num)
{
 unsigned long result = 0;
 result = ((unsigned long) src) << (SEQ_WIDTH+ID_WIDTH);
 result |= ((unsigned long) dataID) << SEQ_WIDTH;
 result |= (unsigned long) seq_num;
 return result;
}



struct PowerConverterStatusBits {
    unsigned int Mode : 2; // controller mode.  (PREINDEXPULSE_MODE,
GENERATOR_MODE, TORQUE_MODE, VELOCITY_MODE)
    unsigned int TorqueCmdLimited : 1; // defines what is limited the Torque
Command (0 = Not limited, 1 = Rate, 2 = Min, 3 = Max).
    unsigned int BattSwitchRequest : 1; // Indicates state of battery switch
that the user desires. 0 = off, 1 = on
    unsigned int BattSwitchSetting : 1; // Indicates instantaneous setting of
Battery Switch
    unsigned int SlowSwitchSetting : 1; // Indicates instantaneous setting of
Battery Slow Switch
    unsigned int EXT_Voltage_Detect : 1; //Indicates if >170V is available on
outside connector
    unsigned int GainScheduleMode : 1; // 0 = off.  1 = on.
    unsigned int OverCurrentShutdown : 1; //1 indicates drive is in
overcurrent shutdown.
    unsigned int OverVoltageShutdown : 1; //1 indicates drive is in
overvoltage shutdown.
    unsigned int SpringRangeValid : 1;  //1 indicates the current SC_Range
value is valid (received within approximately SC_RANGE_VALID_TIMEOUT
milliseconds)
    unsigned int PermissiveMode : 1;   //1 indicates user has selected
