

2nd Year Microprocessors Laboratory

2012-2013

Dr Rachel Edwards, P554

INTRODUCTION

Computer control of experiments is becoming the norm for research and industrial applications, with computers able to analyse data while it is being recorded. This module is intended to give you a grounding in computer control and analysis for experiments, and in using LabVIEW, a graphical programming language, for these tasks.

Preparatory Tasks

Read through the script before the first lab. For the theory section which ***you will need to work on the questions outside of the laboratory sessions***. The online version of the script has **colour** figures. Make sure you have both parts of the script; this one plus the separate Introduction to LabVIEW which is worked through during the first lab session.

Some useful instructions;

USE A LAB BOOK. This should be a record of your work. SAVE your programs to your H: drive and/or a USB stick, and ensure both lab partners have access to the programs (it shouldn't need pointing out that you should keep backups and change the file name between revisions; sometimes you will need to refer to an older version). If your computer is slow to save data (shown by gaps in the data recorded) talk to a demonstrator. The lab outline is given below;

1. **Introduction to LabVIEW** (separate handout); you will learn to write simple LabVIEW routines to do mathematical calculations and to use different loops, and work with a data acquisition board for control of experiments and recording of data. *The separate introduction will take the majority of the first session.*
2. **Main experiment: Theory, programming and further investigation**; using LabVIEW and the data acquisition card you will control a moving arm, calibrating the angle and testing several control mechanisms for setting the arm position. *You must attempt the theory work before you start this section.* You will then choose one further investigation.
3. **Assessment**; *once your final program is completed you must show it working to a demonstrator.*

Approximately one third of the marks will be given to the final program, split between how well it works, program layout (including annotation) and correct use of code. The rest of the marks will be based on your lab book, taking into account results, theory work and discussion. The lab book should be handed in after your laboratory work is complete, and no later than 12 noon on the Thursday following your final session.

Learning outcomes

For successful completion of the laboratory you will be able to

- Demonstrate knowledge of LabVIEW
- Show understanding of theoretical models of control and feedback by comparing these with the behaviour of a real system
- Use feedback to monitor and control a compound pendulum.

IMPORTANT SAFETY INFORMATION

1. The box labelled Servo Controller Interface is the interface between the computer and the experiments. **DO NOT** connect or disconnect the cables between this box and the computer.
2. When using a computer, it is important to take regular breaks to rest the eyes and stretch.
3. Be careful of the nodding donkey – it may bite! And don't electrocute yourself. That would be silly.

BEFORE YOU START THE MAIN EXPERIMENT, YOU MUST COMPLETE THE SEPARATE INTRODUCTION TO LABVIEW. THIS WILL TAKE APPROXIMATELY ONE LAB SESSION.

FEEDBACK AND CONTROL

Feedback and control mechanisms are used in a wide variety of applications, from room temperature control to precision servo controllers on aircraft. Taking as an example the temperature in a room – we don't want the temperature to overshoot what it's set to, or oscillate, but likewise we don't want it to take too long to reach the desired temperature. Feedback of the temperature measured by a sensor can be used to control the room temperature by turning on and off the heaters / air conditioning. In this experiment you will look at stabilisation of a relatively simple system – the position of a moving arm (the “Nodding Donkey”) which is free to rotate and is controlled through moving a weight along the arm.

