# Rapid Controller Prototyping

LabVIEW Fan Module Exercises

## Overview

The aim of these exercises is to develop LabVIEW programs to characterise the steady-state and dynamic performance of the helicopter fans. The data generated in these exercises will be used to develop the dynamic models of the helicopter fans, used in the design of the helicopter controller.

To characterise the performance of the helicopter fans, a fan module has been developed to measure the thrust force developed by the same type of fan that is used on the helicopter. The fan module comprises of a fan mounted to a load cell, and an instrumentation amplifier, as shown in Figure 1.

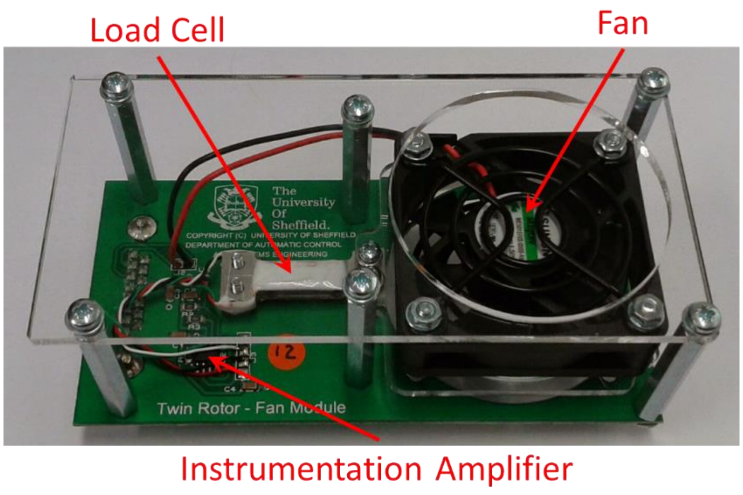


**Figure 1. Photographs of the Fan Module**

The load cell is a force measurement sensor, which uses resistive strain gauges to measure the distortion in the load cell structure, resulting from an externally applied load. An instrumentation amplifier is used to amplify the change in output voltage from the load cell, to be read by the helicopter's myDAQ data acquisition module.

In the first exercise, you will perform a calibration procedure on the fan module, to correlate the voltage measurement of the load cell to a known external force. This will be achieved by incrementally adding known weights to the top of the fan, (which will produce a force acting on the load cell in the same direction is the fan thrust force), and measuring the voltage output from the load cell measurement. The result of this exercise will be a data file that will be used as a reference calibration for the load cell in the other two exercises, i.e. this data file will facilitate the calculation of the fan thrust force from a measured load cell voltage.

During the second exercise you will develop a LabVIEW program to measure the steady-state fan thrust from a static control value. You will then use this program to generate a table of data that correlates the steady-state fan thrust for a range of fan control signal values.

In the final exercise, you will develop a LabVIEW program to measure the step response of the fan module, and write the response results to a data file.

The data gathered from these exercises will be used in exercise 3 of the Modelling, Simulation Control and Laboratory performed in MATLAB and SIMULINK, to generate a time-domain simulation model of the fans.

Accompanying this lab sheet is a LabVIEW project, **LabVIEW Fan Module Exercises [Student].lvproj**, which contains 4 programs, two of which are completed Vis:

- **Main - Fan Module Test Program.vi** – This is a simple test program for the fan module
- **Main - Load Cell Calibration [Student].vi**: - This VI will be used to calibrate the load cell output against a known loading force

These completed VI will provide you with some example code for operating the fan module, and saving the recoded data. You will need to use the descriptions provided, and these programs to develop the following two blank Vis:

- **Main - Fan Characterisation - Steady State - [Student].vi:** This program will be used to record the steady-state response, of the fan module, to a static control value.
- **Main - Fan Characterisation - Dynamic - [Student].vi:** This program will be used to generate the step response of the fan module, and save the data for analysis in MATLAB.

During the development of the LabVIEW programs discussed in these exercises, you will be expected explore the function and controls palettes to find the required functions and controls, and use the LabVIEW help facilities to learn how to correctly implement these blocks. Furthermore, you will be expected to use the information from the previous LabVIEW worksheets, and the some of the example code, provided in the first two VI listed above to help with the development of your programs, described in exercises 2 and 3.

**After you have completed these exercises described in this document, please bring your work to the attention of the laboratory demonstrators, so your work can be assessed.**

