MS150 Modular Servo Workshop - Getting Started

33-008-1M5

Feedback

Computer Assisted Learning

Electricity & Electronics

Control & Instrumentation

Process Control

Mechatronics

Robotics

Telecommunications

Electrical Power & Machines

Test & Measurement

Technology Training for tomorrow's world

MS150 Modular Servo Workshop

Getting Started

33-008-1M5

Feedback

Feedback Instruments Ltd, Park Road, Crowborough, E. Sussex, TN6 2QR, UK.
Telephone: +44 (0) 1892 653322,   Fax: +44 (0) 1892 663719.
email: feedback@fdbk.co.uk   World Wide Web Homepage: http://www.fbk.com

Manual: 33-008-1M5 Ed02 991116   *Printed in England by FI Ltd, Crowborough*

Feedback Part No. 1160–330081M5

Notes

THE HEALTH AND SAFETY AT WORK ACT 1974

We are required under the Health and Safety at Work Act 1974, to make available to users of this equipment certain information regarding its safe use.

The equipment, when used in normal or prescribed applications within the parameters set for its mechanical and electrical performance, should not cause any danger or hazard to health or safety if normal engineering practices are observed and they are used in accordance with the instructions supplied.

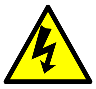
If, in specific cases, circumstances exist in which a potential hazard may be brought about by careless or improper use, these will be pointed out and the necessary precautions emphasised.

While we provide the fullest possible user information relating to the proper use of this equipment, if there is any doubt whatsoever about any aspect, the user should contact the Product Safety Officer at Feedback Instruments Limited, Crowborough.

This equipment should not be used by inexperienced users unless they are under supervision.

We are required by European Directives to indicate on our equipment panels certain areas and warnings that require attention by the user. These have been indicated in the specified way by yellow labels with black printing, the meaning of any labels that may be fixed to the instrument are shown below:

CAUTION -
RISK OF
DANGER



CAUTION -
RISK OF
ELECTRIC SHOCK

CAUTION -
ELECTROSTATIC
SENSITIVE DEVICE

Refer to accompanying documents

PRODUCT IMPROVEMENTS

We maintain a policy of continuous product improvement by incorporating the latest developments and components into our equipment, even up to the time of dispatch.

All major changes are incorporated into up-dated editions of our manuals and this manual was believed to be correct at the time of printing. However, some product changes which do not affect the instructional capability of the equipment, may not be included until it is necessary to incorporate other significant changes.

COMPONENT REPLACEMENT

Where components are of a ‘Safety Critical’ nature, i.e. all components involved with the supply or carrying of voltages at supply potential or higher, these must be replaced with components of equal international safety approval in order to maintain full equipment safety.

In order to maintain compliance with international directives, all replacement components should be identical to those originally supplied.

Any component may be ordered direct from Feedback or its agents by quoting the following information:

1. Equipment type	2. Component value
3. Component reference	4. Equipment serial number

Components can often be replaced by alternatives available locally, however we cannot therefore guarantee continued performance either to published specification or compliance with international standards.

CE DECLARATION CONCERNING ELECTROMAGNETIC COMPATIBILITY

Should this equipment be used outside the classroom, laboratory study area or similar such place for which it is designed and sold then Feedback Instruments Ltd hereby states that conformity with the protection requirements of the European Community Electromagnetic Compatibility Directive (89/336/EEC) may be invalidated and could lead to prosecution.

This equipment, when operated in accordance with the supplied documentation, does not cause electromagnetic disturbance outside its immediate electromagnetic environment.

COPYRIGHT NOTICE

ACKNOWLEDGEMENTS

Feedback Instruments Ltd acknowledge all trademarks.

IBM, IBM - PC are registered trademarks of International Business Machines.

MICROSOFT, WINDOWS 95, WINDOWS 3.1 are registered trademarks of Microsoft Corporation.

MATLAB is a registered trademark of Mathworks Inc.

TABLE OF CONTENTS

1. INTRODUCTION AND DESCRIPTION

The **Modular Servo Workshop (MSW)** is an open-architecture software environment for real-time control experiments using the Feedback© digital servomechanism. The main concept of the Modular Servo Workshop was to create a rapid and direct path from control system design to hardware implementation. The Modular Servo Workshop supports the real-time design and implementation of advanced control methods, using MATLAB© and Simulink tools, and extends the MATLAB environment in the solution of digital servomechanism control problems.

The integrated software supports all phases of a control system development:

- on-line process identification,
- control system modelling, design and simulation,
- real-time implementation of control algorithms.

The Modular Servo Workshop is intended to provide a user with a variety of software tools to facilitate:

- on-line information flow between the process and the MATLAB environment,
- real-time control experiments using embedded algorithms,
- development, simulation and application of user-defined control algorithms.

The Modular Servo Workshop uses standard PC hardware platforms and Microsoft Windows operating systems. Version 1.02 of the software works with MATLAB 5.1 & 5.2 and Simulink 2.1 and 2.2. It is not compatible with earlier versions of MATLAB software.