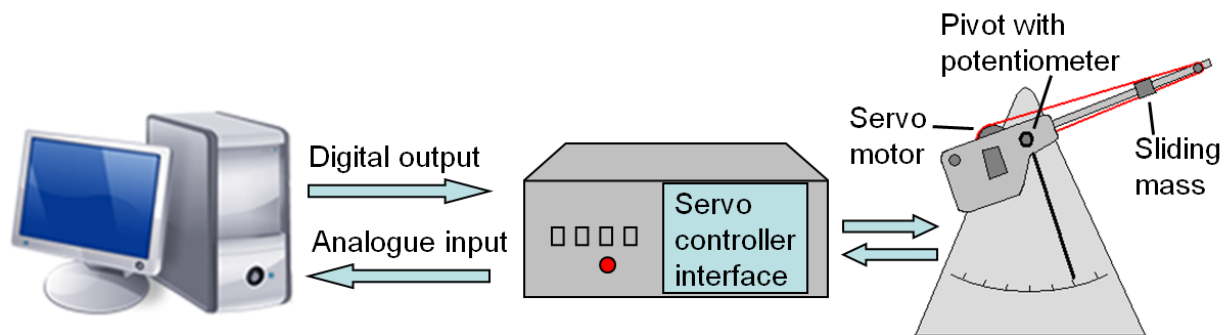


Figure 1; Nodding donkey experiment and controllers

Figure 1 shows a schematic diagram of the equipment used for this experiment. An irregularly shaped object with an arm extending out to one side is pivoted on an axle about a horizontal axis (point P in Figure 2). The angle θ is shown by the pointer and the scale on the stand, or can be measured by calibrating a potentiometer attached to the axle. The voltage of this potentiometer is proportional to the angle of the arm and can be read by the computer through the A-D card. On the arm is a small mass, m , which can be moved by the servo motor and controlled using the digital output of the A-D card; the system should come to rest with the arm approximately horizontal when this mass sits at the centre of its range.

If the system is displaced (for example, by nudging the arm) the centre of mass is no longer

below the axis of support and gravity provides a restoring torque. Moving the sliding mass adds an additional torque and the system will come to rest at a different angle. This mass can be moved fairly quickly, leading to fast control of this additional torque. A requirement may be to bring the system to a series of different positions quickly and with minimal overshoot, or to return quickly to a position despite outside influences, such as nudging of the arm or the addition of extra weights.

1. THEORY

Much of the theory behind this system has been covered in PX145 (Physics Foundations) and PX148 (Classical Mechanics). The system can be modelled as a damped compound pendulum with moment of inertia I . The system is free to move through an angle θ , and with the sliding mass m at x_0 and no outside perturbation will settle at θ_0 . Figure 2 (a) shows the system after it has been moved from equilibrium. A restoring torque,

$$\tau = -Mgr \sin(\theta - \theta_0), \quad (1)$$

where M is the total mass and r is the distance between the pivot (P) and centre of mass (C), will cause it to return it to equilibrium (2(b)). In practice, x_0 is chosen so that θ_0 is zero.

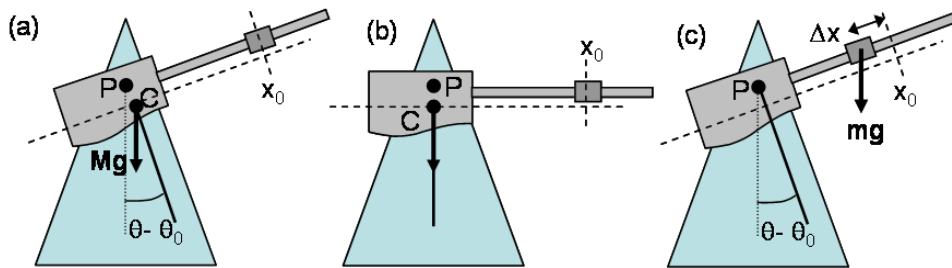


Figure 2; Angles and positions in moving arm experiment

When the weight is displaced a distance $\Delta x(t)$ (fig. 2 (c)) an extra torque occurs,

$$\tau(t) = -mg\Delta x(t) \cos(\theta - \theta_0). \quad (2)$$

$\Delta x(t)$ is defined to be positive for the sliding mass moving away from the axle, and can be varied in order to control the angle of the arm. For small angles the total torque for a system with moment of inertia I can be approximated as,

$$\tau_T = I\alpha = I \frac{d^2\theta}{dt^2} = -Mgr\theta - mg\Delta x(t). \quad (3)$$

Derive (1)-(3) in your lab book. Diagrams may be useful to explain directions and angles.