## Exercise 1: Calibration of the Load Cell Module

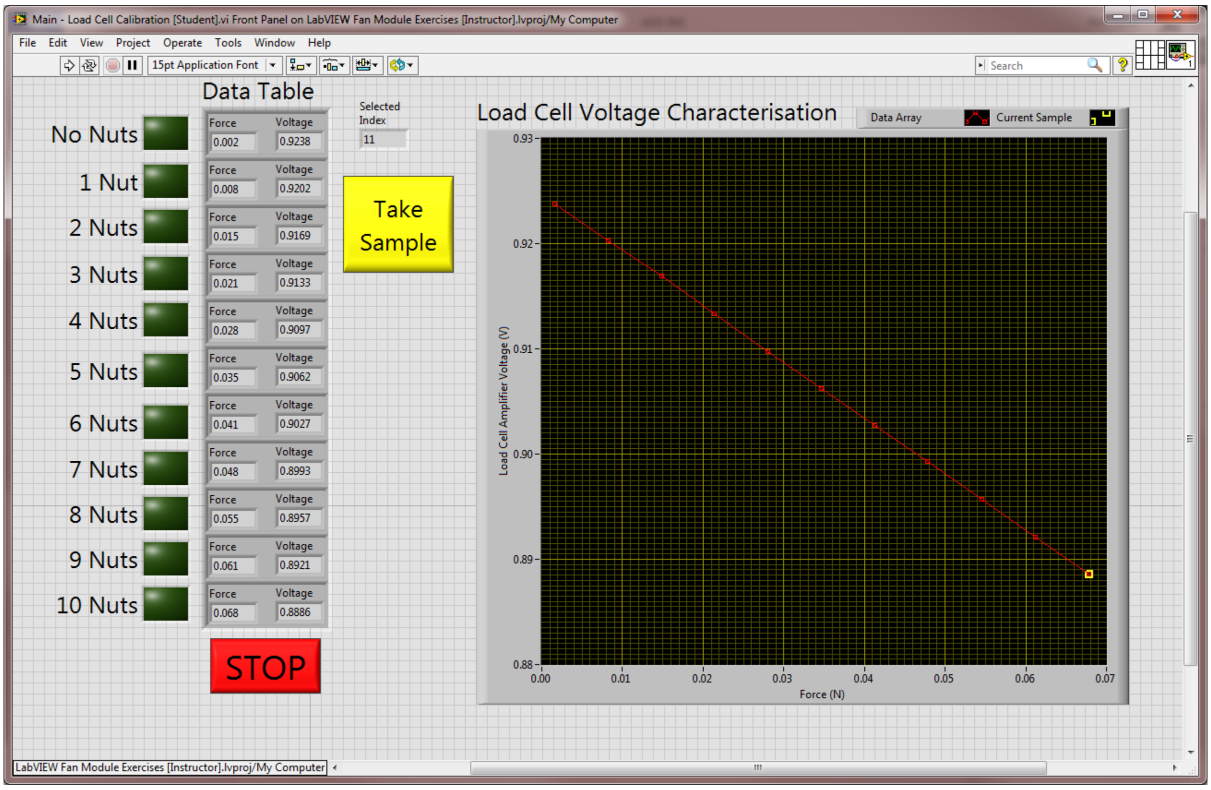
During this exercise you will calibrate the load cell module, and generate a data file: ‘Load Cell.txt’ that will be used in the subsequent exercises to provide relationship between the load cell output voltage and the force generated by the fan.

### Description

The thrust of the fan module acts in an upward direction, thus causing a downward reaction force on the fan actuator. In this exercise you will add know weights to the top of the fan, to apply a force in the same direction as the fan thrust, and measure the resulting voltage from the amplified load cell measurement. The output from this exercise is a data file: ‘Load Cell.txt’ that contains the force to output voltage calibration data for this particular fan module unit.

Before starting this exercise you will need to down load the fan module .zip file, and place the contents of the .zip file into your working folder. The .zip file is located on the MOLE web site: **MOLE Site»RCP»LabVIEW»ACS336-ACS6110-ACS6336 - LabVIEW Fan Module Exercises [Student].zip**

Navigate into the **Fan Module** folder and open the project file: **ACS336-ACS6110-ACS6336 - LabVIEW Fan Module Exercises [Student].lvproj**. Inside this project you will see 4 VI files; for this exercise you will need to open **Main - Load Cell Calibration [Student].vi**. The front panel for this VI is shown in Figure 2.


Main - Load Cell Calibration [Student].vi Front Panel on LabVIEW Fan Module Exercises [Instructor].lvproj/My Computer

Data Table

| | Force | Voltage |
|---|---|---|
| No Nuts | 0.002 | 0.9238 |
| 1 Nut | 0.008 | 0.9202 |
| 2 Nuts | 0.015 | 0.9169 |
| 3 Nuts | 0.021 | 0.9133 |
| 4 Nuts | 0.028 | 0.9097 |
| 5 Nuts | 0.035 | 0.9062 |
| 6 Nuts | 0.041 | 0.9027 |
| 7 Nuts | 0.048 | 0.8993 |
| 8 Nuts | 0.055 | 0.8957 |
| 9 Nuts | 0.061 | 0.8921 |
| 10 Nuts | 0.068 | 0.8886 |

Selected Index: 11

LabVIEW Fan Module Exercises [Instructor].lvproj/My Computer

**Figure 2. Front panel view for the Load Cell Calibration LabVIEW program**

During the calibration process, you will add known weights, (in this case M4 nuts, provided in a small bag to accompany the fan module), to the top of the fan and measure the resulting voltage for each unit of weight you add. For each step of the calibration process, the data will be plotted in the graph, shown to the right of Figure 2.

During this exercise, a small piece of paper, approximately 50mm by 50mm, is required to stop the nuts falling through the fan guard during the loading procedure. A simple line template for this is provides at the back of this document, but any piece of paper this approximate size can be used, as long is flat and does not touch the support pillars or the plastic cover.

## Procedure

1. Download this .ZIP file, and place the contents into your LabVIEW working folder.
2. Launch LabVIEW
3. Open the ACS336-ACS6110-ACS6336 - LabVIEW Fan Module Exercises [Student].lvproj file, located in the Fan Module folder.
4. Open the **Main - Load Cell Calibration [Student].vi**, and start the VI.
5. Place the piece of paper on top of the fan guard, between the fan and the plastic top cover. (You must ensure the paper is not touching the support pillars or bending, such that it touches the top plastic cover.)

   Look at the graph on the right of Figure 2. You will see a yellow marker surrounding a red marker around. This is the current measurement point. Allow this point stabilise on the graph before continuing. You should also see that the green indicator, to the right of Figure 2, labelled ‘No Nuts’ is illuminated. This is the current measurement point of the procedure.

6. Once the measurement is stabilised, press the big yellow ‘Take Sample’ button.

   This will store the measurement for the force of the paper acting on the fan module. You will notice now, that there is a second point on the graph, the value you have just recorded marked in red to the left, and a new active measurement point to the right. You should also notice that the ‘1 Nut’ indicator is illuminated.

7. Add a single nut to the paper, and wait for the active measurement point on the graph to stabilise.
8. Once the measurement is stabilised, press the big yellow ‘Take Sample’ button.

   This will store the measurement on the graph, and illuminate the next measurement marker. As you proceed you will notice that the data for the applied force and the measurement voltage is also shown in the data table, to the left of Figure 2, for each measurement point you take.

9. Repeat steps 7 and 8, until you have added 10 nuts to the paper.
10. When the press the ‘Take Sample’ button for the ’10 Nuts’ sample, the VI will write the ‘Load Cell.txt’ data file and the program will terminate.

