ENMT201 - Elevator Project
Lab Assessment 2

1 Position Measurement & Calibration

1.1 Position Measurement

The first step to be able to creating a control system for positioning the elevator carriage is to be able to measure the current location of the carriage. You do this using the optical encoder attached to the winch motor, the same way you used the optical encoder during the PLC labs earlier in the year. You will need to configure the PLC settings appropriately (if you use the provided `Template.cxp` file then this is already done for you), and then use an instruction to read the memory location containing the present value of the high speed counter which reads the encoder.

- Configure the PLC correctly, then use appropriate instruction/s to read off the encoder value. By moving the carriage up and down, show that the encoder value changes to represent the location of the carriage.

1.2 Calibration

What you've got now is a 'number' which corresponds to the position of the carriage. But how do you know which value will correspond to being postioned at each floor? Because the optical encoder is a 'relative' rather than 'absolute' encoder, each time your software runs, the absolute values returned by the encoder will be different. You need to use some method of calibration, so that the location of each floor is obtained. Perhaps you will move the carriage to some known location, then zero the counter; or perhaps you will manually move the carriage to each floor and record the encoder value there. The model code does this automatically, when it moves up and down when first started.

- First, decide on the calibration method you will use. You will need to consider whether or not you will zero the encoder count and how you will record the position of each floor. You will also need to decide what your 'reference point/s' will be, and perhaps which limit switch 'edges' you will use. You will need to justify your decisions to the TAs.

- Then, you can implement your calibration system. You will need to use instructions such as `MOV` or `MOVL` to move data around within the PLC's memory. At this point, you need to consider the numerical aspects of your data; how much space do your variables take up, and are they signed or unsigned? You will need to demonstrate your calibration method to the TAs.

2 Control Automation

Now we're getting to the crux of the assignment! By the end of the lab, you should be able to demonstrate to the TAs a working control system which offers suitable response characteristics. First, you need to select a controller design (see the appendix in the introductory handout for some options). The following instructions assume you've selected a basic closed-loop controller.

2.1 Error Signal & Controller Correction

The first thing to do is to compare where you are with where you want to be, and thus calculate the current 'error' in the position of the carriage. This is simply basic subtration. Then, you can use this error signal to generate some 'correction' signal which you will use to control the winch. If you've selected a proportional controller, then this is simply done by multiplication. Finally, you may need to do some final signal conditioning on your correction signal, to make it suitable for output to the DAC which controls the winch motor.

- Set the 'desired' position of the carriage somehow (using the pushbuttons for the moment), then use maths instructions to calculate the controller 'error'. Demonstrate to the TAs that this error value is correct for all carriage positions (note that both positive and negative error values are valid).
- Implement the controller you've selected to generate your correction signal. Sounds simple, right? When it seems to be working, send the output from the controller to the DAC which drives the winch motor; your controller should (hopefully) now move the carriage in the desired manner. Demonstrate your controller to the TAs!
- Make sure that your controller correctly handles different situations. For instance, you may discover that under certain circumstances, numerical overflow becomes a problem. The TAs will check to see if your controller handles such circumstances without disaster.

2.2 Performance Tuning

Initially, your controller will (probably) have somewhat poor performance characteristics. You will need to incrementally adjust your gains (or other constants) to obtain a better response. The model code is available for comparison; it is poorly optimized, being designed to run on all four of the elevator rigs; so you should be able to optimize your software to be as good as the model code (on your chosen elevator) fairly easily. You are aiming for a controller which has a smooth motion with little or no overshoot, whilst maintaining as fast a rise time as possible. The amount of steady state error which is permissible will depend on the range which the floor limit switches operate over. You will need to test your controller on different combinations of floors, since there may be some difference.

- Tune your controller as best you can. The TAs aren't really marking you on how good your controller response is today. So long as it's operating reasonably, then proper performance tuning can wait until the final demonstration.