"permissive mode" which allows WindingCurrent commands to be set even if
SpringController Range packets aren't arriving
    unsigned int DriveEnable : 1;   //1 indicates user has enabled the
transistor drive to turn on.
};

typedef union
{
    unsigned int word;
    struct PowerConverterStatusBits bitfields;
} POWER_CONVERTER_STATUS;
```

Specifications

Absolute Maximums

Bus Voltage: 450 VDC
Motor winding current: 50 Amps
Bus Current: 35 Amps

Nominal

Bus voltage: 80 – 350 VDC
Motor Winding current: 30 Amps

Appendix A

This section outlines the connections between the motor and power electronics along with some information needed to correctly align the encoder with the motor phases. This includes the three motor phases and encoder connections.

Motor Phase connections:

The three motor phase connections are attached to the Bridge Board PCB, shown in Figure 6 and is located under the controller board in the assembled device. The ordering of the phases is important and described below in the encoder alignment section.

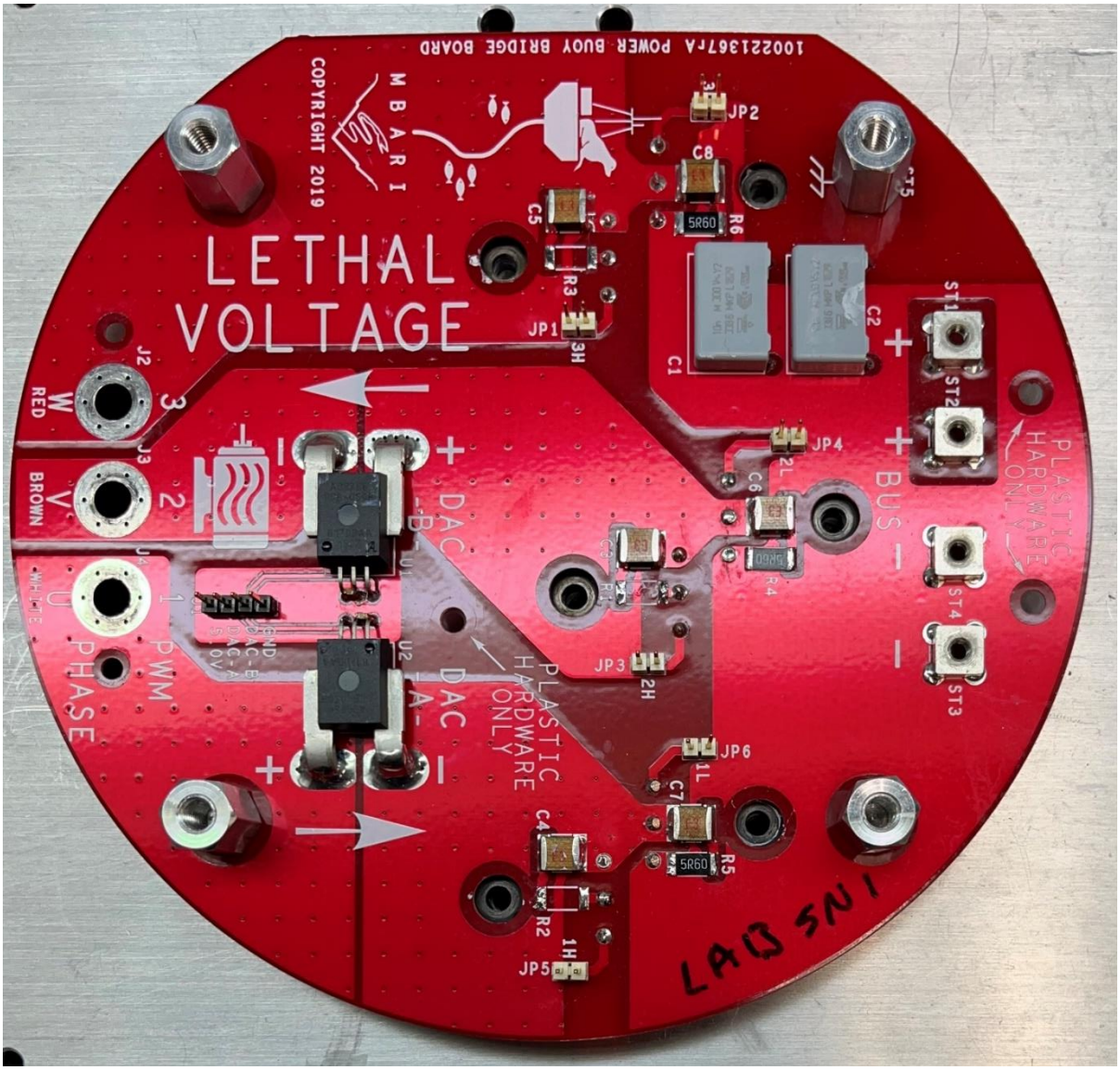


Figure 6: Bridge PCB, motor connections are shown on the left hand side.

Encoder Connection

The encoder connects to J8 on the controller PCB, shown in figure Figure 7. The mating connector for J8 is Harwin PN: M80-8881205 with the following pinout. The Hall connections are not required or used.