1.1 Feedback and control

There are several mechanisms for performing feedback and control, each with its own advantages and disadvantages. The control mechanisms you will use are known collectively as PID control (proportional – integral – derivative), and determine the output to the controller by measuring the error $e(t)$ between the system state and the desired set-point;

$$\text{output}(t) = k_P \cdot e(t) + k_I \cdot \int_0^t e(t) dt + k_D \cdot \frac{de(t)}{dt}, \quad (4)$$

where k_P , k_I and k_D are constants for each mechanism.

- **No control**; with no control the arm will act like a damped harmonic oscillator centred at θ_0 when the mass is at x_0 , and the system will slowly return to its equilibrium point after perturbation. Damping is due to dissipative effects, with an equation of motion

$$\ddot{\theta} + 2\mu\dot{\theta} + \omega_0^2(\theta - \theta_0) = 0, \quad (5)$$

where 2μ is the natural damping coefficient and ω_0 is the undamped natural frequency, given by $\omega_0^2 = Mgr/I$.

Derive the solutions to equation (5), considering the general solution AND the different damping regimes.

- **Damping (derivative control)**; this aims to increase the natural damping of the system to bring it to rest as quickly as possible. Equation (5) only applies when the mass is at x_0 , i.e. $\Delta x(t) = 0$. When the mass is moved to damp the motion an extra torque must be considered, and for small angles

$$\ddot{\theta} + 2\mu\dot{\theta} + \omega_0^2(\theta - \theta_0) = -\frac{mg}{I}\Delta x_D. \quad (6)$$

The displacement of the sliding mass Δx_D is proportional to the instantaneous angular velocity of the pendulum $\dot{\theta}$, i.e. $(mg/I)\Delta x = k_D\dot{\theta}$. Equation (6) then becomes

$$\ddot{\theta} + (2\mu + k_D)\dot{\theta} + \omega_0^2(\theta - \theta_0) = 0 \quad (7)$$

which is again damped harmonic motion.

Find, by putting $\ddot{\theta} = \dot{\theta} = 0$, the equilibrium value (i.e. the angle where the donkey is stable). Then derive the **general solution** for this motion.

This allows the motion of the system to be damped, but gives no control over θ_0 .

- **Proportional control**; in a real system we wish to set a target, e.g. an angle for the arm to move to, which requires feedback. By moving the mass a distance Δx_P we can apply an extra torque which is proportional to the error between the desired and actual states, i.e.

$$\frac{mg}{I}\Delta x_P = k_P(\theta - \theta_T) \quad (8)$$

where θ_T is the target angle, changing equation (7) to

$$\ddot{\theta} + (2\mu + k_D)\dot{\theta} + \omega_0^2(\theta - \theta_0) + k_P(\theta - \theta_T) = 0. \quad (9)$$

Find the equilibrium value and the general solution. You will see from your solution that this method of control allows you to approach the target angle but not reach it, unless $\theta_T = \theta_0$.

- **Integral control**; the torque applied in the previous situations will not reliably move the system to the desired state under all conditions. Integral control will sense that the angle at which the system is settling is different from the target angle and adjust the mass by Δx_I until the desired state is reached. This applies a torque τ_i at a time t by setting

$$\frac{mg}{I}\Delta x_I = k_I \cdot \int_0^{t_i} (\theta(t) - \theta_T) dt. \quad (10)$$

If k_I has the right sign this will tend to push the system towards the target angle. This, in

principal, should work on its own, but can have problems when used in isolation.

