Safety features

Author : Andras Tantos

Introduction

So far we've discussed a lot of the operational details of an H-bridge. We've seen what the basic operating modes are, how to select the power components, how to drive and how to control the bridge. This is enough to build and operate an H-bridge. And also enough to destroy it.

In this article I will concentrate on how to make the operation of the bridge safe. I will explore ways to safeguard the elements of the bridge itself. I will look into how to protect its power supply and the motor. I will also talk about what to do to protected the mechanical environment in which the bridge (and the motor) operates as well.

In many cases the described protection techniques are contradictory: if you want to protect the bridge for example, you might not be able to provide a safe mechanical state at the same time. Of course you can argue that a destroyed bridge won't provide much in terms of mechanical safety either, still judicious selection of which techniques to use and when is important and should be tailored to the specific application.

Creating lists is always dangerous as you bound to leave things out. Please let me know what's missing and I'll include it as an update!

Over-temperature detection

We've talked a lot about heat dissipation and techniques to minimize and evenly spread it across all the power elements of the bridge. Whatever you do however, it's unavoidable that – when used outside of the design parameters – the bridge would over-heat.

There are some run-away conditions inside a bridge that makes temperature-monitoring even more important: as the FETs heat up, their channel resistance (r_{dson}) gets higher. If they still delivering the same current – because of closed-loop current-control for example – the increased resistance will increase the heat-dissipation, which in turn will heat the FETs up even more. This positive feed-back eventually over-heat and destroy the bridge.

Detecting the temperature of the power elements (the FETs and the catch diodes) can be quite simple, especially if all of them are mounted on a single heat-sink. Simply attach a heat-sensor, like a [the TMP35 from Analog Devices](#) to the heat-sink and monitor the temperature either in the analog or (through an A/D converter or by choosing a digital-output sensor) in the digital domain. Sounds simple enough.

In practice attaching this extra element to the heat-sink is error-prone – what if it comes loose? –

and might not even be possible if the FETs have individual heat-sinks or no heat-sink at all in the case of surface-mounted designs. You can of course add four sensors, one by each FET, but that adds complexity.

There's another way though, if precision is not your primary concern: as we've discussed r_{dson} increases with temperature on the FETs. Consequently if you can measure the current through the FET and it's voltage drop (V_{ds}). You can figure out its resistance and from that you can determine its junction temperature. Even better: if all you need is to stay within safe limits, you can set your trigger on V_{ds} alone, and just assume the maximum allowed current all the time.

For example, if you use the IRF1010Z FET, it has 7.5m? on-resistance at room temperature, and roughly twice of that at 170°C:

If we want to limit the junction temperature to be below 100°C (to leave some design margin), we need to trigger some action when r_{dson} gets bigger than 1.5 times 7.5m?, or 11.25m?. If our bridge design allows for a maximum current of 30A over the FETs, this means that if V_{ds} gets above 337.5mV, we should start ringing alarm bells. If we have ways to measure the bridge current, we can set the voltage threshold depending on the current, and not need to depend on worst-case calculations.

As always, there are complications though: there are times in the operation of the bridge, when the FET doesn't conduct. During those periods – if the body-diode is off as well – V_{ds} can be quite high, and that's normal. So the monitoring should only trigger when the FET conducts.

The other problem is actually measuring V_{ds}: its value is rather low compared to the voltages that are normally present in the bridge, so the measuring circuit needs to have good common-

mode rejection capability.

The last problem of course is that four FETs need four monitors, each with their own amplifier, inhibit logic and trigger settings. This additional cost can easily offset the benefits.

Integrated solutions have an advantage here: when the power FETs are integrated onto the same die as the control circuit, it is possible to integrate a temperature monitor as well. This monitor measures the die temperature directly, so its more precise than any of the previous techniques.

Current detection

We've talked a lot about current over the bridge, now its time to talk about how to measure it.

In order to be able to measure the current, first we'll have to decide which current to measure and at what point in the H-bridge can we measure it.

We can measure the current through the load (the motor), or the bridge itself. We could also theoretically at least measure the current through any of the four switching elements, but that technique is rarely used. The reason is that current may or may not flow through any of the four elements, depending on the operating conditions, the phase of the cycle and the drive mode of the bridge. To properly measure current at all times, all four switching elements should be monitored. That is almost always too much of a burden to be troubled with.