Pin	Description	Pin	Description

1	5V	7	Enc Z+
2	Gnd	8	Enc Z-
3	Enc A+	9	Hall W
4	Enc A-	10	Hall V
5	Enc B+	11	Hall U
6	Enc B-	12	No Connect

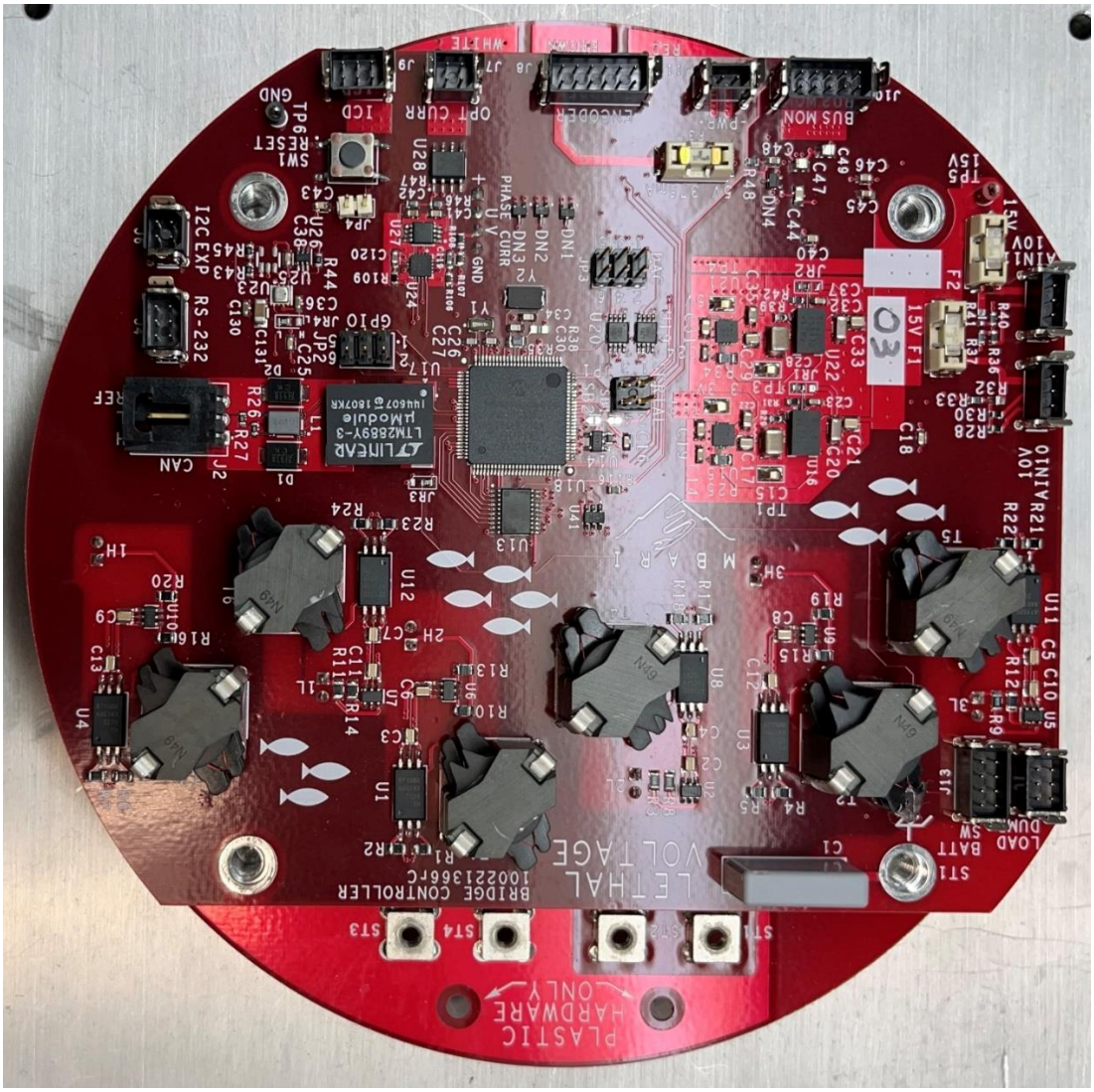


Figure 7: Controller PCB, this houses the processor, drives the bridge transistors, and manages the winding currents.

Phase and Encoder Alignment

The system uses the encoder as a basis for commutation and therefore encoder direction and Z-pulse location must be aligned in a specific manner relative to the phase windings. Figure 8 shows the relationship between the line-to-line voltage, hall sensors and the Z encoder pulse (One pulse per revolution). The line-line voltages are listed as signal-reference.

To align the encoder to the windings, attach the reference lead of an oscilloscope probe to the to-be-U phase and the scope probe to the to-be-W phase. Connect a second channel of the scope to

the Z encoder pulse. Spin motor slowly clockwise while facing the motor shaft and set the scope to trigger on the Z pulse. Adjust encoder rotation to align the U-W voltage zero crossing at the center of the Z pulse high time. The crossing should be within 2x the Z pulse width from the center of the Z pulse.