You should now proceed to the next exercise. If you change fan module unit at any point, the data that you recorded for the previous fan module unit will not be valid for the new unit, so you will need to rerun this calibration exercise each time you change of fan module unit.

## Exercise 2 – Steady-State Characterisation of the Fan Thrust

During this exercise you will write a LabVIEW VI, **Main - Fan Characterisation - Steady State - [Student].vi** contained within the fan module project, to calculate the steady-state thrust output from the fan module, for a range of excitation voltages. The load cell calibration data, recorded in exercise 1, will be used to translate the load cell output voltage into a fan thrust measurement.

This exercise is split into a number of parts, some of which will be carried over into the next exercise, to form the basis the **Main - Fan Characterisation - Dynamic - [Student].vi**.

- **Part1 - Initial VI Configuration:** you will use the framework provided in Figure 3 to configure the data acquisition elements of the system.
- **Part 2 – Read the Load Cell Calibration file:** in this part you will read the load cell calibration file, display the data on an X-Y graph and use linear interpolation to calculate the slope and y-axis intersect of the calibration data.
- **Part 3 – Control and measure Fan data:** during this you will import the load cell characterisation data into your VI and use it to convert the measurement voltage into a fan thrust measurement
- **Part 4 – Averaging of Fan data:** in this part of the exercise you will apply point-by-point averaging to the measurement to reduce the noise contribution in the load cell measurement.
- **Part 5 – Offset Cancelation of Measured Data:** in this part of the exercise, you will add an offset cancelation function into your VI to eliminate the offset voltage in the measurement.
- **Part 6 - Recording of Steady-State characteristics:** in this part you will use the VI you have developed to record the steady-state characteristics of the fan thrust.

## Part 1: Initial VI Configuration

### Description

The starting point for this exercise is to build the block diagram, shown in Figure 3. This will provide you with the basic structure for the data acquisition elements of your program. **NOTE**: the initial structure for this program is similar to the open loop helicopter exercise, but for these exercises you will be using the following data acquisition blocks from the SubVI folder within the **ACS336-ACS6336-ACS6110 FanModule.zip** file:

- Create Fan Tasks.vi
- Fan Weight Read-Write.vi
- Close Fan Tasks.vi

### Procedure:

1. Open the **Main - Fan Characterisation - Steady State - [Student].vi** file from the Fan Module Project. This should be a blank VI
2. Using the **Select a VI…** option from the functions palette on the block diagram, import the above three VIs from the SubVI folder within the Fan Module directory, and place them on the block diagram.

3. Build the VI as shown in Figure 3.
4. Save your work and move on to the next part

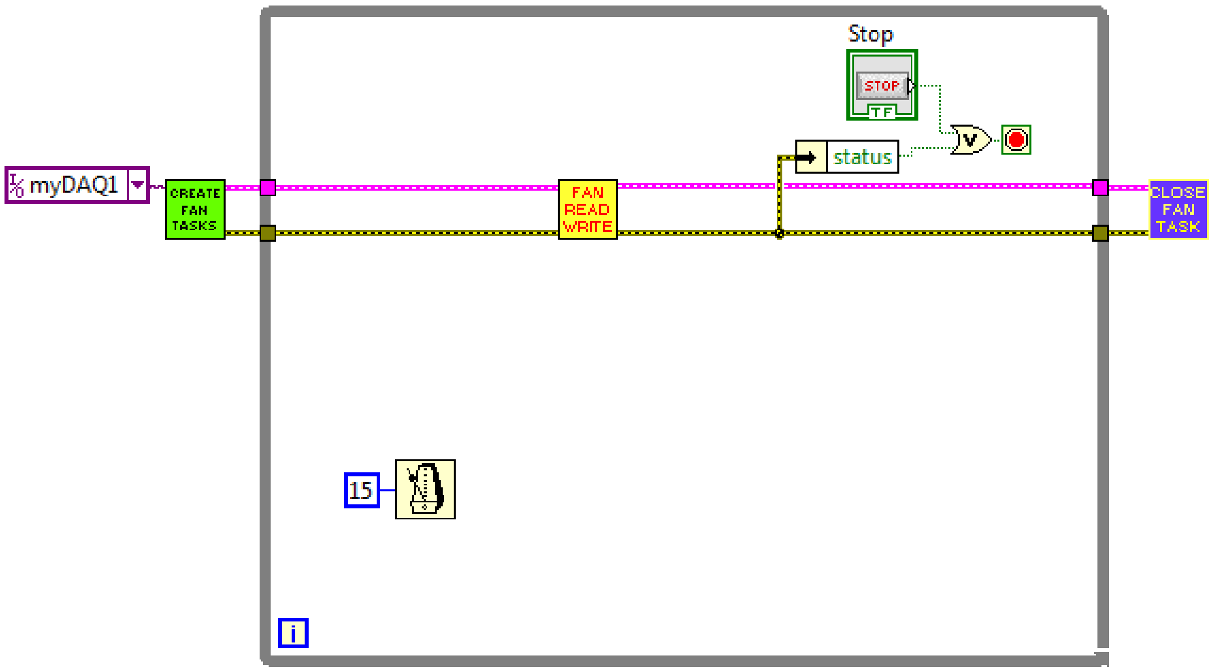


**Figure 3. Initial acquisition framework for the fan module VIs**

## Part 2: Read the Load Cell Calibration File

During this part of the exercise you will write code to import the data from the ‘Load Cell.txt’ file, generated in exercise 1, and display the results on an X-Y Graph, to verify the data has been read correctly.

It can be seen, from observing the characteristic data from exercise 1, that the force to voltage characteristic of the load cell is linear, therefore, the data follows the trend of Y=mX +c, (where: m is the slope and c is the y-axis intersect). To use this load cell data with in your program, you will first need to extract the slope and intersect value from the calibration data, using a linear fit operation.

### Description

The ‘Load Cell.txt’ data file is a simple text file with two rows of data, the first row is the force in Newtons, and the second row is the output voltage from the instrumentation amplifier.