This leaves us with either monitoring the load or the bridge current. If we were to monitor the load current, the current sense element needs to be put in series with the load:
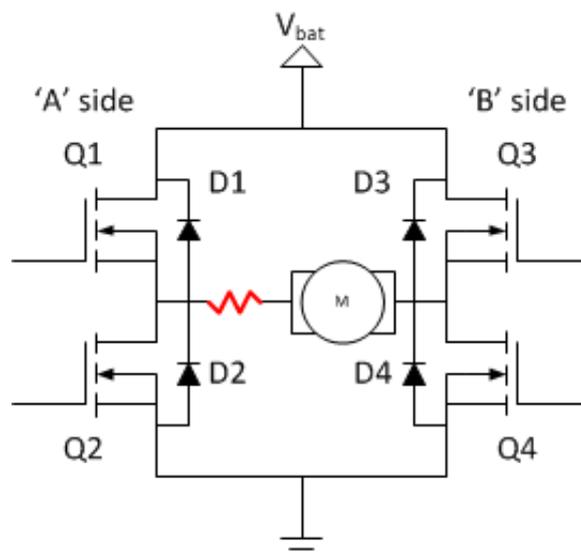
If we want to monitor the bridge current, we have two choices: either to put the sensor on the high-side of the bridge, between it and the battery, or on the low-side, towards ground:
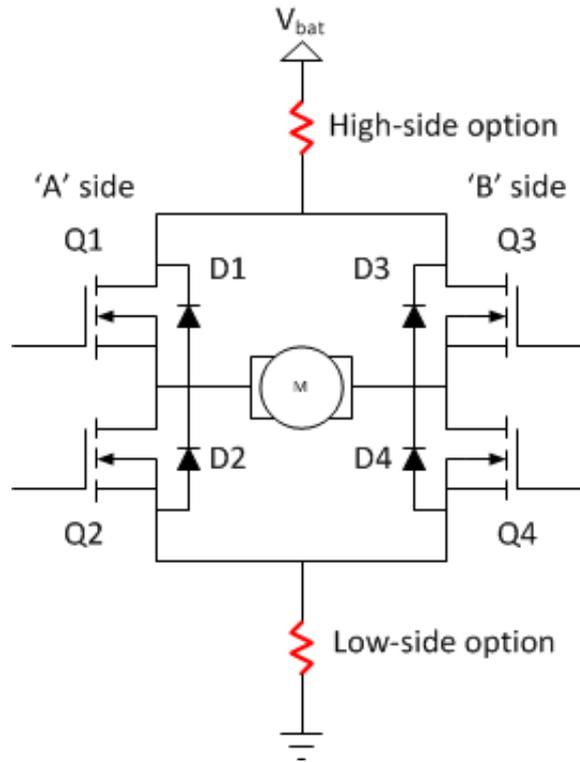
As far as actual monitoring circuits go, there are way too many techniques developed to list here. So instead, here's a good document summarizing the techniques: http://cds.linear.com/docs/Application%20Note/an105.pdf. Other techniques involve using hall-effect sensors and achieve very low insertion loss. Allegro MicroSystems makes such integrated devices.

While I didn't detail the circuits, It's worth noting the advantages and disadvantages of the detection locations:

Load Current Sensing

This approach is attractive as it measures current flowing through the motor both during the on- and the off-time. If we're only interested in the average current, a simple analog averaging circuit, like an R/C filter will do, and the monitor will not need to be concerned with the drive mode and the phase of the bridge drive.

There are several down-sides however. First, the circuit is a bit asymmetrical, which will be problematic when trying to detect short circuit conditions. For example, if the a-side wire to the motor gets loose and shorts to the chassis (which is grounded in many cases), the current sensor will correctly see the high current peaks during the turn-on time of Q1. However if the b-side wire comes loose and short the same way, the current sensor is completely unaware of the problem.

Second, while the currents through the sensing elements are rather smooth, the common-mode voltages jump around quite a bit. The maximum voltage is a bit higher than V_{bat}, any time D1 conducts. The minimum value is slightly negative, can be as low as $-V_F$, in case D2 conducts. Both of these situations can in fact occur for a significant portion of the time in asynchronous sign-magnitude drive, but can happen momentarily in all drive modes during the shoot-through protection window. Finding a current sense amplifier that can properly handle these conditions is not easy, especially one with a single, positive power supply.

