An experiment on motor driver

Science, Physics

Data Analysis

Below in Figure (1) the data from the LVM data can be seen. The corresponding columns are time, measured RPM, and then wanted RPM.

Below in Figure (2) is a graphical representation of the data shown in Figure (1). This is the result of doing various actions including stopping the motor with our bare hands.

Overall this data may look like a mess however when looking at the individual segments it might be a little easier to understand. First starting off with Figure (3) as seen below showing the first curve.

While at first this curve may seem completely random, with context it may be a little easier to understand. Before this point we had the motor running set to a different set “ wanted RPM” that was slightly higher than the 2000 RPM used in this set. This graph shows the transition between the different RPM’s to have an inverse curve. This could be be a possible result of the voltage instantly changing. This would result in the motor having to change its RPM, however since it can’t change instantly it has to gradually settle towards the desired RPM the closer it gets to what is wanted.

The next area shown in Figure (4), shows the flat part of the graph.

Here we decided to let the motor run for a while to see what happened. The result was the RPM aligning itself very closely to the specified RPM.

In this area we decided to manually increase the RPM to different times. This explains why the range seems to jump in a start looking formation. Although as you can see to the very left, it does look like an exponential increase. This signifies what the inverse curve did before, being that there couldn’t be a dramatic change in RPM; it was gradual. Towards the end we return back to the 2000 RPM and once again you can see that inverse curve.

Here what was supposed to be captured was us stopping the motor manually and then releasing it, however we did not capture the data. Though from the various attempts what we saw was a sudden spike in the RPM that at its max point went higher than the desired RPM. This shows that the RPM while trying to its original state overcompensated the RPM and readjusted itself afterwards.

Conclusions

Explanation

Starting with the input DAQ Assist, it should first be stated that there are three channels being measured: The proximity, the motor voltage, and the resistor voltage. The first two channels have a range of positive and negative 10 volts, while the resistor voltage has a range of positive and negative 10 millivolts. These conditions were set in place because these are the expected ranges that these measured values should fall within. The number of samples for each is 100 and the frequency is 1000 hertz. This 1 to 10 ratio will allow a smooth collection of readable data.

Continuing on from the DAQ Assistant a wire emerges that leads into a splitter that separates into three different paths correlating to the order of the channels listed previously. The first path is the path that deals with all of the calculations involving the rpm and loop gain. So from the top part of the splitter a wire first leads a graph module to display the data and also into a " Tone Measurement" module to convert the proximity data into a numerical frequency. This Frequency is then led into a divider to be divided by 8, then into a multiplier to be multiplied by 60. The resultant after these two modules is the measured RPM of the motor at a specific point in time; of course these value is displayed on the front panel through the use of a numerical indicator named " Measured RPM". So after the data is multiplied by 60, the wire then leads into the part of the VI that adjust the data so that the motor performs better. This section first involve the wire leading into a subtraction module to be subtracted from the Target RPM into a merger; note that the target RPM also leads into the merger. Then the resultant data is then leads into an addition module to have 412. 5 added on, then into a multiplier to be multiplied by the " Loop Gain" which is received through playing with the data. From there the wire is led into an addition module to be added to let's say " Wire X" which be explained later. The resultant of this is then led into a shift register in the edge of the while loop encompassing the whole VI.

The other shift register that correlates to the one mentioned previously has a constant of 9 attached to it. This will ensure that the motor will start right away which will negate us having to wait for the motor to start up. The wire coming from this then leads into an " In Range and Coerce" module which looks at if the data is within a certain range of data. If it is not the signal will result in a null, and if it is within the range it will allow the data to proceed. The range in this instance is between 5 and 10, which is the speculated range of the applied voltage we should have in this experiment. The wire coming from this module splits off into two paths. The first path leads to a addition module to be added to let's say " Wire Y" which will be explained later. The second path is the " Wire X" mentioned early. This path basically updates the calculation for the voltage adjustment mentioned before.

Now let's talk about the second path of the splitter mention at the beginning. This path deals with displaying the input voltage of the motor. The data is taken and fed into a graph so that we may see it on the front panel.

The third path from the splitter deals with overload prevention. If the motor gets to hot or has an excessive amount of current this part of the VI will shut off the loop so that the motor is not damaged. Starting off the wire from the splitter first leads into division module to be divided by the resistance of the resistor used in the circuit. This takes the voltage and converts it into a current; this is an example of Ohm's Law. From there the wire leads into a " Greater Than" module to be compared to the max measured current to see if it is or not. If it is not, then that data will proceed to an " Or" module which is also attached to a shift register that is combined with " true" on one end, and " false" on the other end. This setup makes it so the first run will always run, while later will only run if the current signal from the " Greater Than" module didn't return as null. From there the wire leads into an " And" module which makes sure that both signals have to be present in order for the signals to proceed. The other signal in this instance is attached to a " Run Button" and the repeat module for the loop so that we can manually run the system. If both are returned as true, then the " Wire Y" mentioned before is not made. The combination of this wire and the one coming from the range module in the multiplier will ether make the output voltage 0 or the adjusted voltage. The set-up of the output DAQ Assistant is to output 1 channel ranging between -10 and 10 volts set to 1 sample.