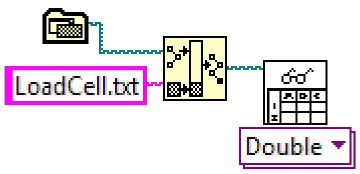


**Figure 4. Example method for loading the ‘Load Cell.txt’ data file**

To load the data into LabVIEW, a **Read Delimited Spreadsheet** block can be used. This function requires a path variable input to provide an address reference for the data file. The path variable can be generated using a **Build Path** and the **Application Directory Constant** blocks from the functions palette, as shown in Figure 4.

The output data from the **Read Delimited Spreadsheet** block is a 2-D array, and you will need to split this into individual 1-D arrays, using an **Index Array** block.

The simplest method of platting the X-Y graph of the calibration data, is to use the **Express XY Graph** which can be found on the controls palette from the front panel.

(**Note:** You can use the search facility in the top right of the controls and functions palette to find the elements discussed in this document)

The slope and intersect point of the calibration data can be calculated using the **Linear Fit VI** found in the functions palette. (*Note*, there are a number of optional inputs to this block which are not required for this exercise: weight, tolerance, method and parameter bounds.)

**Procedure:**

1. Using the method discussed above, add the following functionality to your VI:
    a. Read the 'Load Cell.txt' data file
    b. Plot the X-Y characteristic
    c. Calculate the slope and intersect of the data, using a linear fit
    d. Display the slope and intersect point on front panel indicators
2. Add a constant to the Fan Input to the **Fan Read Write** block. This will allow your code to be tested without flagging an unattached input error for this block.
3. Run your VI and check your results:
    a. Ensure that you have formatted the X-Y graph with appropriate axes legends, and the data is as expected, from the results you obtained in experiment 1.
    b. Ensure also that the calculated slope and intersect points of the characterisation data are also as expected.
4. Save your work and move on to the next part

## Part 3: Control and Measure Fan data

### Description

During this part of the exercise, you will add a front panel control to provide a control value for the fan and a waveform chart to display the time history of the measured signal. You will also be required to convert the measured voltage from the load cell into a force measurement, using the slope and intersect variables generated in the previous part.

The slope of the load cell characteristic data is the Volts-per-Newton gain of the load cell, and the intersect value is the offset Voltage of the load cell measurement, (at the time the calibration test was carried out). You will need to use these values to convert the measured load cell voltage into a force, (fan thrust), measurement.

**Procedure:**

1. Replace the numeric constant, which is wired to the Fan Input of the **Fan Read Write** block, with a front panel numerical control. You will need to constrain the value of this control between the values of 0 and 10. To achieve this:

   a. Wire a numeric control the Fan Input terminal
   b. Switch to the front panel
   c. Right click on the control and select **Properties**
   d. From the **Data Entry** tab of the properties box, you can specify the maximum, minimum and increment values for the numeric control.
2. Develop an appropriate algorithm to convert the L/C Output signal from the **Fan Read Write** block into a Newtons force measurement. This measurement is the thrust force of the fan.
3. Add a waveform chart onto the front panel of your VI to display the time series of your calculated fan thrust force.
4. Run your VI.
5. Save your work.

When you run the VI, you will notice two attributes in the data displayed on the waveform chart:
1. The measurement signal contains a high level of noise
2. There is a small offset force on your waveform measurement, but the signal gain is not affected. (This may not be immediately obvious but does vary over time as temperature and other factors affect the load cell)

During part 4 of this exercise, an averaging filter will be applied to the measured data to reduce the noise power on the measurement signal. In part 5 of this exercise, you will implement a method to address the signal offset.

***At this stage of development, you may wish to save a copy of your VI into a temporary location, because parts 1 to 3 of this exercise will form the first part of exercise 3.***

## Part 4: Averaging of Fan data

### Description

The noise observed in the load cell measurement has a symmetrical distribution, spatially, with the distribution centred of the measurement value. This noise characteristic makes it possible to use data averaging on the measured data to reduce the signal noise.

During this part of the exercise you will apply a point-by-point averaging filter to the measurement data, to significantly reduce the noise power on the measured signal.

The point-by-point averaging filter is implemented using the point-by-point mean block from the functions palette: **Signal Processing»Point By Point»Probability and Statistics PtByPt»Mean**. The degree of noise reduction is a function of the number of samples in which the average value is calculated. Unfortunately, the buffer length also affects the bandwidth of the averaged signal, (i.e. large averaging buffer length leads to lower noise, but reduced signal bandwidth and lag in the output response). As a result, this noise reduction method is suitable for measurement of the steady-state, DC, response of the fan module, but will be detrimental to the dynamic response characterisation.

To observe the effects of the implementation of the averaging filter, the output of the averaging filter will be compared to the unfiltered force measurement, using a waveform chart.

When observing the force measurement of the fan, you will notice that there is a settling time between changing the control signal level and the output reaching its steady state level. For the filtered data, this settling time is partly due to the dynamic response of the fan thrust production, and partly due to the filter implementation. To ensure that you are measuring the data in steady-state, it is useful to know the length of time that the filter will have influence over the signal. As part of this section of the exercise, you will calculate the buffer length, in seconds, of the averaging filter and display this information on the front panel.

**Procedure:**

1. Implement the averaging filter on the measured data. Experiment with the buffer length to observe the effects of this implementation, (a buffer length of approximately 100 should be sufficient)
2. Use a **Bundle** block, from the **Programming»Clusters, Class & Variants** menu on the functions pallet, to combine the filtered and unfiltered signals into a cluster, before plotting this cluster on a waveform chart. (Due to the plotting order of the data, the filtered data should be the top input to the **Bundle** block, with the unfiltered data below. This will plot the filtered data on top of the unfiltered data, and allow the filtered data graph to be seen.)
3. Use the **Wait Until Next ms Multiple** Function input value and the point-by-point **Mean** buffer length value to calculate the buffer length of the averaging filter, in seconds.
4. Save your work and move onto the next part.