- **PID control;** To overcome all the problems with each individual control mechanism, PID control is used. The equation describing this is

$$\ddot{\theta} + (2\mu + k_D)\dot{\theta} + \omega_0^2(\theta - \theta_0) + k_P(\theta - \theta_T) + k_I \cdot \int_0^t (\theta(t) - \theta_T)dt = 0. \quad (11)$$

Zeigler & Nichols (see references) developed a method for tuning k_P , k_I and k_D . For a rapid response it is tempting to set them all to be large, leading to a large output. However, this will amplify every error and the system will tend towards chaos. For small values of the constants the system is slow to respond and may not eliminate an error before the next one appears.

2. MAIN EXPERIMENT

Your program to talk to the moving arm experiment needs to read the arm position from the potentiometer analogue input, and control the sliding mass through the digital output. Create a new VI, and on the front panel place an indicator to display the angle (analogue input), and a control for the digital output to the 8-bit motor, *remembering the limits for an 8-bit digital output you calculated in section 2.2 of the Intro to LabVIEW.*

On the block diagram place a DAQ Assistant VI to read the analogue input, As before, the settings are: Analogue input / voltage; Dev1/ai0; 1 Sample (On Demand); Settings: -10 to 10 V, RSE Terminal Configuration, no scaling. Click Run/Stop on the Assistant window to test, and nudge the arm to check that it is reading the right values.

Use a second DAQ Assistant VI to send an output to the sliding mass motor, with the following settings: Digital output; Port output; Dev1/port0; 1 Sample (On Demand). Click Run/Stop to test – the digital outputs that are active are 0 to 7.

Create a front panel with the required inputs and outputs, and wire up your system (a suggested program is shown in figure 3); remember that the DAQ Assistant VI for the digital output expects an array input, so you will need to use a *Build Array*. In the following experiments the time delay you choose will affect your data acquisition.

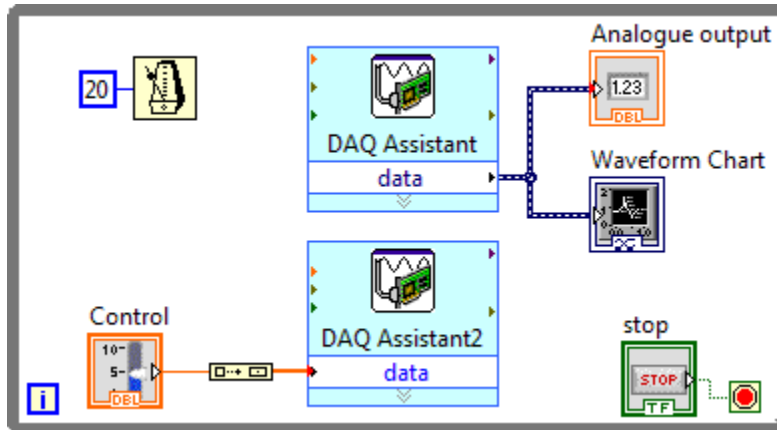


Figure 3; Block diagram for the initial analogue input / digital output routine for moving arm

Guidance

At the end of this section you should have one program capable of using all three PID control mechanisms, and we expect you to correctly implement and discuss each control mechanism. Some guidance for how to do your experiment is given in the steps below. The graphs you produce for your lab book should have labelled and enumerated axes, and for high marks you

should connect theory and experiment. Note that the equations given are for an ideal system, so may not precisely describe this experiment, and you may need to consider the relationship between angle, position and the format the motor expects (8-bit input) which will scale your k-values. The k-values will also vary depending on which donkey you chose.

1. It is important to know what angle your system is at. Find "Offset", the value for the digital output such that when the mass is at x_0 the angle is θ_0 ; this will vary slightly depending on the direction of movement of the sliding mass. Calibrate the system to find the relationship between potentiometer reading and arm pointer angle. You can choose to use radians or degrees; radians are generally better for a physics experiment, but the equipment is marked in degrees, so using degrees may be more practical. Update the VI to display the angle of the pointer rather than potentiometer reading (you can use numeric functions, or program a formula using the express VI).

Give your value of Offset, and present your calibration.