Third, this detection technique completely fails to detect switching element failures. MOSFETs are easily damaged by voltage spikes on their gates and when they do get damaged, they can close permanently. When that happens to – let's say – Q1, every time Q2 closes, it essentially creates a short circuit through Q1 and Q2. This in turn will eventually destroy even more stuff in the bridge or worse, overheat the battery. The current sensor is completely oblivious of this fact as the short-circuit current doesn't flow through the sensing element.

High-side Current Sensing

If you choose to implement high-side current sensing, your trade-offs shape up a little bit differently.

The main advantage is that any current that leaves the battery through the bridge is measured. It doesn't matter if it goes through the motor, through the chassis towards ground due to a loose wire or through the FETs due to internal damage, it is caught by the sensor.

The other advantage is that the common-mode voltage is always around V_{bat}, though still can be slightly higher in case of regenerative braking.

On the negative side, the current through the sensing element is 0 during the off-time for both sign-magnitude drive modes, and changes direction in lock-anti-phase drive. To correctly infer motor current, the circuitry (or the SW) will need to correlate current measurements with the phase of the bridge drive.

Another negative is that finding current-sense amplifiers with good-enough common-mode rejection ratio and high enough input voltage range can be a challenge – easier than in the previous case, but still not necessarily trivial.

The final slight problem is that short-circuits towards V_{bat} on the motor terminals is not detected by this circuitry. While in many cases that is not a very likely worry, in other cases it might be a concern.

Low-side Current Sensing

Finally, let's consider the low-side current sensing option!

The main advantage here is that the common-mode voltage across the sense-resistor is close

to 0 (could be slightly negative in case of regenerative braking). This is the easiest option to find a sense-amplifier for.

It also has the advantage, just as the previous case, that any current that gets into the battery is measured, independent of whether if flows through the load or not.

The main disadvantage is that this circuit can't detect shorts to ground (or grounded chassis) on the load: the short-circuit current flows around the sense element.

The other disadvantage carries over from the previous idea: the choppy current and the need to correlate current and drive phase to generate reasonable measurements.

Average v.s. peak currents

It's nice that we can measure current through the bridge, but how do we know if it's too high? While the answer of course could be as simple as 'if it's over a certain limit, it's too high', there is one thing to consider: the current through the bridge is not constant. That means that peak and average current might not be the same. Worse, the peak-to-average ratio depends on the inductance and the generator voltage of the motor, none of which is usually known precisely during design-time (or at all). The result is that knowing the average doesn't tell us much about the peak and vice-versa. Best is measuring both independently, if possible. This means however that the bandwidth of the current measurement circuit will need to be able to capture the current-changes throughout a cycle. It also means that – if digital domain measurements are done – the sampling frequency of the A/D converter will need to be set accordingly. As a rule of thumb you need to set the bandwidth of your current measurement circuitry to be at least three times as high as your switching frequency, if you measure the current through the load. If you measure the current on the low- or high-side, you need to set the bandwidth higher – due to the sudden changes in the current between the on- and off-times. Probably a 5x bandwidth is a reasonable starting point. This easily pushes your measurement bandwidth into the 100kHz range with a 20kHz switching frequency, further limiting your sense amplifier selection.

A final problem is that with increased bandwidth comes increased noise-sensitivity. Watch out as datasheets often specify noise numbers for a significantly lower bandwidth than the maximum supported by the device!

Why would you care about peaks though, you may ask? In fact, most of the problems arise from the average current being too high. While that's true, there are instantaneous current-limit specs on MOSFETs and violating those limits can destroy your device. Short-circuit conditions, where the load (that is the short-circuit) doesn't behave like an inductor any more, currents rise very rapidly, and by the time your bandwidth-limited current-sense circuit realizes the problem and reacts to it, your might be left with a broken MOSFET. You might get saved by the internal resistance of the battery, the FETs and the wiring but make sure you in fact can rely on those elements if that's your plan!