## Part5: Offset Cancelation of Measured Data

### Description

During this part of the exercise, you will implement a strategy to remove the offset voltage from the load cell measurement, (both from the filtered and unfiltered data). One method of achieving this is illustrated in Figure 5, and described below:

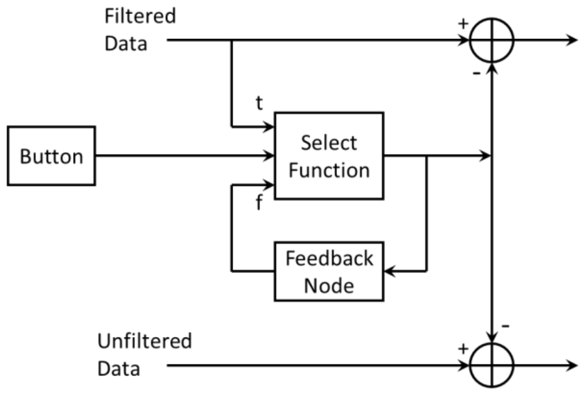


**Figure 5. Diagram of offset cancelation method**

The offset cancelation method illustrated in Figure 5 comprises of:

- **Feedback Node**: used to store a variable value from the previous loop iteration.
- **Select Function**: used to select which variable to store for the next iteration
- A front panel button: used to provide an event to trigger the **Select Function**.

The offset compensation is achieved by subtracting the value stored in the **Feedback Node**, (which is routed from the **Feedback Node** through the **Select Function**), from the measured and filtered data. The **Feedback Node** stores the output of the **Select Function** for the next iteration.

If the button is pressed, the **Select Function** routes the filtered data signal to its output, and this value is stored for the next iteration. If this is done when there is no extra force acting on the fan, either due to an external mass or from the fan thrust itself, then the value stored in the **Feedback Node** will be the averaged offset value of the load cell measurement.

**Procedure:**

1. Implement the offset cancelation method, as described above
2. To ensure correct operation of this method, you must ensure that the value of the button returns to zero after the button is released. To achieve this, switch to the **Front Panel**, right click on the button and select **Mechanical Action** from the menu. This will bring up a new menu, shown in Figure 6, from which you should select the top right selection: **Switch Until Released**.

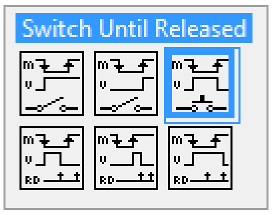


**Figure 6. Mechanical Action Menu for the switch operation**

3. Remove the input connections to the cluster Bundle block, and connect the outputs of the offset cancelation function you have just created.
4. Save your work.

When you run your VI you should now be able to remove any offset in the measurement when the fan is at rest.

As you operate the system, you will notice that the offset value drifts slightly, which can be mitigated by performing an offset calibration.

This offset is partly due to non-uniform thermal expansion of the thermal expansion of the load cell, and other small temperature related affects. The issues will affect the offset of the device, but not affect the gain, therefore, repeating exercise 1 should not be necessary, unless you change the Fan Module.

The VI that you have developed during the previous parts of this exercise should not perform the following functions:

- Read the ‘Load Cell.txt’ calibration file, generated in exercise 1

- Display on an XY graph the load cell calibration data
- Calculate and display the calibration data slop and y-axis intersect
- Convert the measured load cell voltage into a Newtons force measurement
- Have an averaging filter for the measured data
- Have a zero offset cancelation for the load cell force measurement
- Display the averaging filter buffer length
- Have a numerical control for the fan input signal
- Display the current filtered value for the measured fan force, to be recorded during the steady-state measurement test.

An example of how the front panel could be arranged is shown in Figure 7.

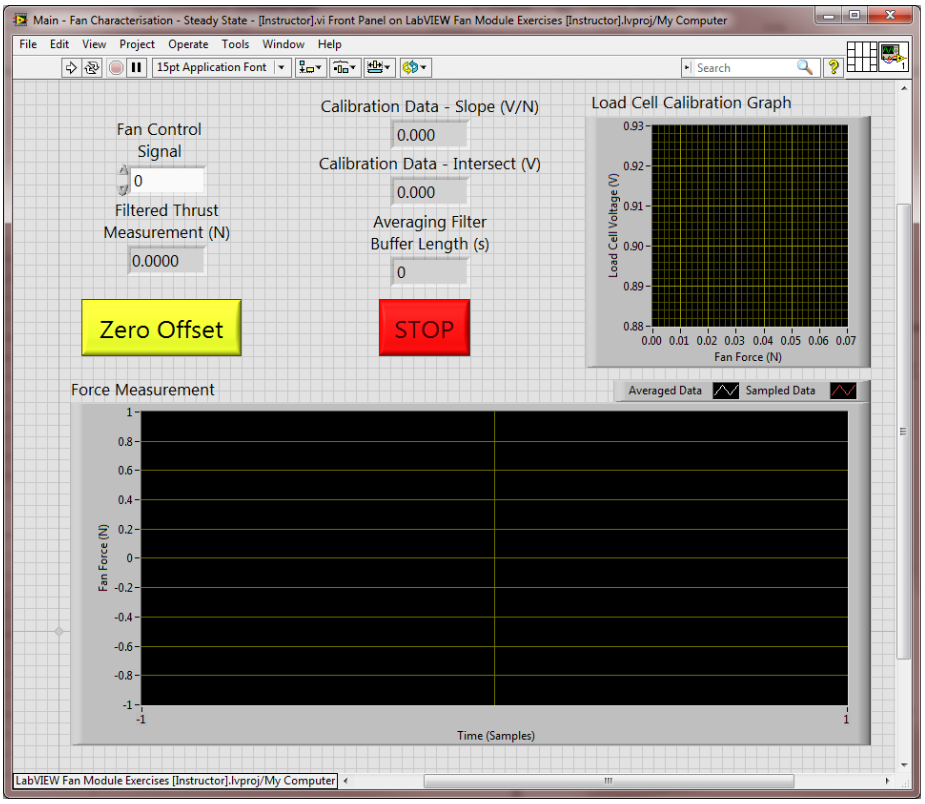

Main - Fan Characterisation - Steady State - [Instructor].vi Front Panel on LabVIEW Fan Module Exercises [Instructor].lvproj/My Computer
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
Fan Control Signal
0
Filtered Thrust Measurement (N)
0.0000
Calibration Data - Slope (V/N)
0.000
Calibration Data - Intersect (V)
0.000
Averaging Filter Buffer Length (s)
0
Zero Offset
STOP
Load Cell Calibration Graph
Load Cell Voltage (V)
Fan Force (N)
Force Measurement
Averaged Data
Sampled Data
Fan Force (N)
Time (Samples)
LabVIEW Fan Module Exercises [Instructor].lvproj/My Computer

