Rapid Controller Prototyping

Helicopter Open-Loop Control Exercise
[Software Version 4]

Overview

The aim of this exercise is to develop an open loop controller for the 3-DOF helicopter, and to experiment and observe the dynamics of the helicopter system.

After you have completed this exercise, please bring your work to the attention of the laboratory demonstrators, so your progress can be assessed.

Helicopter Open-Loop Control Exercise

Goal: During this exercise you will build a VI to operate the helicopter under open loop control.

You will be asked to build your program from scratch, but will be provided with a number of predefined VI functions:

1. Create Helicopter Tasks.vi
2. Helicopter Read-Write.vi
3. Close Helicopter Tasks.vi
4. Write Pitch Data.vi
5. Import Pitch Calibration Data [Open Loop].vi

These subVI blocks have been configured to ensure the data acquisition tasks are configured correctly. Feel free to investigate their contents, by double clicking on their icon to reveal their front panel. The blocks are located in a .ZIP file on the ACS6110 Mole web site: **ACS6110 MOLE Site»RCP»LabVIEW»ACS6110 - LabVIEW Blocks.Zip**

Also included in this .ZIP file are a number of other VI blocks, which are internal sub functions of these Vis. They must be located in the same directory as those listed above.

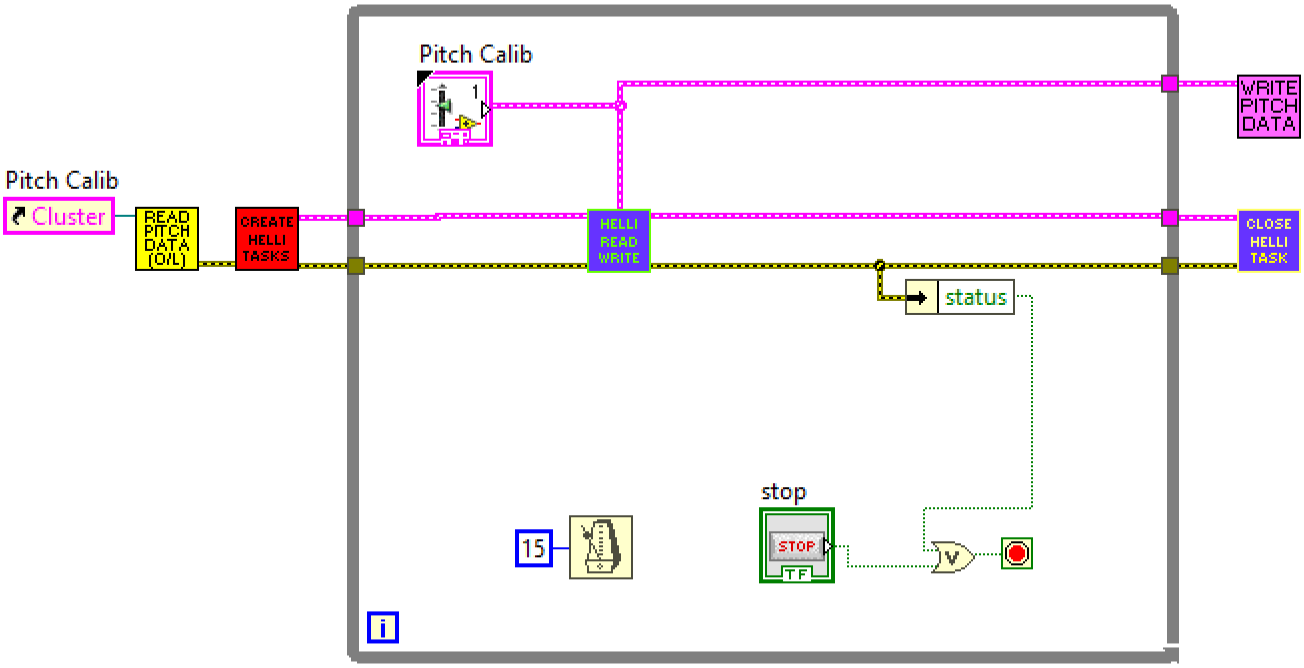
Download this .ZIP file, and place the contents into your LabVIEW working folder, where you have saved your exercise 1-7 files.