Take one example: let's say you used MOSFETs with a minimum r_{dson} of 5m?, your battery voltage is 20V, and your nominal motor current is 20A. You've used to 5m? current-sense resistor to get a 100mV differential voltage for your current-sense amplifier. You've sized your wires and connections so that their internal resistance is significantly lower than these numbers. Now, if you're unlucky enough that you get a short to GND on the load, during on-time, only one MOSFET and your current-sense resistor is in the way of the current flow with a total of 10m? of resistance. At 20V, your short-circuit current is 2000A! It's almost certain that your MOSFETs will not live through such a current-spike. Chances are of course that the internal resistance of the battery will limit your current to something lower, and your wiring inductance will make the appearance of this current somewhat gradual, but essentially you will get the short-circuit current of your battery flowing through your circuit. Can your bridge survive that?

Reverse-current protection

Whenever current flows from the motor to the battery, we have a reverse-current situation.

Just as with over-current detection, we'll have to differentiate two cases of reverse-current detection as well: permanent and transient events.

Transient reverse current

Transient reverse-currents naturally occur during the operation of the bridge in all drive-modes. As we've discussed before, the cause of the reverse current is the reversal of the battery voltage on the motor without the reversal of the flow of the current. We've talked about how to deal with this current as well for the various drive modes here, here and here: an appropriately sized input capacitor can soak up the extra charge and provide effective protection. With extreme low switching frequencies the size of the capacitor becomes problematic, but at ultrasonic frequencies, it is usually manageable.

Permanent reverse current

Permanent reverse current detection is not much harder than average over-current detection. Mostly the same circuitry can be used, the only complication is that the voltage drop on the sense-element will be negative. As a consequence, the sense amplifier will have to have appropriate input voltage range to handle these negative voltage drops.

Protection against permanent reverse-current is much harder to achieve though. It happens any time the bridge operates in regenerative braking mode. This situation can't occur with asynchronous sign-magnitude drive, only with the other two: sign-magnitude and lock anti-phase drive-modes. The problem is that this condition can stay around indefinitely, so the amount of energy pumped back into the battery can be arbitrarily large as well. No capacitor can handle this, and no battery is capable of storing infinite charge either. There are essentially three things we can do:

1. Disconnect the motor from the battery, and short-circuit it (by turning on both high-side

or low-side FETs). This will break the motor and dissipate the energy on its internal resistance. The main problem with this solution is that we have no control over the braking torque, we loose all control over the device. The braking torque is also speed-dependent and as the motor slows down, it decreases. Finally, the motor might overheat if the situation stays around long enough.

2. To regain some control over the motor, we can switch the bridge into asynchronous sign-magnitude drive mode if its not operating in that mode to begin with, in which mode we have no regenerative braking, but can control the braking torque. The problem here is that the braking current is flowing through the catch diodes, which need to be sized appropriately to be able to handle the dissipated heat.

3. We can switch in an artificial load at the battery side (a bank of resistors between V_{bat} and GND) to burn off the excess power. This artificial load will need be able to dissipate as much power as is pumped back by the motor in regenerative braking (and not used up by the additional loads) but not too much because it would than load the battery too much. In fact what you would want is this load to dissipate just as much as the motor pumps back, trouble is, that would require an adjustable load. To solve that, you can implement a time-division system where the load is in the system for a short while, than disconnected using a fast PWM signal. An appropriately sized capacitor on V_{bat} would make sure that the battery voltage is within safe limits during the off-time and doesn't droop too much during the on-time. You can also implement a variable load using a (bank of) transistors or FETs as well. Either way, it is important to provide adequate cooling for the artificial load: it converts the excess energy to heat. The motor can – depending on its size of course – easily generate several hundred watts of power.

What you shouldn't **ever** do is to completely disconnect the bridge from its load or the battery. As you disrupt the current flow, the motor voltage will sky-rocket and damage something. Circuit breakers and fuses do just that. They can protect your battery from exploding by disconnecting the load, but they can't protect your bridge. If you really really want to employ such an element, make sure you do the following:

- Your bridge control and drive circuitry is powered from a supply that's not interrupted by the fuse.
- You employ some detection circuit to detect if the fuse went off – you probably want to monitor the voltage across the fuse.
- If your bridge looses power, immediately turn at least one, if you can both lower-side FETs on to provide a path for the motor current to flow.

Torque-limiting

An electric motor can have a huge torque. We've seen before that it's torque is proportional to the current flowing through it. If the shaft speed is not allowed to change – so the back-EMF voltage is constant – the only thing limiting the current and the torque is the various internal resistances in the circuit: the motor itself, the bridge switching elements, the battery and the wiring. In some cases the resulting maximum torque is too high for the mechanical components connected to the motor to withstand. If you don't want to end up with broken shafts, stripped

gears or burned rubber, you want to limit the torque of the motor.