**Figure 7. Example of the Front Panel for Exercise 2.**

## Part 6: Recording of Steady-State characteristics

### Description

During this part you will use the VI that you have developed during this exercise to record the steady-state thrust of the fan at different control signal levels. During the test you will increase the Fan Control Signal in steps, allow the filtered response to reach steady-state and record the Filtered Thrust Measurement, for each step. This data will need to be stored for use in exercise 3 of the Modelling, Simulation Control and Laboratory exercises to will be performing in MATLAB and SIMULINK.

**Procedure:**

1. Start your VI
2. Set the Fan Control Signal to zero

3. Allow the system to run until the filtered data is in steady-state.
4. Zero the offset that is on your output.
5. Record the Filtered Thrust Measurement
6. Increase the Fan Control Signal by 0.5.

(You will find that the fan will not start to spin until the fan control signal has reached a value of about 3.0)

7. Allow the filtered data measurement to reach steady state, (this will take a few seconds to stabilise).
8. Repeat points 5 to 7, until you have recorded the data for the control values in the range of 0 to 10 in steps of 0.5.
9. Keep this data for use in exercise 3 of the Modelling, Simulation Control and Laboratory exercises to will be performing in MATLAB and SIMULINK.
10. **Plot your results in MATLAB, ensuring that you have correctly formatted the graph with appropriate labels and title, and show your results to the demonstrators to ensure that you have recorded is correct.**

## Exercise 3: Step Response characterisation of the Fan Thrust

During this exercise you will develop a LabVIEW VI, **Main - Fan Characterisation – Dynamic - [Student].vi** contained within the fan module project, to perform a step-test on the fan module and record the response.

The data generated from this VI will be used in exercise 3 of the MATALB/SIMULINK: Modelling, Simulation and Control laboratory Exercises to calculate the time constant of the fan thrust.

You will be required to incorporate the following elements into your program, for this exercise.

1. Fan Module DAQ Tasks
2. ‘Load Cell.txt’ data file read and conversion of the data into slope and intercept
3. X-Y plot of the characterisation data
4. Conversion of the acquired load cell Voltage into a force measurement in Newtons
5. Real-time calculation of the excitation signal
6. Displaying of the data as the test I in progress
7. Automatically stopping the program after 10 seconds
8. Post processing data to remove initial offset value
9. Displaying the post processed data at the end of the test
10. Writing the post processed data to spreadsheet file: 'FanDynData.txt'

From the above list, elements 1-4 are similar to those described in exercise 2, and an example of element 10, writing to spreadsheet file, is provided in the **Main - Load Cell Calibration [Student].vi**. The functionality of these element will not be further described within this document.

When your VI is complete, you will need to use it to record the step response of the fan module, which will be recorded in the spreadsheet file: 'FanDynData.txt'. This file will be used in Exercise 3 of the MATALB/SIMULINK: Modelling, Simulation and Control laboratory.

## Part 1: Initial Starting point

The initial stage of the development is to, tasks 1 to 4 listed at the start of this exercise are described in parts 1 to 3 of exercise 2. As recommended, you should start with the VI that you saved at the end of part 3 of exercise 2.

The output from this part of the exercise should be a force measurement variable calculated, as a combination of the load cell measurement voltage and the calibration data from experiment 1.

**Procedure:**

Save the temporary file that created at the end of Exercise 2, Part 3, in your project directory and name it **Main - Fan Characterisation – Dynamic - [Student].vi**.

## Part 2: Excitation Control

During the dynamic test, you will stimulate the fan module with a step input signal.
The initial value for the control signal should be zero, and this should step up to a control value of 10, after 1 second of operation. After 10 seconds of operation the while loop should be terminated.

**Procedure:**

1. Use the **Wait Until Next ms Multiple** function input value to calculate the loop iteration time in seconds.
2. Calculate the total run time by multiply the loop iteration time by the loop step number, (the output from the i icon in the while loop)
3. The program run time should be wired to the right-hand edge of the while loop, to create an orange 'Tunnel' similar to that shown in Figure 8.

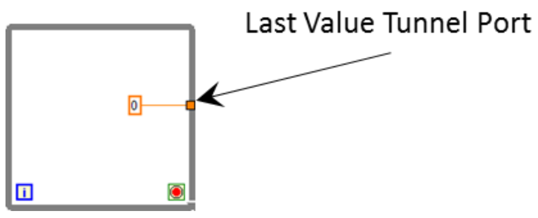


**Figure 8. While loop, showing the tunnel port on the right-hand edge**

4. A Select function can be used to switch the Fan input Signal between zero and 10. You will need to use the Boolean variable from Step 7 to select the true/false input.
5. The excitation signal should be wired from the output of the **Select Function** to the right-hand edge of the while loop, to create an orange 'Tunnel' similar to that shown in Figure 8.