2. You need to understand the damping of your system before you add control mechanisms. Investigate the behaviour of the system with no feedback control. You will need to output the time and angle data to a file. Figure 4 shows a suggested block diagram using Dynamic Data to save angle and timing information to a file (the output to the arm control motor is useful for setting the equilibrium angle to the centre). The delay value should be tested; your system may work best with a smaller delay.

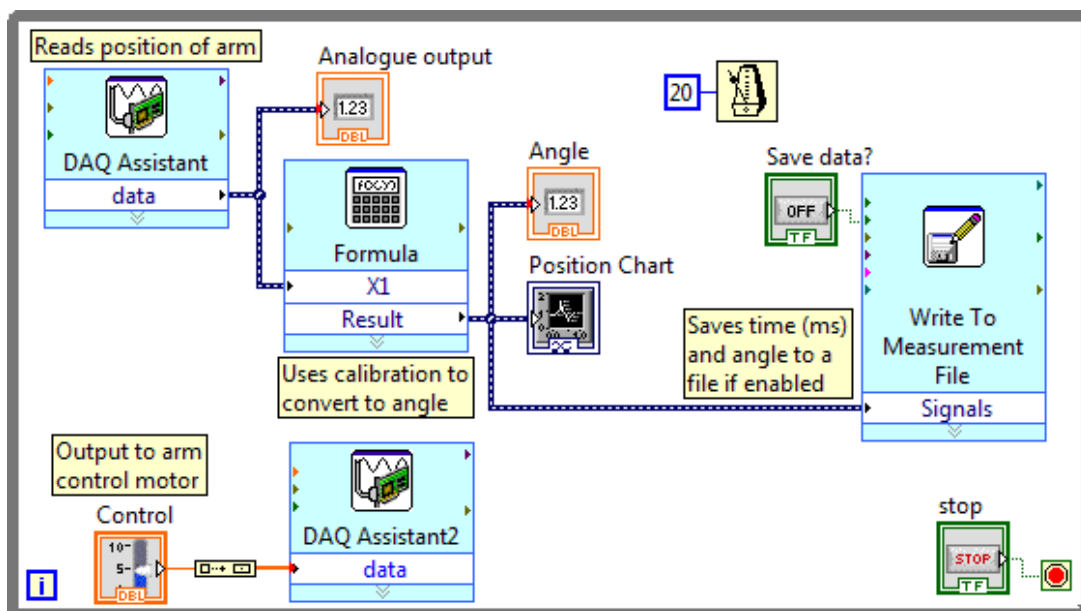


Figure 4 Calibrated routine for moving arm experiment, with output for time-angle data

Explore the settling motion of the arm after an initial displacement, and calculate μ and ω_0 (see theory); figures and analysis should be included in your lab book.

3. The first donkey control mechanism to program is derivative (damping) control. Following equations 4 & 6 we assume an extra torque of the form $(mg/I)\Delta x_D = k_D \dot{\theta}$ due to moving the mass. On each iteration the angle changes and the extra torque can be approximated

as $k_D \Delta\theta/\Delta t = k_D(\theta_i - \theta_{i-1})/\Delta t$. Figure 5 shows a *suggested* routine implementing this equation; it is simpler to use “Convert from dynamic data” to get the angle before doing calculations, and the output to the motor should include your offset (to set x_0). Investigate the optimal value of k_D for bringing the donkey to rest, and which values give an unstable system. You may wish to investigate the effect of adding a perturbing mass. Note that your k_D value will be scaled so that the output to the motor is in the correct format.

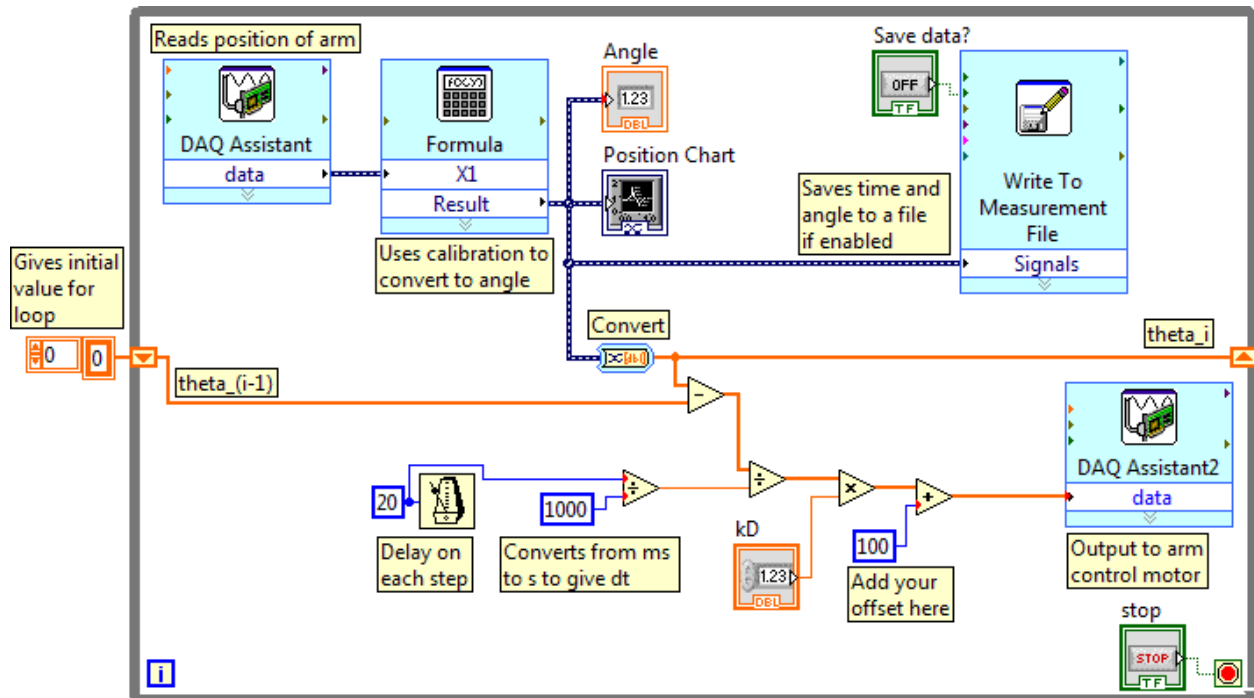


Figure 5 Moving arm routine with Derivative control.

What sign should k_D have? (consider the theory). Should there be any constraint on the number sent to the motor (“coerce” in LabVIEW)? Present your results for finding the optimal k_D and justify your choice using graphs or otherwise. Did you achieve critical damping? If not, why not? To what extent is the simple model from the theory valid here?

4. This has (hopefully!) damped the system, but you need to add controls to set the angle. Add Proportional control, following the equations in section 1.1. You will need to create a control to set the chosen angle, θ_T . The output to the motor should consider offset, derivative, and proportional control.

How does the behaviour depend on k_P , and is it always stable (for a good value of k_D)? Do you expect the arm to reach θ_T ? For your optimal value of k_P produce a plot of final angle achieved against target angle. How well does it work with a perturbing mass?

5. Add integral control. Investigate its effect by itself and with the other forms of control. If you are stuck, consider how the differentiation in equation 4 was done for derivative control.

Present and discuss your results – it may be useful to produce plots of the motion of the arm, and/or the angle achieved vs. target angle.

6. Use your PID control program to set the system to a chosen target angle. You may need to alter k_P , k_I and k_D when all three mechanisms are used. **When your program is completed show it working to a demonstrator** – successful implementation will also be demonstrated by the results and final program printout presented in your lab book.

Again, present and discuss your results, and show that your program works when perturbed by knocking the arm and/or adding extra weights.

Integral control is just the integral of proportional control. Why then does integral control work? Why do we always include proportional control?