Luckily, just as I've said above by monitoring the motor current you can get a pretty good idea about the motor torque as well. For this application, you're interested in the average motor current. Any of the previously described monitoring techniques can be used.

If an over-torque situation is not very likely in your application, your reaction be the same as for over-current situations. If you expect to encounter too much torque on a regular basis, you will probably need to modify the control circuit to essentially switch to constant-current operation – we will talk about closed-loop control in a later installment.

Over-temperature protection

Bridge elements

On the surface over-temperature protection is rather simple: when the temperature of the protected elements (mostly the power MOSFETs and the catch diodes) is too high the bridge should be shut down. In practice however, blindly shutting an operational bridge down can be a rather bad idea: as we've discussed several times, if current is flowing through the bridge and the motor, simply opening all switching elements is a fast path to destruction. It's usually much better to short-circuit the motor through either the two high- or low-side FETs. This allows the motor current to safely reduce to zero, however it doesn't allow for any torque-control during the ramp-down and if there's external torque applied to the motor it doesn't even guarantee proper shut-down: the motor will enter generator mode and start pumping energy back into the circuit and more heat will be generated in it.

It might be a better idea to short-circuit the motor until its motor current is non-zero. As soon as the current reaches zero, open all four switches. This way the motor will loose all its drive – which in itself might be problematic – but at least the circuit is allowed to cool down. To implement this approach however the zero-crossing of the motor current needs to be detected quite accurately which is not necessarily easy to do. The less precise you detect the zero-crossing the more residual motor current the circuit will need to be able survive with all four switches open.

Motor

Whenever the motor is applying torque, the current through it will burn some power on its internal resistance. This happens during braking as well as acceleration. When the current is high, the motor can get quite hot. It is especially dangerous when the high-current situation coincides with low speed because the slow movement of the rotor will provide less than adequate air-flow to cool the coils down. To prevent the motor from over-heating, in some application monitoring the motor-temperature might be required as well.

When you try to protect the motor from over-heating, you essentially dynamically limit the motor-

torque depending on the motor temperature: at low temperatures you allow higher torque than at high temperatures. Of course you can be conservative and allow only as much torque as the motor can survive at any temperature but such design will have lower than optimal performance.

Over-voltage protection

An H-bridge interfaces with the external world at essentially two points: at the battery and at the load. No surprisingly these are the two points where too high voltages can be problematic and damaging.

We also have to differentiate between permanent and transient over-voltage situations.

Permanent over-voltage protection

Permanent over-voltage situations on the battery can mostly result from misuse or misconfiguration. Detecting slightly above the limit voltages is not that problematic: a voltage divider can bring the voltage back into the allowed range and during design we can ensure that the components have enough voltage margin to survive such situations without damage.

Detection of really high voltages (for example 120V instead of 12V) is a challenge however. These cases are rather rare under normal circumstances and mostly are a result of gross mishandling, so many designs take the approach of not caring at all and essentially taking the stance of 'well, don't do it!'. If you really have to do something about it, one possible technique is to convert the over-voltage condition into an over-current condition by using a high-voltage, high-current transient voltage suppressor (TVS) zener diode. The over-current condition can be handled by a fuse or a circuit breaker. You can read about this technique at length in this app-note (from page 44): http://www.onsemi.com/pub_link/Collateral/HBD854-D.PDF

Transient Over-voltage Protection

Transients – voltage spikes – can come from many sources, and can plague both the battery-side and the load-side of the H-bridge. Effective protection against them is important, as these spikes can easily destroy your power MOSFETs. Let's look at the sources of these spikes first!

On the battery side high voltage spikes can essentially come from other noisy elements on the same supply – for example another H-bridge.

We've already discussed one source of voltage spikes on the load side: during turn-on and turn-off, there's a short period when neither the FETs nor the catch diodes conduct, due to the internal capacitance of said diodes. This is not the only way though that voltage spikes can get into the system: they can be generated by the motors themselves due to commutation (http://en.wikipedia.org/wiki/Commutator_(electric)). I will get back to the consequences of this in a later article, for now, it's enough to understand, that when the commutator switches the motor windings around, it can momentarily disconnect them, while current is flowing. That's not a nice thing to do with what is essentially an inductor, and it punishes us with huge voltage

spikes, and sparks across the commutator to complete the current loop.

For all cases, there are essentially two things that can be done: we can either add a low-pass filter that filters out the high-frequency content of the spike, or we short-circuit it.

Creating a low-pass filter is a nice idea, but is problematic as the currents that flow through the filter are quite high, preventing us from using high series-resistances or inductances. This leaves us with using (relatively) large capacitors, but those have (again relatively) high series resistance and lead-inductance, limiting their effectiveness at high frequencies.

Still, adding a capacitor in parallel with the motor – and as close to motor leads as possible – is an effective way to keep the voltage spikes at bay, just as we've done that for the short interval between the FETs turning off and the catch diodes taking over:
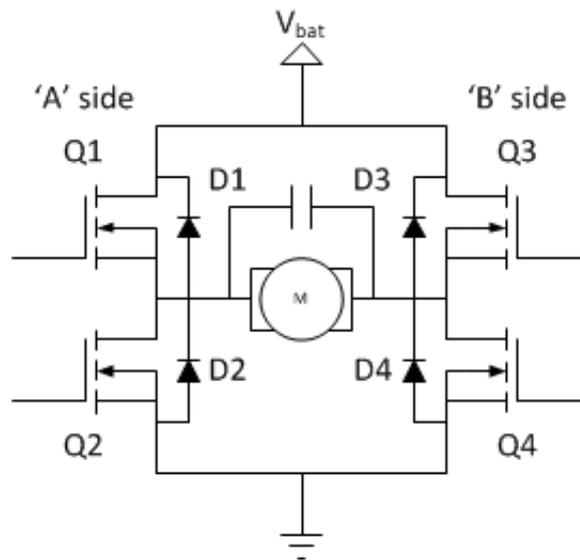
Adding a capacitor to the input of the bridge is also necessary for other reasons as well, namely to deal with the ripple-current through the bridge. This effect and the sizing of the input capacitor have been covered for the various drive modes here, here and here.
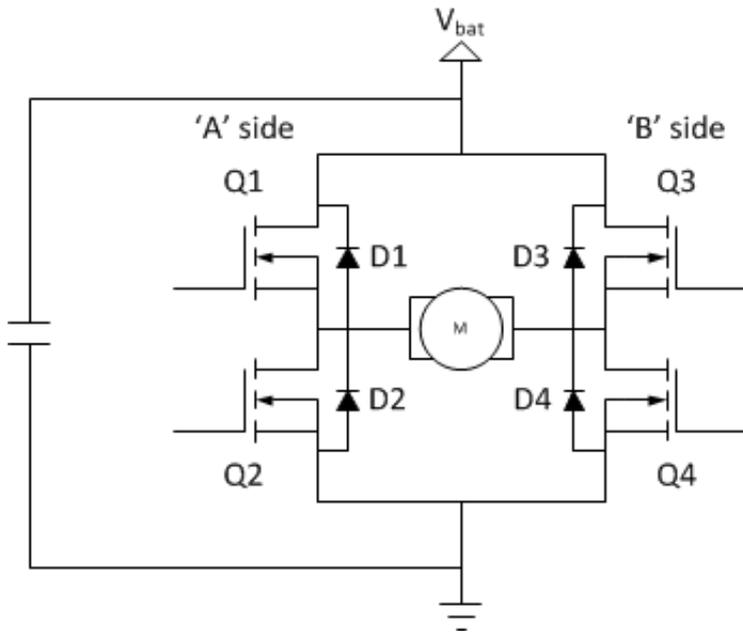
This takes care of the spikes generated by the bridge itself, but not necessarily of ones coming from the rest of the system. Without knowing what the source of such noise is of course its impossible to give a solution, but in general, its always better to deal with the noise at its source than try to protect the circuit that's affected by it.

This covers filtering techniques, but how about short-circuiting voltage spikes? Here, we can rely on 'snubbers', zener diodes, or TVS devices to take care of the problem. These however should be last resort solutions: they can be problematic to size well, slow to react and expensive.

Under-voltage protection

So far we've tried our best to limit the maximum voltage we can supply to the bridge (or receive from the load). Why would we care about too low of a voltage?

Well, there are a few cases where this might be a problem:

- With low voltages, the motor might not have enough torque to move its mechanical load
- The low battery voltage might result in too low Vgs voltage in the driver circuit, which in turn might not close the MOSFETs enough, increasing their on-resistance, and heat-dissipation
- If the low voltage is the result of a drained battery, it is also accompanied by increased internal resistance of the battery. When we try to turn on the motor – getting high currents out of the battery – the battery voltage might drop or even collapse, resetting, or even worse browning out other electronics it supposed to power.

Thankfully, detecting low voltages is relatively simple, many driver circuits already include UVLO

(under-voltage lockout) circuitry. If you need an external solution, many integrated solutions are available in the form of supply monitor and reset generator devices. If you really need a discrete solution, this app-note provides some ideas:
http://cdn.vicorpower.com/documents/application_notes/UV_OV_APPNote.pdf
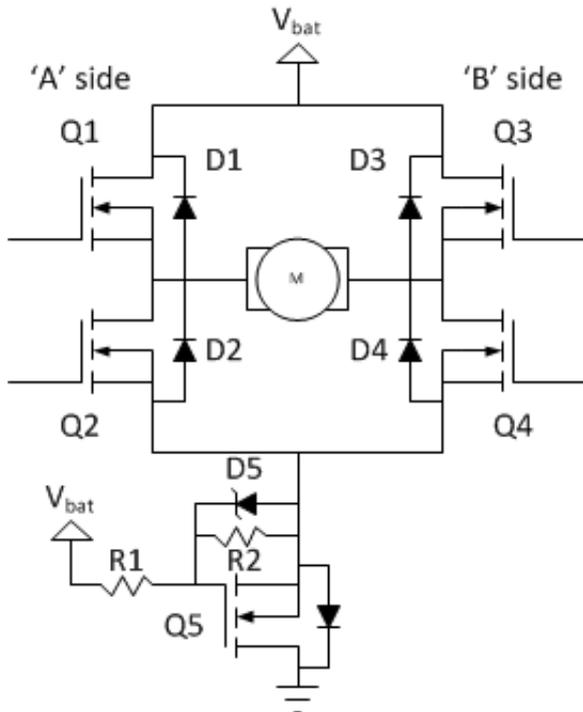
Inverse battery protection

Connecting a battery backwards to a bridge is a rather common, but – without protection – quite deadly mistake. The problem is the following: when the battery polarity is reversed, all four catch-diodes are forward-biased. This creates a very low resistance path for current to flow through the bridge, essentially short-circuiting the battery. Even worse, this large current will flow though the diodes which have a relatively high voltage-drop. This results in a lot of heat being generated, and destroying your power devices quite fast.

A fuse or a circuit breaker in series with your bridge is one of the simplest protections against this problem, but they must be fast-acting to have at least a chance at being effective.

As we've discussed before, having a circuit breaker or fuse in series with the bridge is not without its own dangers either, so if you want something safer (and faster-acting), a series MOSFET with its gate connected to the opposite side of the supply can work:
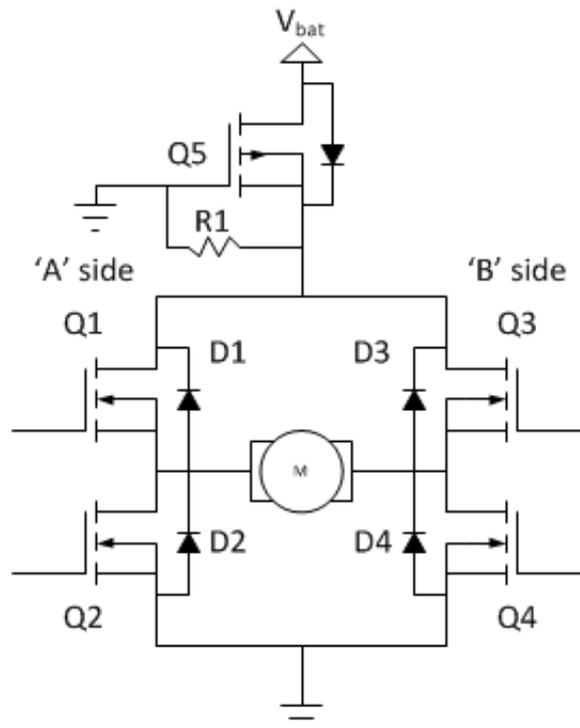
The idea is that when the battery is connected with normal polarity, it will forward bias the body-diode of Q5, driving its source to V_{bat}. When that happens, the grounded gate will close Q5 (it's a P-channel device), making it capable of handling large(er) currents.

When the battery is connected in reverse, the diode is reverse-biased, leaving the gate and the source of Q5 at the same potential, keeping the Q5 open, and the rest of the system un-powered.

The value of R1 should be high (100k or so) because in normal operation it's connected between V_{bat} and GND. Finally, if V_{bat} can be higher than the maximum allowed V_{GS} for Q5 (15-20V normally) limiting the gate-source voltage will be needed:

Here, under normal operation D5 will limit V_{GS} to be within safe limits with some current flowing through R1. In reverse operation, R2 ensures that Q5 can't close. The value of R1 should be select to set the appropriate zener current (a few mA). R2 should be large enough so it doesn't interfere with the voltage limiting effect of D5 under normal circumstances.

All these circuits used a PMOS device to achieve reverse-battery protection. If you want to use an NMOS device to take advantage of its lower r_{dson}, you have to put it on the low-side of the bridge:

What you have to be careful about though, if you employ any of these techniques is that you have to protect the **whole** circuit, not just parts of it. It's not enough to protect the power FETs. The driver circuitry, and the large electrolytic input filter capacitors are also sensitive to reverse battery connection. Don't make the mistake that the [MonsterMoto Shield](#) guys did. Their protection circuit (Q1 and Q2) protects the bridges and the drivers (U1 and U2), but not the input capacitors (C2, C3, C5, C6). Those can easily explode if you connect the battery in reverse!

Limit-switches

When your motor drives something that has only limited travel (say an elevator) you really want to make sure that your load isn't driven over the end.

The easy part of the assurance is to include switches at each end of the allowed range that signal whenever your load is touching them.
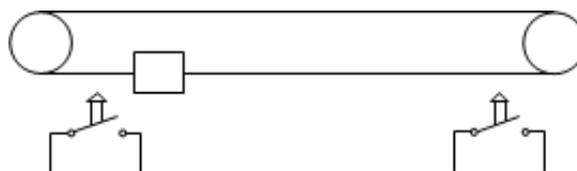
The hard(er) part is to make sure that your motor in fact stops when the switches signal. The trouble lies in that the action you take – while in theory the same – in practice depends on the drive-mode of the bridge.

Want to make sure of is that whenever the forward limit switch is on, your only allow the motor to spin in the reverse direction. Similarly, the reverse limit switch – when on – should limit operation to the forward direction.

In the sign-magnitude drive modes you have a separate signal telling you the intended direction. So what the limit switches should do is to make sure that the PWM input is forced to 0% duty cycle if the direction signal is pointing in the wrong way.

For lock anti-phase drive mode, the direction is encoded in the PWM signal as well. So the action should be the following:

- The forward limit switch should limit the PWM signal to never go beyond 50%, when on
- The reverse limit switch should limit the PWM signal to never go below 50%, when on

This simple action might not be enough though: as the load travels towards the limit switch and actuates it, all that happens is that the motor abruptly changes from forward (let's say) to stationary operation. This will **in some uncontrolled way** slow down and eventually stop the load. During this transient you might have all kinds of problems, like too much travel or too much torque and there isn't much you can do about them. It would be better for example if you could drive the motor backwards to further reduce the travel overshoot or gradually decrease the forward drive to zero to limit the torque. In order to do these you will need to essentially tie in the limit switches to the control loop of the motor.

However doing that isn't without dangers either. As you make the reaction to a danger signal more and more complex, the more likely it is that something in your reaction will go wrong and bad things happen. You might want to include several levels of protection for example: one set of limit switches that trigger a soft-stop operation through the control circuit and another fail-safe set that shuts down the bridge through the previously described control-signal override mechanism.

Summary

In this post we've went through a lot of potentially dangerous situations for an H-Bridge. We've talked about how to detect them and how to react to them when they occur.

Some of the introduced techniques are contradictory (for example sometimes I suggest you included a fuse, in other cases I've adviced against it). When you build a bridge, you always have to examine the use-cases and determine which of the potential problems are real and which ones are not likely to happen.

In the next installment of the series I'll look into closed loop control techniques to implement constant torque or constant velocity controls.