6. The measure Fan Thrust signal should be wired from the output of the **Select Function** to the right-hand edge of the while loop, to create an orange ‘Tunnel’ similar to that shown in Figure 8.
7. From the **Programming»Comparison** menu in the function palette, use a **Greater or Equal?** Function to generate Boolean value to change the Fan Input signal, from zero to 10, after 1 second of operation, and another to trigger the termination of the while loop after 10 seconds.
8. Use a Boolean OR block to connect the Boolean termination value, which you trigger after 10 seconds, to the termination mechanism currently in place.
9. Add a vertical fill slider to the front panel, scaled between 0 and 10, and connect it to the program run time variable. This slider will show the progress of your VI.
10. Save your work
11. Run your VI

Your VI should now:

- Run for 10 seconds and terminate
- Contain a vertical fill slider should thats display value increases, starting at zero at the start of the VI execution and reaches 10 when the VI terminated after 10 seconds
- Drive the fan input signal, such that, for the first second of operation the fan input signal is zero and the fan is stationary. After 1 second, the fan input signal should step up to 10, and the fan start should spin up to its maximum speed, for a further 9 seconds before the VI terminates. (The Close Fan Task will automatically stop the fan).

## Part 3: Plot the Excitation and Measured Force Signals on a Twin Y-Axis Waveform Chart

The excitation signal and the measured thrust response of the fan should be plotted on the same chart, so that the time domain characteristics of the generated thrust can be observed in relation to the occurrence of the step input.

For this system, the magnitude of the excitation signal and the measurement signals are considerably different, (in the order of x100 difference), which will cause problems when attempting to compare the data on the same graph. To overcome this problem, a second y-axis can be added to the chart to compare signals of different orders of magnitude.

**Procedure:**

1. Add a waveform chart to the front panel, to display both the excitation signal and the measured force, using a build cluster block to route these signals into the waveform chart block.
2. Run the VI once to configure the waveform chart indicator for the incoming data
3. Right click on the y-axis and select **Duplicate Scale**, which should add a second y-axis scale to the left side of the chart
4. Right click on the new y-axis scale and select **Swap Sides**, which should transfer this axis to the right hand side of the chart.

5. Label the two axes appropriately for the measurement signal on the left and the excitation signal on the right
6. Expand the plot legend so that the legend has two plots
7. Open the Property Box for the waveform chart, by right clicking on the chart, and open the **Plots** Tab.
8. Select **Plot 0** From the dropdown box, at the top of the tab, and change the **Name** to an appropriate name for the excitation signal
9. From the **Y-Scale** dropdown select the right-hand axis
10. Select **Plot 1** From the dropdown box, at the top of the tab, and change the **Name** to an appropriate name for the measurement signal
11. From the **Y-Scale** dropdown select the left-hand axis
12. Select the **Display Format** tab, in the chart properties box
13. From the dropdown box at the top of the tab select the measurement signal axis.
14. Select **Default Editing Mode** from the radio button at the bottom of the tab
15. Change the information in this tab to:
    a. **Type:** Floating Point
    b. **Precision Type:** Digits of Precision
    c. **Digits: 3**
    d. Uncheck the **Hide Trailing Zeros** selector
16. Clock the Properties Box OK button.
17. Run your VI
18. Save your Work and Move on to the next Part.

As the VI runs, the waveform chart should display the two signals, with both axes auto-scaled to associated signal.

You will notice that there is an offset in the recorded thrust measurement, when the excitation signal is set to zero. This will be removed in post processing.

## Part 4: Convert the Output Tunnels on the While loop Into Indexing Ports

### Description

The three data output ports on the right of the while loop:
1. Program Run Time
2. Excitation Signal
3. Measured Fan Thrust

will by default, only provide the last value processed by the while loop. The output data file from this exercise must contain data for each time instance of these variables, from that start of the while loop execution. To achieve this the **Tunnelling Mode** of this port needs to be changed to **Indexing**.

### Procedure:

1. Right click on the solid orange tunnelling port on the right side of the while loop, similar to that shown in Figure 8, for either of the three signals listed at the start of this part.
2. Select **Tunnelling Mode** from the menu, and select **Indexing** from the sub menu.

This should have changed the port from that shown in Figure 8 to one similar to port shown in Figure 9.

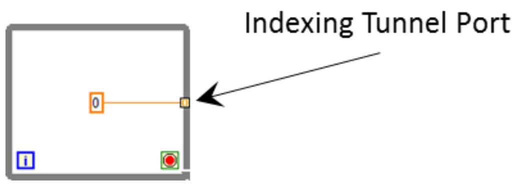


**Figure 9. While Loop with an Indexing Tunnel Port**

The output data from this indexing tunnelling port will be a 1-D array of data for each iteration of the while loop.

3. This process should be repeated for all three signals ports listed above
4. Save your work and move on to the next part.

## Part 5: Post Processing of Dynamic Thrust Measurement to Cancel the Offset in the Force Measurement

It will be noticeable on the run time data, generated from the graph described in part 3 of this exercise, that there is an offset on the measured fan thrust data.

### Description

During this part of the exercise, you will create a post processing algorithm to analyse the first second of the thrust measurement data, when the excitation signal is set to zero. The algorithm will then calculate the mean value of the measurement, over this period, then remove this value from the entire measured data array.

One method to achieve this is to use the average value of the fan thrust measurement, while the fan control signal is equal to zero, as illustrated in Figure 10.

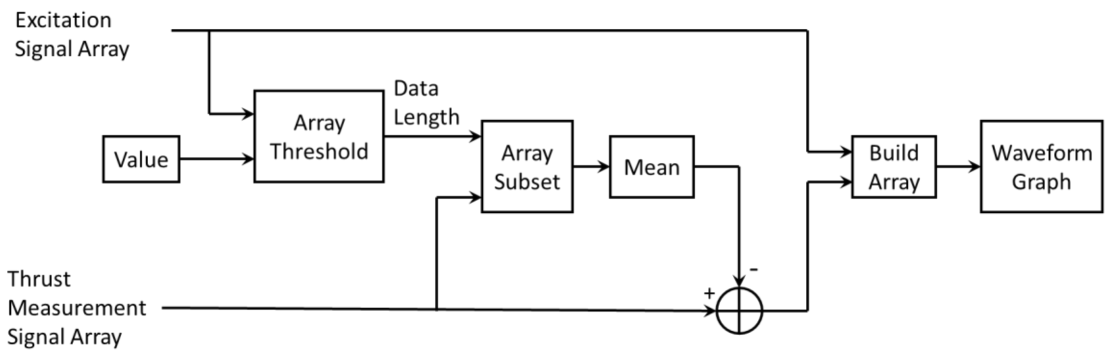


**Figure 10. Diagram of the offset cancelation post-process algorithm**

The algorithm uses a **Threshold 1D Array Function** to analyse the excitation signal array, to count the number of elements that are below a threshold value, (I have used a threshold value of 1.0 in my test code). This will give us the number of elements at the start of the data array that correspond to the excitation signal of Zero.