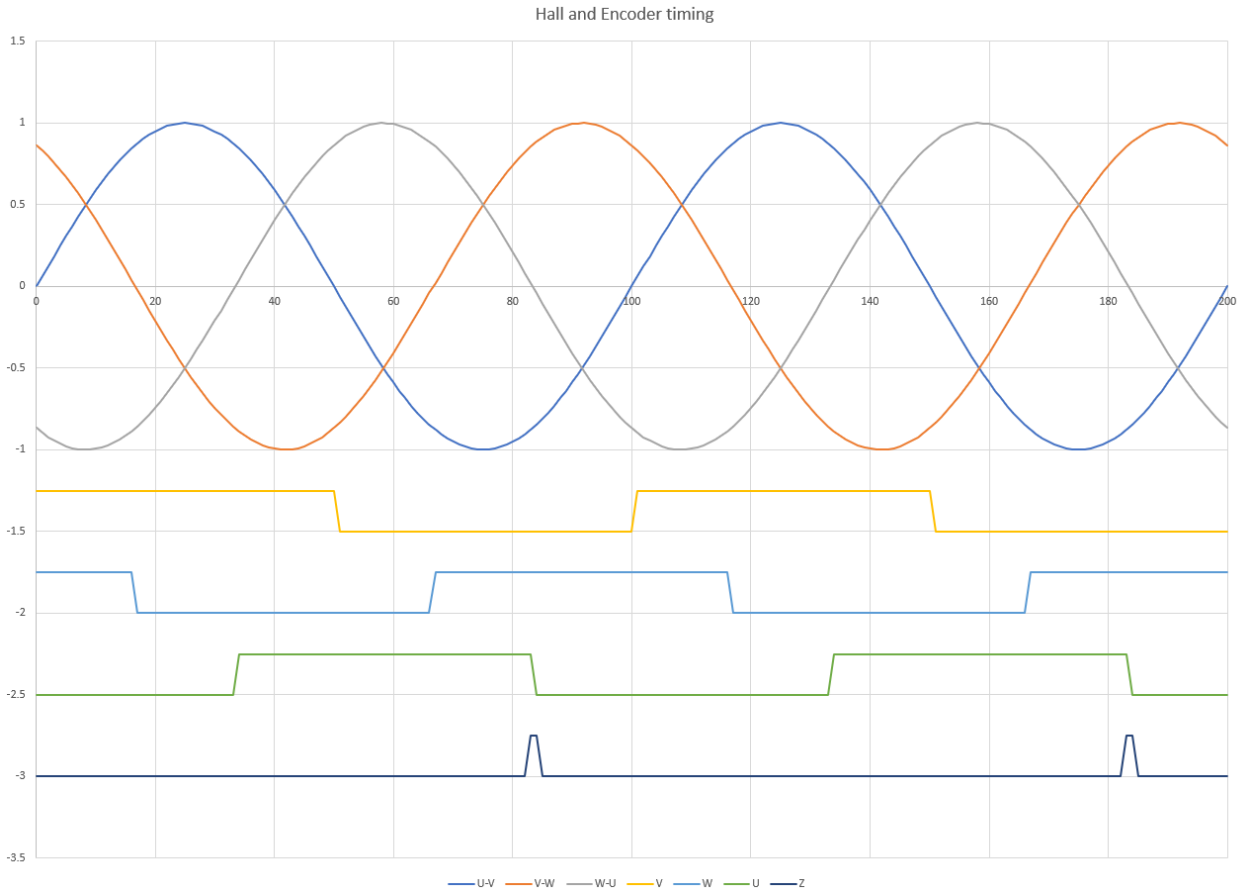


Figure 8 Winding and Encode phasing