3 FURTHER INVESTIGATIONS

You now have a fully working donkey, with the motion damped and controlled. Congratulations! You must now investigate a section of the work in more depth. Choose **one** of the following options (please talk to a demonstrator if you need some advice);

1. *Experiments*: Zeigler and Nichols developed a method of setting the k-values of the system. Investigate this method (see references). How well does it control your system? How do the values compare to those you found?
2. *Theory and experiments*: the k-values you have found include a conversion factor to convert from the calculations using angles in degrees to the format required by the 8-bit motor. Investigate the scaling factor for the different k-values by comparing experiments and theory.
3. *Programming*: A program which stops when the arm is at rest would be very useful, and would save you having to press the stop button. Attempt to program this. You will need to consider how you define your arm to be at rest, and explore the logic and comparison sections of LabVIEW.
4. *Background*: Previously, PID was done using e.g. analogue electronics or pneumatics. How was this done? Compare these methods with computer control. NOTE: this choice requires you to think about the methods, and not just copy from books / Wikipedia!

We love giving extra credit. If you finish the main experiment and have done one of the above choices, then feel free to choose from the following for extra marks.

1. Anything else from the previous list.
2. *Programming*: It may be desirable to set your donkey to go to several different angles (see e.g. heating, and setting different temperatures for day and night). Attempt this. You will need to have thought about how to stop your program (option 3 above), and think about your angle input method, or using sequence structures, or...
3. *Theory and experiments*: you may be able to relate the total mass of your system to the value of ω_0 . We have lots of different weights which can be added to the system. Consider the value for ω_0 given after equation 5 – can you get M ? Does changing the mass affect k_D ?
4. *Theory and programming*: LabVIEW has a built in express vi for looking at the frequency content of the oscillations (in spectral analysis) by performing a fast Fourier transform (FFT). Investigate this, and see whether you can measure the damping at different times for no control.
5. Something else of your choice...

4 ASSESSMENT; A CHECKLIST

A good final program, one further investigation and a well-presented lab book will give you a good mark. ***Even though the laboratory work is done in pairs you should prepare your own lab book for assessment.*** Approximately one third of the marks will be given to the final program from section 2, split between how well it works, layout (including annotation) and correct use of code. The rest of the marks will be based on your lab book work, taking into account results, theory and discussion. You should **print** programs which you haven't been given here for inclusion in your lab book. To do this choose *File » Print » Choose file » Icon, description, panel & diagram » RTF file → Print*, which will give you a word-readable file with figures.

The report of your work should include (at least!) the following;

- **your name and kit number**
- brief notes and results as requested in the Introduction to LabVIEW
- calculations and derivations as requested in section 1 of this handout
- experimental results and discussion for the main experiment (section 2)
- **the front panel and block diagram for the final PID routine**
- your chosen work from section 3
- any references you have used, other than the script

plus any other information you feel is relevant.

REFERENCES

There are several LabVIEW 8 textbooks in the cupboard in the lab; these are now slightly out of date but will still help with the more advanced programming questions! The library also has several LabVIEW textbooks. Other useful resources may include books or papers on control mechanisms; several are given here. The control engineering books discuss the Zeigler Nichols method, along with earlier PID methods.

- LabVIEW 8 Student Edition, Robert Bishop, Prentice Hall, London (2006)
- LabVIEW for Everyone, Jeffrey Travis and Jim Kring, Prentice Hall, London, 3rd edition (2006)
- University Physics, Young & Freedman, Addison Wesley (any year)
- Introduction to Control System Technology, R.N. Bateson, Prentice Hall, London (1996)
- Control Engineering, C.C. Bissel, Chapman & Hall, London (1994)
- Control Engineering, W. Bolton, Longman (1998)

It can be useful to get information from websites, but be aware that not everything is official or even right!

<http://www.ni.com> is the official National Instruments site

<http://www.controleng.com> – a search for PID brings up a lot of useful articles.