After the merger mentioned before the wire then leads into an RPM chart to display the data and also into a “ Write to Measurement File” to display the data on an LVM file. The settings of the “ Write to File Measurement” should be placed as follows. “ Save to one file” should be checked to ensure that all data is placed on a single document. “ Append to file” should be marked so that that the results of multiple files will be added to the document and not replaced. “ Text (LVM” should be hit be cause it is the most reliable format out of all of the options. And the options “ One header only”, “ One column only”, and “ tabulator” should all be pressed due to various formatting strengths such as making it easier to read the data.

Analyze any trends noticed and comment on the best control action for various motor applications.

The main trend that i saw in this experiment was that the motor when transitioning between the different RPM’s had to gradually do so. Taking this into consideration I would say that when doing motor controls, it would be best leave room between RPM transitions before acquiring data for that specific RPM. Also during the experiment one our jobs was to make sure that the motor didn’t overhead, so we implemented a part of the VI to counteract this. If the motors to overheat, the motor itself could be damaged. So implementing a system to do this in all motor experiments is a must.

Conclusion:

Clearly document, and thoroughly explain the operation of your VI

To start the VI, one DAQ Assist was placed. This DAQ was set up with two different channels, one channel for Voltage In and another for the Voltage of the motor. For both of the channels, they were set up to have a minimum of -10 volts and a maximum of 10 volts. This was because we were putting 10 volts through the motor itself. This was also set to read 100 samples at 1000 Hz. This was optimal for recording the data for graphing. Then there was a third channel for the current of the system. This channel was setup to have a minimum of -10 mVolts and a maximum of 10 mVolts, while also reading 100 samples at 1000 Hz. This was also the optimal values for graphing and recording the data.

From there the DAQ was split up into the three different channels. From the first channel, the signal was taken to the graph for the proximity pick up. This would display the frequency of the motor. From there, a tone measurement function would take the signal and turn it into a frequency and using a convert to a dynamic function, would turn it into a value for equations. This value would then be divided by 8 with a division function and then multiplied with a multiplication function by 60 to get the RPMs. This would be displayed with an indicator for the rpms. This data would then be taken to the write to file function for recording along with being displayed on a chart function to display the difference between the target rpm and the actual rpm values. Also in this branch a target rpm control was used to allow the setting of the target rpm we would want. From there the target rpm would be taken to the write to file to be recorded as well as checked with the chart graph to be compared with the actual rpm value. This target rpm was also subtracted by the actual rpm with a subtraction function and then divided by 412. 5 rpm with a division function to turn that value into a voltage. This was then added to the Loop gain of the system with an addition function and then multiplied by the In range and coerce function and then applied to the shift register. The second channel of the system was just taken to a motor voltage chart graph that would display the frequency of the motor voltage.

The third channel was the current that was divided by the resistance to calculate the voltage and then this current would be compared with a greater than or equal to function with a constant of 0. 11 mV, which is the current of the base motor, and if it was greater, this would be our lock up protection system for the motor, so then it would not overheat or wear out.

A second DAQ assist was then placed and was set up as an output for the system with a Voltage Out. This was setup to have a maximum of 10 volts and a minimum of 0 volts, while acquiring 1 sample on demand. This will only generate 1 sample for each run. This part of the VI starts with a shift register with a constant of 9 that is then taken into a In Range and Coerced function. This function with a upper limit of 10 and a lower limit of 5 will check the shift register to see if it is within that limit. The coerced portion will check the value if it is in range and then whether it is in the range of the limits or not, it will output a value within that range and then that range is added later on in the VI with the target rpm value and loop gain on the other shift register. This data is then used with a multiplication function along with the DAQ Assist output and taken to a boolean to “ 0, 1” function. This will tell the value to be either true or false. The previous current of the system that was checked is check to see if the motor is locked up or not with an “ or gate” function, then this signal is taken to the “ and gate” function with the value from the boolean function whether it is true or false. This is the lock up protection system and if it is true, this means that the motor has locked up and this will stop the program. Otherwise, the program continues as normal. All of the VI is then placed within a Loop to allow the VI to run many times during a single experiment.

Finally a write to file measurement was created, along with a boolean button to tell whether the data should be collected or not. The write to file uses the file format of text. LVM, because that is the easiest format to transfer data into word and excel without corruption. The action is to save to only one file and to not append or overwrite the data since there would be multiple runs that need to be distinguishable. The segment header uses one header per segment because we will be acquiring the same list of signals at the same constant acquisition rate. We then use one column per channel because we get different signal types and need values for the x axis for every value of the y axis for graphing. Finally we use Tabular delimiter to help distinguish the different data points and number in a word document.

Analyze any trends noticed and comment on the best control action for various motor applications.

One trend could be that when the motor was being loaded with our hands, the graphs would fluctuate in frequency quite rapidly as you can see in figure

2 towards the end of the graph. The best action for acquiring data would be to allow the motor time to ramp up to the target rpm and then you can find the data you want. Another application is our implementation of the lock up prevention system. This will allow the motor to not overheat and damage the motor. This will stop the motor and hopefully prevent any damage that could have been caused to the system.