To insert an external VI into your code, switch to the Block Diagram screen, open the functions palette and select **Select a VI…** In the file dialogue box, navigate to the location that you placed the external VIs, and select the VI you wish to insert into your program. The external VI block should now have appeared on your block diagram, and can be moved to a convenient location and connected.

Part 1: Initial VI Configuration

Description:

The starting point for this exercise is to build the VI as shown below.



This VI contains a master while loop, with the helicopter DAQmx running through it. The master while loop is terminated either by the stop button on the front panel, or by an error occurring during the data acquisition process. The loop timing is enforced by a Wait Until Next ms block with an input value of 15ms. The Helicopter Read-Write sub-VI has a cluster input, which provides calibration data for the pitch sensor and switches between calibration and operational modes.

- Create Helicopter Tasks sub-VI

This sub-VI creates the DAQmx tasks required to operate the helicopter plug-in: analogue input, analogue out and counter tasks. To reduce the number of wires on the block diagram, the task wires are bundles into a cluster and passed to the output of the sub-VI.

Look inside the sub-VI to see how this has been achieved.

(It should be noted that the Create Helicopter Tasks sub-VI will automatically select the myDAQ module connected to your system, so you do not need to specify the device number)

- Helicopter Read-Write sub-VI

The helicopter read-write sub-VI contains the DAQmx read and write functions for the helicopter system. The input to the block is a 2 element array of double precision numbers, one for each fan, (i.e. element 0 – fan 1, element 1, fan 2)

The numerical outputs from the block are 2 individual double precision values for the travel and elevation angles. The elevation angle is taken from the analogue read task, and scaled to output the correct angle. The travel angle is generated using the counter input task.

Look inside the sub-VI to see how this has been achieved.

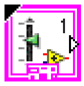
- Close Helicopter Tasks sub-VI

The close helicopter tasks sub-VI ensures that the helicopter tasks are correctly closed and cleared; ensuring the myDAQ module is correctly released when the VI terminates.

Look inside the sub-VI to see how this has been achieved.

- Pitch Calibration Data - Cluster Input

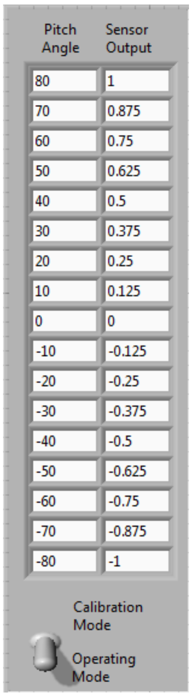
Pitch Calib



The Pitch Calibration Data Cluster is an externally defined data structure, saved in the LabVIEW control file - Pitch Array 1.ctl, and is used to convert the pitch sensor voltage to a angular measurement.

The Pitch Calibration Data Cluster comprises of a 2-D array of data, used to translate the pitch angle sensor voltage into an angular measurement, and a Boolean switch used to select the pitch sensor operating mode.

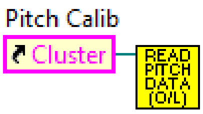
This element can be added to your VI, by right clicking on the Pitch Calib port, at the top of the Helicopter Read-Write sub-VI, and selecting **Create»Control.** The Front Panel view of the cluster control is shown below:



When the switch is in Calibration Mode position, the raw sensor voltage is passed to the pitch angle output of the Helicopter Read-Write sub-VI, allowing the sensor data to be recorded. When the switch is in the Operating mode position, the raw pitch angle voltage is converted into an angle, using the data entered into table.

The data in the table is initially as shown above. For correct pitch angle measurement, the pitch angle sensor voltage must be measured and entered into the table. This data is specific for each helicopter system.

- Import Pitch Calibration Data [Open Loop] Sub-VI



This subVI reads the Pitch Calibration data file - PitchAxisData.txt, and copies the data into the data table in the pitch calibration cluster. To link this VI to the Pitch Calibration Data cluster, you need to link the reference number of the Pitch Calibration cluster to the input of the Pitch Calibration Data [Open Loop] Sub-VI. To achieve this:

1. Ensure the Pitch Calibration Data - Cluster is placed, as described above
2. Place the Import Pitch Calibration Data [Open Loop] Sub-VI
3. On the Block Diagram, right click on the Pitch Calib cluster icon, and select **Create»Reference.**
4. Place the reference icon next to the top left of the Import Pitch Calibration Data [Open Loop] Sub-VI, and wire into the Pitch Data Ref port, as shown in the figure above.

- Write Pitch Data SubVI

This VI writes the Pitch Calibration Data into the Pitch Calibration data file - PitchAxisData.txt. This data can then be used to in the Closed Loop Controller VI, and is also reloaded when the open loop control VI is run. This way you can check the calibration of the helicopter pitch sensor at any time, by comparing the data from the file with measurements taken from the real system, and modify the calibration if necessary – see later sections of this document.

Procedure:

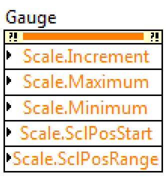
1. Download this .ZIP file, and place the contents into your LabVIEW working folder, where you have saved your exercise 1-7 files.
2. Launch LabVIEW
3. In the getting started window, create a new VI.
4. Save the VI with an appropriate name
5. Use the method described above to import the external Vis:
   - Create Helicopter Tasks.vi
   - Helicopter Read-Write.vi
   - Close Helicopter Tasks.vi
   - Write Pitch Data.vi
   - Import Pitch Calibration Data [Open Loop].vi
6. Build the VI as shown in the wiring diagram the start of the exercise
7. Save your VI

Part 2: Adding the Elevation Indicator and the Travel Angle Gauge Pointer

Description:
During this part of the exercise you will add the output indicators for your VI. The elevation indicator should be a tank indicator, with limits of -50 to +50, and your travel angle should be a gauge pointer.

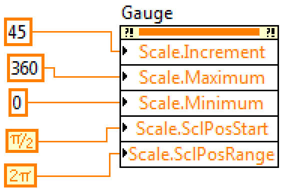
- The Gauge Pointer

The gauge pointer, found on the Controls Palette by selecting **Numeric»Gauge**, will be used to provide a screen indication of the travel angle of the helicopter. The initial configuration of the indicator is for a gauge with a range of 0 to 10, and a rotational angle of approximately 280^{O}. For our purposes we need the gauge to operate through a rotational angle of 360^{O}, ideally with the zero angle ate the centre-top position. To achieve this, we must adjust the property attributes of the gauge indicator, with the use of a property node.



The property node is used to get or set properties and methods on local or remote application instances, VIs, and objects – in this case change the functionality of the gauge indicator. The properties we wish to configure for the gauge indicator are: scale Increment, scale maximum, scale minimum, scale position start, and scale position range. Initially we will create the property node for the scale increment, and expand this for the other properties.

To create the property node, right click on the gauge icon in the Block Diagram and select **Create»Property Node»Scale»Range»Increment**. Right click on the property node and select **Change All To Write**. Select the property node, by clicking on the orange bar at the top, and expand the node to 5 elements. Unselect the property node, and Left click, (with the finger cursor), on 2nd element and select **Create»Property Node»Scale»Range»Maximum**, on the 3rd element and select **Create»Property Node»Scale»Range»Minimum**, on the 4th element and select **Create»Property Node»Scale»Scale Position»Range**, and on the 5th element and select **Create»Property Node»Scale»Scale Position »Start**. Add the inputs, shown in the figure below, to the property node, (the 2π and $\pi/2$ constants is from the **Programming»Numeric»Maths & Scientific Constants** in the functions pallet).

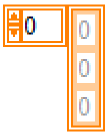


Connect the dial input to the Travel output port of the Helicopter Read-Write sub-VI, and add a digital display to the indicator, as described in the initial exercises. Resize the dial gauge to suit, and move the digital display into a convenient place on the dial gauge.

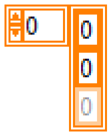
For this part of the exercise, you will need to create an array constant containing two 0 values and connect this to the fan input port of the helicopter read-write block.

The array constant is created using a number of steps: firstly an array constant construct is added to the wiring diagram, **Programming»Array»Array Constant**, then a double precision numeric constant, **Programming»Numeric»DBL Numeric Constant**, is be added into the square box of the construct to cast the array as a double precision array.

The array should now be resized to display the internal elements, as shown:



This is achieved by holding the cursor over the outline of the box, where the double precision numeric constant was dropped into, and dragging the sizing handle at the bottom of the box so that 3 elements appear, as shown above. Double click on the top element and enter the value of 0, repeat with the 2nd element. You will notice that the elements you have entered the value of 0 into now have a solid orange border, as shown below:

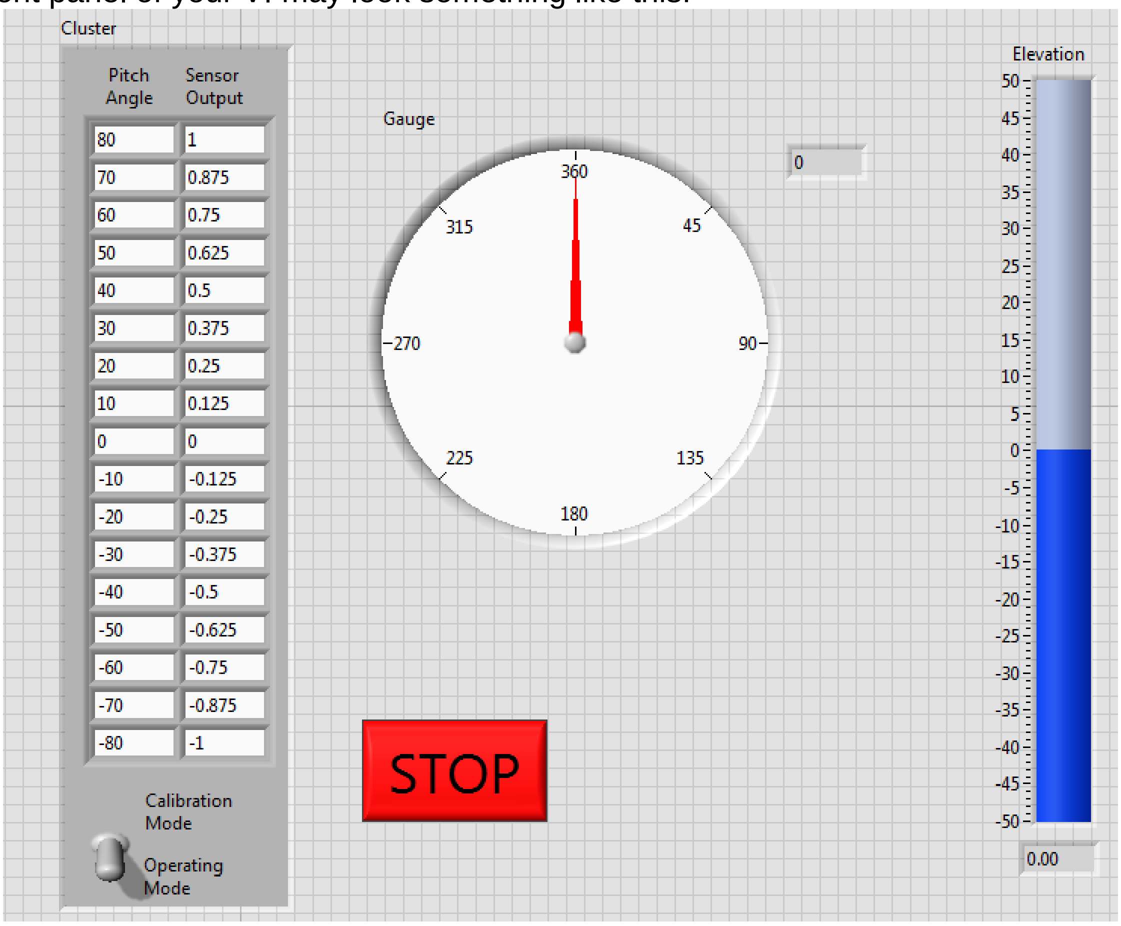


This has now created a 2 element array constant, with both values set to zero

Procedure:

1. Connect the elevation output from the helicopter read-write to a tank indicator
2. Configure the travel angle gauge indicator, as described above.
3. Create the array constant, as described.
4. Connect an array containing two 0 values onto the fan input port of the helicopter read-write block, using the method described above.
5. Resize and re-colour your front panel indicators as you wish.
6. Save your VI

The front panel of your VI may look something like this:

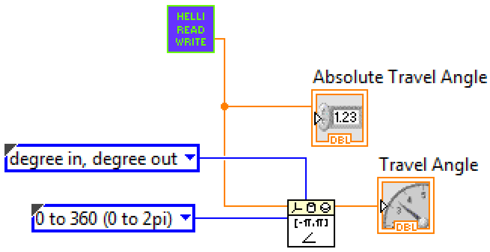


7. Run your VI
8. When you move the elevation angle of the helicopter, the tank indicator should move to approximately +45^{O} at its highest extreme, and approximately -45^{O} at its lowest extreme.

9. When the travel angle is rotated clockwise, from its initial position, the gauge pointer will follow the movement up to 360^{O} but no further. Similarly, if the travel angle is rotated back to anti-clockwise, from its initial position, the gauge pointer will go no further than 0^{O}, in that direction.

   The output from the Helicopter Read-Write sub-VI is the absolute travel angle of the helicopter, i.e. it is not bounded or wrapped between 0^{O} and 360^{O}. The dial gauge will only display data that is between its upper and lower bounds. The absolute travel angle is illustrated in the digital display of the dial gauge.

   To allow the dial gauge to continuously follow the rotation of the travel angle, the travel angle needs to be wrapped between 0^{O} and 360^{O}. This can be achieved using the angle wrap function from the functions palette, **Mathematics»Geometry Angle»Wrap Angle**, as shown below:



   When adding the **Wrap Angle** function, remember create the constants for the ‘Angle Range’ and ‘Angle Units’ inputs, and adjust to the configuration shown above. Also shown above is a numerical indicator added to the output of the Helicopter Read-Write sub-VI to show the unwrapped travel angle, (this should also be added to your VI).

10. After adding these new blocks to your VI, the gauge pointer should continuously follow the motion of the travel angle, and the digital indicator you have added to the gauge should display a numerical value for the wrapped travel angle. The absolute travel angle indicator will provide the absolute rotational angle of the helicopter from its initial position, when the VI was started.

11. Stop your VI by pressing the stop button on the front panel, and save your work.

Part 3: Adding the Pitch angle reading and Calibrating the measurement

Description:
During this part of the exercise you will add the tank indicator for the pitch angle, and calibrate the pitch angle sensor, for your helicopter system. You *must* do this for your individual system because each system is different and, therefore, the calibration values vary from system to system.

The pitch angle is calibrated manually, using the measurement tool cut out from the back of this document. Once cut out, the pitch angle measurement tool is placed over the boom beam and the pitch angle can be measured from the relative position of the fans and the scale on the measurement tool. At each scale interval, the subsequent sensor voltage is entered into the cluster table.

Procedure:

1. Add a tank indicator to the pitch angle output of the Read-Write sub-VI.
2. Reposition, and resize this indicator to suit. It is also recommended that this indicator should be a different colour to the elevation tank indicator.
3. Create a digital display indicator for the tank indicator.
4. Change the range of the tank indicator scale to operate between -80 and +80.
5. Cut out the pitch angle measurement tool from the back of this document. (Further copies can be made using the ‘Pitch Measurement Tool.pdf’ document found on the MOLE site).
6. Start the VI
7. Switch the cluster switch to calibration mode.

   The output voltage of the sensor is now passed directly to the tank indicator.

8. Place the pitch angle measurement tool over the boom bean, with the back of the fans just touching the measurement tool.
9. For each angle listed in the pitch angle column of the cluster table, record the measurement voltage of the pitch angle sensor. (You may not be able to rotate the pitch angle to the extremities, i.e. -80^{O} or $+80^{O}$. Do not worry, just change the angle in the table to approximately the maximum angle achieved and record the measurement voltage.)
10. Plot the measured voltage against sensor voltage in Excel to ensure you have a smooth S-shaped curve. Show this to one of the demonstrators to ensure you have a suitable profile.
11. Switch the switch to ‘Operating Mode’ and verify the pitch angle calibration - recalibrate if necessary.

    As you operate the system, you may notice that the calibration of the pitch sensor changes slightly – particularly at the extreme angles, $>60^{O}$ and $<-60^{O}$. This is generally not a problem because the system will be linearised around the 0^{O} operating point, where the calibration will change significantly. If you are unhappy with this, simply re calibrate the sensor, by repeating points 8 to 13.

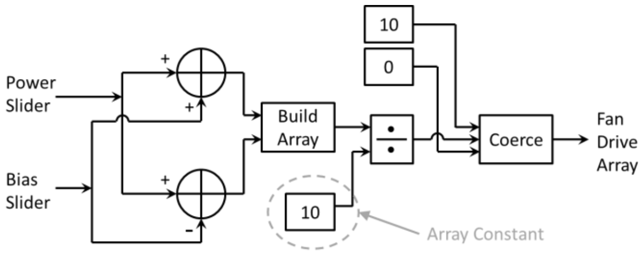
12. To save the calibration data, you will need to make the current VI data default. This is done by clicking on the **Edit** menu in the menu bar and selecting **Make Current Values Default**.
13. Now save your VI.

Part 4: Adding the Fan Control Signals

Description:
During this part of the exercise you will add two sliders to control the helicopter fans:

1. A power slider to control average output to the fans, (0% to 100% of full power)
2. A bias slider to increase the output to one fan and decrease it from the other, (±25% of full scale power)

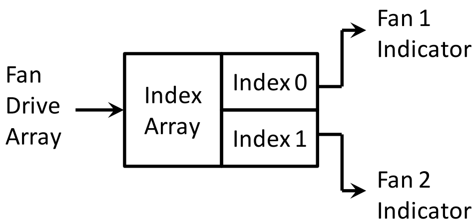
To achieve this control you will develop a simple mathematical function on the power and bias signals to generate the fan input array, as described in the diagram below:



(The Coerce function, in the above diagram, is selected from **Programming»Comparrison»In Range and Coerce** from the functions palette)

(The array constant is created using the method described previously)

It is required that you display the values of the fan inputs on two tank indicators. To achieve this you will need to use an index array block to separate the fan input signals:

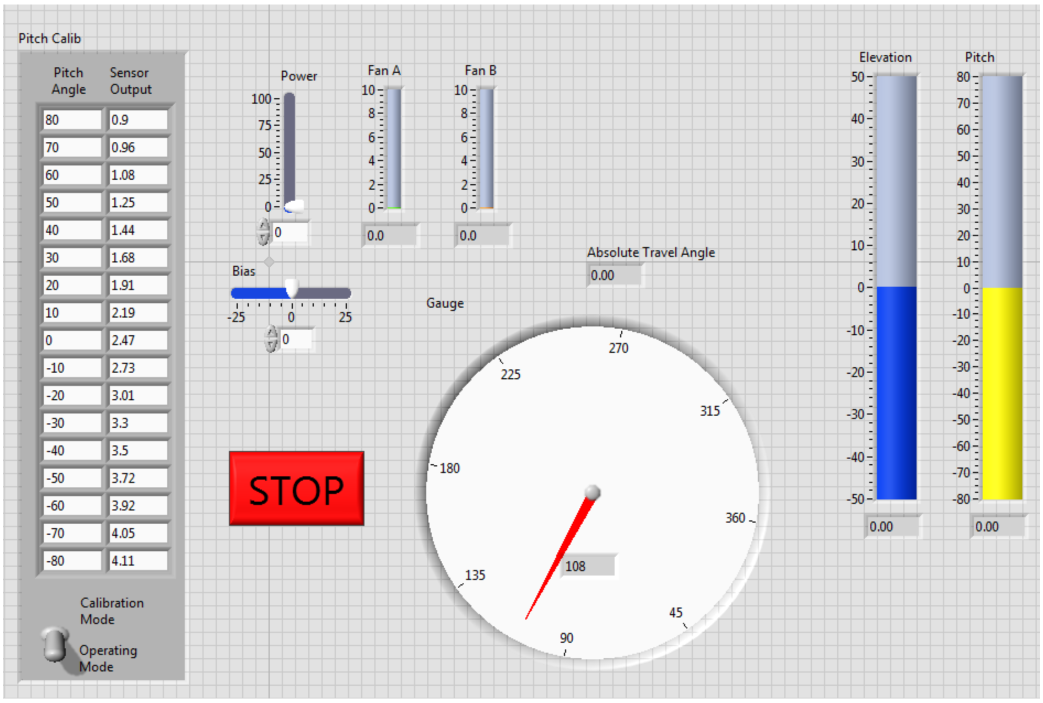


A numerical control will be required for both of the sliders, and similarly, numerical indicators will be required for the tank indicators, (See Experiment 1).

Procedure:

1. Add a vertical pointer slider for the power input, and add the numerical control.
2. The bounds of the power slider should be 0 to 100
3. Add the numerical control for the slider
4. Add a horizontal pointer slider for the bias input, and add the numerical control.
5. The bounds of the bias slider should be -25 to +25
6. Add the numerical control for the slider
7. Add the blocks to achieve the fan input array signal, as described above, and connect them to the power and bias sliders.
8. Add the two indicator tanks for the fan indicators.
9. Set the bounds on the tank indicators to 0 to 10
10. Add the numerical indicators to the tanks
11. Use the index array block to decode the fan input array, (ensure you have enlarged it to provide two outputs), and connect the outputs to the fan indicators
12. Resize and recolour your VI front panel to suit.

    Your VI should look something like the following:



13. Run your VI

The remainder of this exercise is aimed at allowing you to play with the helicopter in open loop, to get a feel of how it perform under open loop control, and grasp how difficult it is to control. The following tests are just suggestions of what you can do and observe.

14. Move the power slider pointer and observe the operation of the fans, such as:

At what value of power slider do the fans start to spin?
At what power slider value does the elevation arm start to lift?
Does pitch angle of the helicopter move as you slowly increase the power, when the elevation angle starts to increase?
If you start from zero bias and zero power, and with the helicopter stationary, what are the effects of putting a step power input of, say, 80% into the power control? Does the same thing happen every time?

15. Try adjusting the fan bias, and see how the helicopter flies.

Can you get the helicopter to hover in one place?
Can you get the helicopter from one place to another?

16. Play with the helicopter, and observe how difficult it is to control manually
17. Make any other observations you fell necessary, and try to get a feel for the dynamics of the system

18. Save your VI.
19. Ask the demonstrator to check your progress

Feel free to make any modifications to the VI code to observe any element of the system or to improve the open loop control of the system.

The Pitch Angle Measurement Tool.
(Scissors are available from the demonstrators to cut-out this measurement tool)