The value generated by the threshold array function is a double precision real number. This needs to be rounded down towards zero before converting it to a 32-bit integer, so that it can be used with an Array Subset function, to extract the start of the measured data array.

After subtracting the mean of the initial measured data set, the data can be plotted using a waveform graph. (NOTE: this is a waveform graph, not a waveform chart as previously used. See: [http://zone.ni.com/reference/en-XX/help/371361J-01/lvconcepts/types_of_graphs_and_charts/](http://zone.ni.com/reference/en-XX/help/371361J-01/lvconcepts/types_of_graphs_and_charts/) for a discussion on the differences between LabVIEW Charts and Graphs)

Use the method described in Part 3 to add a second y-axis to the graph indicator, (this same method is applicable for both graphs and charts). To ensure the 0 values of the two y-axes line up, allowing the data to be easily viewed, remove the AutoScaling from the two y-axes, and the excitation range between -1 and 11, and the measured force range between -0.005N and 0.055N.

An example of how the front panel could be arranged is shown in Figure 11.

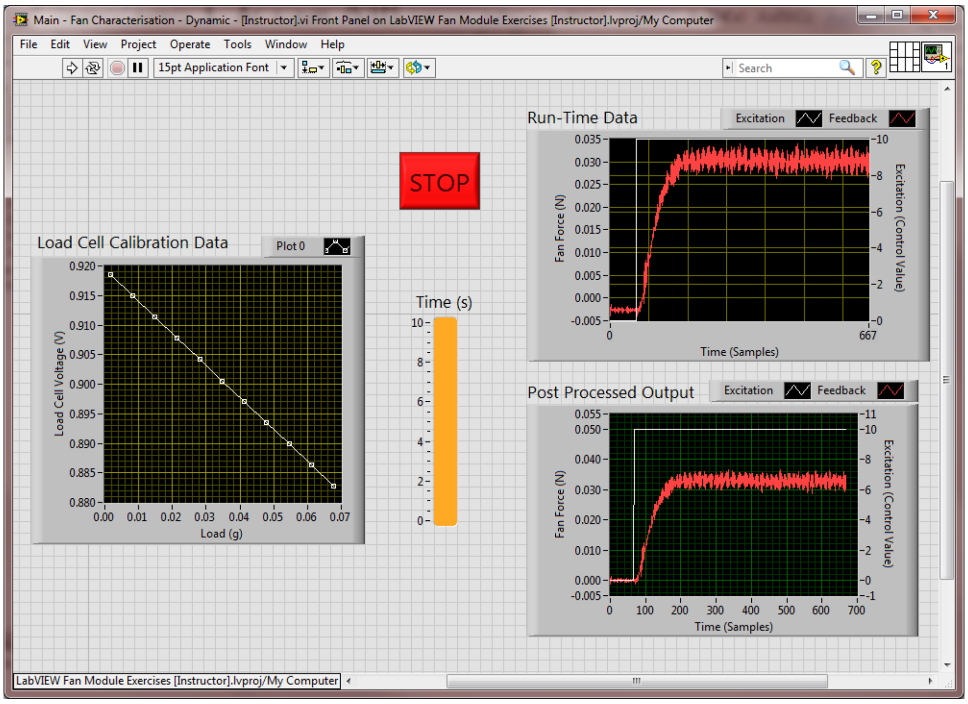

Main - Fan Characterisation - Dynamic - [Instructor].vi Front Panel on LabVIEW Fan Module Exercises [Instructor].lvproj/My Computer
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
STOP
Load Cell Calibration Data
Plot 0
Load Cell Voltage (V)
Load (g)
Time (s)
Run-Time Data
Excitation
Feedback
Fan Force (N)
Excitation (Control Value)
Time (Samples)
Post Processed Output
LabVIEW Fan Module Exercises [Instructor].lvproj/My Computer

**Figure 11. Example of the Front Panel for Exercise 3.**

## Part 6: Recording the Dynamic Force Measurement

### Description

The three array outputs generated after the termination of the while loop, (with the measurement data post processed), need to be written to a spreadsheet data file. This data will them be used in exercise 3 of the MATALB/SIMULINK: Modelling, Simulation and Control laboratory Exercises, which you will use to generate the dynamic, time-domain model of the helicopter fans.

The file **Main - Load Cell Calibration [Student].vi** provides an example of how to write two of 1-D arrays to a spreadsheet data file. You will need to use this method, and modify the build array block to accommodate three 1-D arrays, and write to a spreadsheet file, called 'FanDynData.txt'

The 2-D data array should be ordered in the following manner:

1. Program Run Time
2. Excitation Signal
3. Measured Fan Thrust

This file can be read into MATLAB and plotted using the following section of MATLAB script:

```
close all
clear
clc

FanDynData = load('FanDynData.txt');


TimeArray = FanDynData(1,:);
Excitation = FanDynData(2,:);
Response = FanDynData(3,:);

figure(1)
plot(TimeArray,Excitation,TimeArray,Response)
grid
xlabel('Time (s)')
ylabel('Amplitude')
legend('Control Signal Excitation','Force (g)')
```

**After you have completed these exercises described in this document, please bring your work to the attention of the laboratory demonstrators, so your work can be assessed.**

## Cut out templates for the 50mm by 50mm piece of paper required for Exercise 1: