MODEL WIND TURBINE MOTOR SPEED CONTROLLER SOFTWARE OVERVIEW:

The primary function of this device is to maintain a constant shaft RPM. Consequently during typical operation the most commonly used VI will be the one containing the name “Change RPM gracefully”, the front panel of which is shown below:

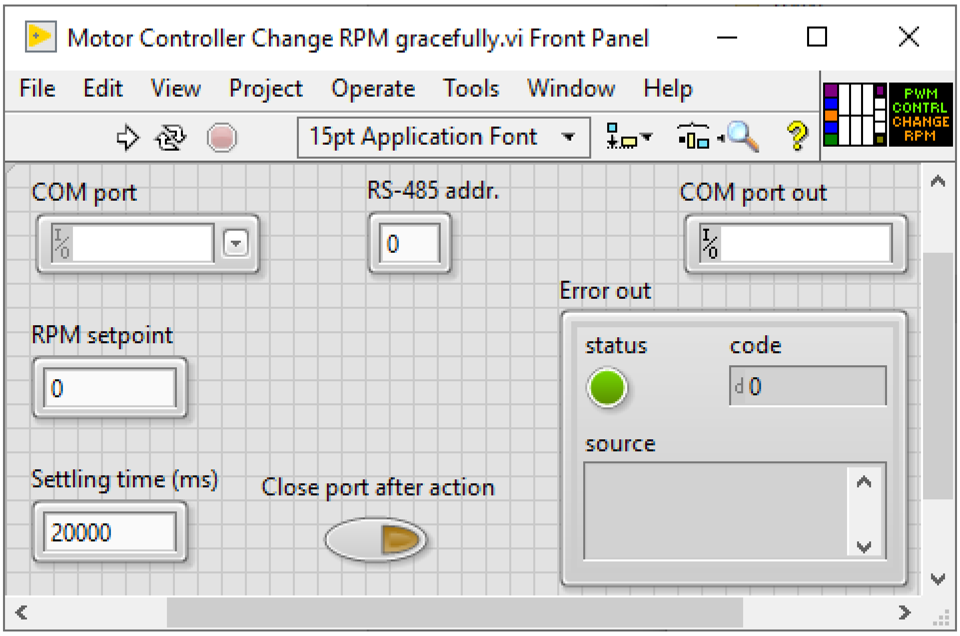

Figure 1: "Change RPM gracefully" VI front panel.

This VI automatically switches to the soft-mode PID coefficients to enact a change in the RPM, then after the specified settling time will switch to the normal-mode PID coefficients for tighter control of the RPM. The operating steps for this VI are similar to all the other VIs:

1. Select COM port. If using the USB interface, this is the COM port of the device itself. If using the RS-485 interface, this is the COM port of the USB to RS-485 converter.

2. If using the RS-485 interface, enter the RS-485 address of the device to which the change in RPM is desired. Otherwise simply leave as zero. Note that entering 0 for the RS-485 address when using the RS-485 interface will send the command to ALL connected devices regardless of their individual address. This feature is useful for enacting a global change in RPM for all devices, for example.

3. Set the desired RPM setpoint. This is the shaft RPM (i.e. final output after reduction gearbox).

4. Enter the settling time (in ms) between soft-mode and return to normal-mode PID control.

5. Select “Close port after action” if the COM port is not open by any other VI (typically USB data streaming) and can be closed without issue.

6. Run the VI.

Upon successful execution, the “Error out” status should remain green with no source message and the COM port will be returned in “COM port out”.

When optimising the PID parameters, the VI containing the name “Config PID” is used:

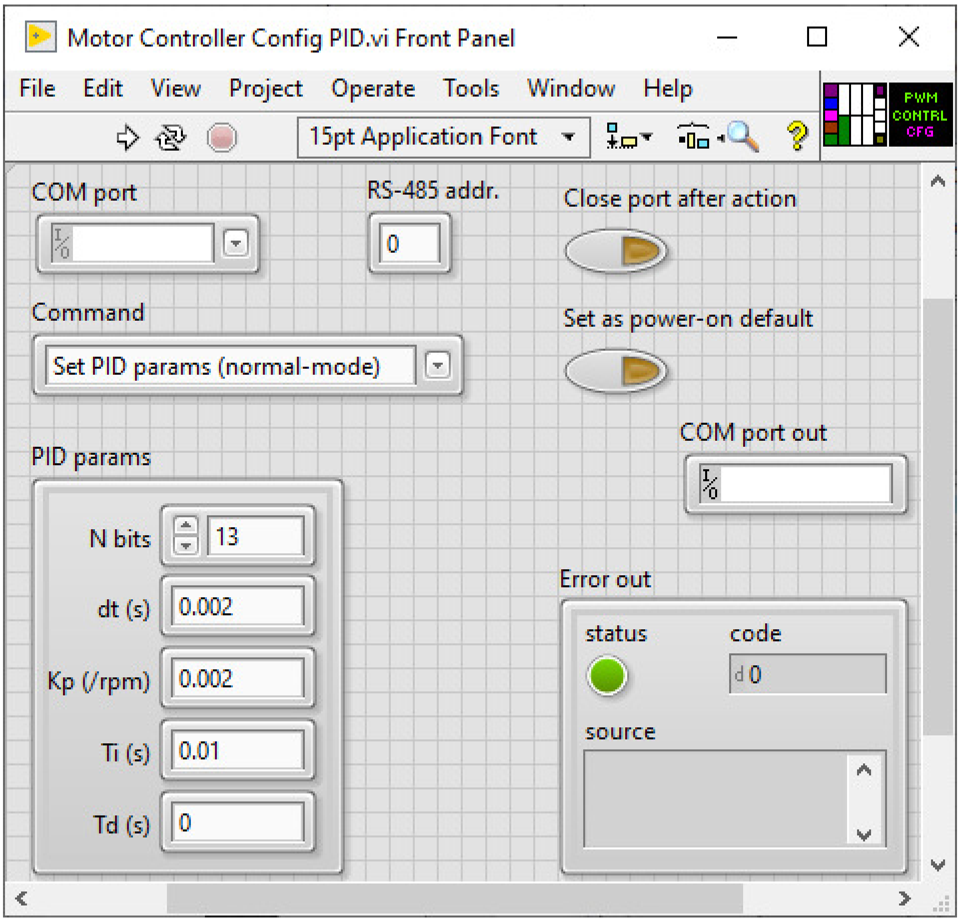

Figure 2: “Config PID” VI front panel.

The PID parameters can be changed accordingly, with the “Command” pulldown menu allowing selection of whether to change the normal-mode or soft-mode PID parameters. If the parameters are desired as the new power-on defaults, then the “Set power-on default” can be enabled. This will write the values to the on-board EEPROM.

It is advisable to use the “Read Live Config or EEPROM” VI to first read back the existing values to ensure a sensible starting point if only wanting to change one of the parameters. This can be done either programmatically or by manual entry. Note that the “Live Config” is that which is currently active, while the “EEPROM” contains the power-on default values.

It is possible to have lower level control of the controller by using the VI containing the name “Drive Command”:

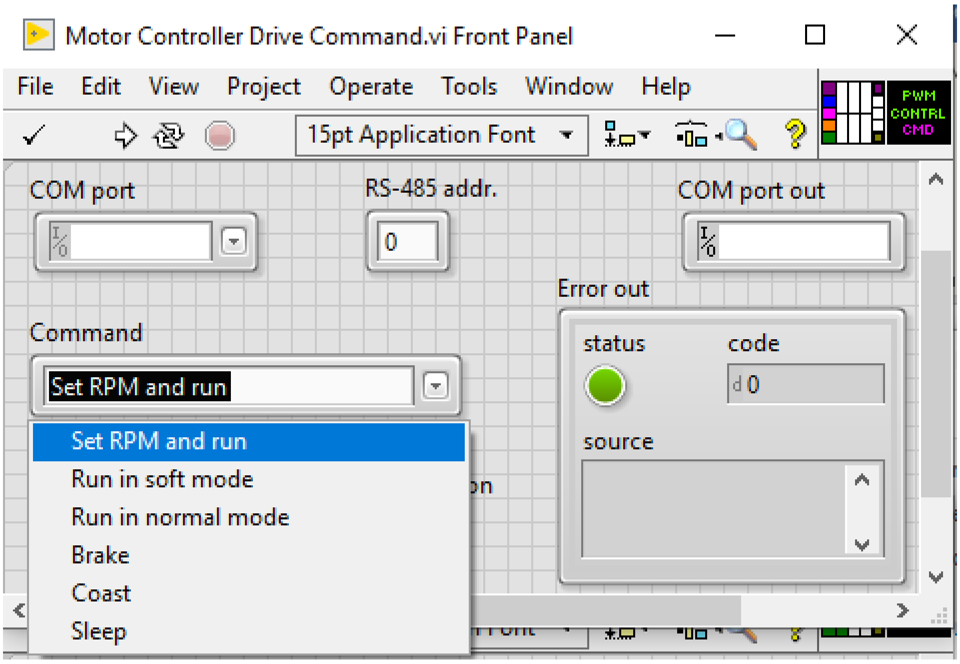

Figure 3: "Drive Command" VI front panel, with Command menu shown extended.

“Set RPM and run” simply sends a new RPM setpoint (entered in the numerical control below the menu – obscured in the above image) without changing to soft-mode first. All the other drive commands ignore the RPM setpoint entry. It is possible to manually change the run mode, as well as put the drive into coast mode (motor terminals effectively floating, high-Z state), brake/slow-decay mode (motor terminals shorted together), and sleep mode (drive completely disabled). Note that the brake mode is only passive and weaker than braking by setting the RPM setpoint to zero. In the latter manner of stopping, it is advisable to enter coast or sleep mode once the RPM has reached zero to minimise power consumption (otherwise the controller will be actively holding the rotor at zero RPM, unless of course this is the desired behaviour).

If using the USB interface, the Vis containing the name “Datastream (display only)” and “Datastream (get N samples)” can be used to monitor the real-time values of RPM, motor current, photoreflector signal, and PWM duty cycle. The latter VI also features a data logging capability that returns an array of clusters containing the monitored values acquired at the desired data rate (which itself can be manually set using the VI containing the name “Set Data Rate”, or using the pull-down menu in the VI itself). If using the RS-485 interface, data streaming is not possible but must be done in a repeated one-shot manner. This can be done using the VI containing the name “One-shot data”. Note that this VI also outputs data packet frame and checksum status Booleans to confirm data transmission integrity. The data packets employ CRC-16 CCITT checksums on both the command and data segments of the packets.

If using the RS-485 interface, it is good practice to periodically check and clear the error state of the system. Any error flags remain set until cleared. This can be done using the VI containing the name “Get and clear error state byte”.

The mechanical configuration of the motor + encoder + gearbox setup can be set either using the VI containing the name “Mech Config” or by direct entry into the appropriately named numerical controls in the VI containing the name “Write EEPROM”. These are namely the number of encoder edges used (1,2 or 4), the encoder resolution (PPR), and the final drive ratio (X:1).

Finally, an example of carrying out PID parameter optimisation can be found in the VI containing the name “two-parameter optimisation”. This VI simply loops through an array of Kp and Ti values over the specified range and resolution (Td is typically not used for motor speed control) and records the variance of the RPM (the optimisation objective to minimise in this example). This is then plotted graphically, the minimum of which indicates the optimal values of Kp and Ti for minimum RPM variance. More complex combinations of objectives can be chosen such as minimum RPM variance and power consumption together.

Paul Nathan 03-MAY-2021