1.1. Equipment and Requirements

The Modular Servo Workshop is distributed in compressed format on a CD-ROM. Installation procedure is a standard one applied for Feedback MATLAB compatible products (see The Software Installation Guide – 33-000 for details). A full set of software and manuals consists of:

• CD-ROM,		
• Getting Started	33-008-1M5	(this manual),
• Reference Guide	33-008-2M5	
• External Interface	33-008-3M5	
• Advanced Teaching Manual,	33-008-4M5	
• Software Installation Guide.	33-000M5	

1.1.1. Knowledge Level Required

The manual assumes that the user has a basic experience with one of the Feedback digital servomechanisms, with MATLAB 5 & Simulink from MathWorks Inc, and with the Microsoft Windows 95 or NT operating System. More experience in control is required for experiments described in the ADVANCED TEACHING manual. Some practise in C-language programming is expected for application of the EXTERNAL INTERFACE.

1.1.2. Minimum Hardware and Software Requirements

Modular Servo system - Feedback part numbers

OA150A	Operational Amplifier
AU150B	Attenuator
PA150C	Pre-Amplifier
SA150D	Servo Amplifier
PS150E	Power Supply
DCM150F	DC Motor
IP150H	Input Potentiometer
OP150K	Output Potentiometer
LV150L	Magnetic Brake Load
GT150X	Gearbox/Tachometer
MS150Z	Accessories plus Magnetic Baseplate
33-301	Interface unit
33-300	Digital Encoder

PC and boards

486DX or above PC with 16 Mbytes memory, and 200 Mbytes available disk space, 16x CD Rom, SVGA screen,

PCL-812PG or RT-DAC data acquisition board

Software

The MS150 uses a standard PC hardware platform and the standard Microsoft Windows 95 or NT operating system, and *minimum* MATLAB 5.1, Simulink 2.1, Control and Signal Processing Toolboxes.from the Mathswork Inc. Certain specialised controllers, developed in future Teaching Manual releases, may require the use of *additional* MATLAB toolboxes. An appropriate C compiler is required for use of the External Interface.

CD-ROM with Feedback Modular Servo software

1.2. Principal system interconnections and software operation

The control system applied for the servomechanism is given in a block diagram form in Figure 1-1.

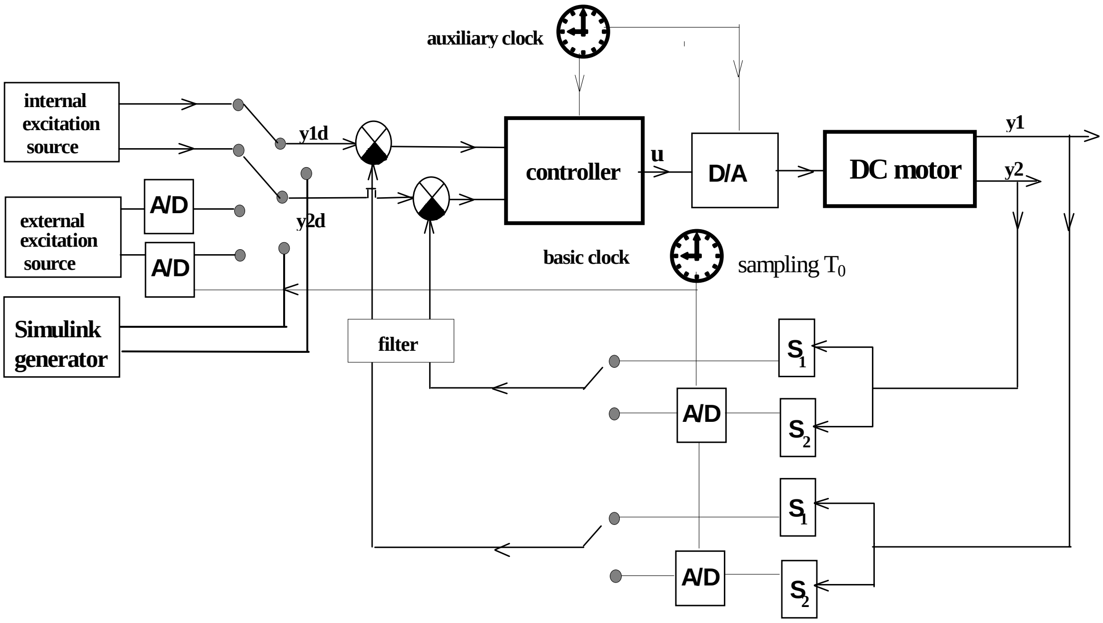


Figure 1-1: Block diagram of the control system

Two process outputs are considered: the position y1 and the speed y2. Process outputs can be measured as continuous signals (sensors S2) digitised by analogue-to-digital converters (A/D) or a direct digital measurement technique can be used (sensors S1 - encoder). The reference input (desired value) can be generated in a digital form using an internal excitation source or, alternatively, an external MS150H Input potentiometer can be applied. Additionally, the Simulink Signal Generator can be applied as a source of excitation signal when the Simulink model is used.

The hardware and software connections of the *Modular Servo Workshop* are given in Figure 1-2.

The software of the MSW consist of the following main parts:

- *Real-time Kernel (RTK),*
- *Modular Servo Toolbox.*

The Real-time Kernel is collection of real-time tasks supervised by a task manager. Control algorithms, filtering procedures and outputs from the software generator of the excitation signal, are activated by the timer interrupt. Notice, that, inside the real-time kernel, the basic feedback loop of the control system is closed.

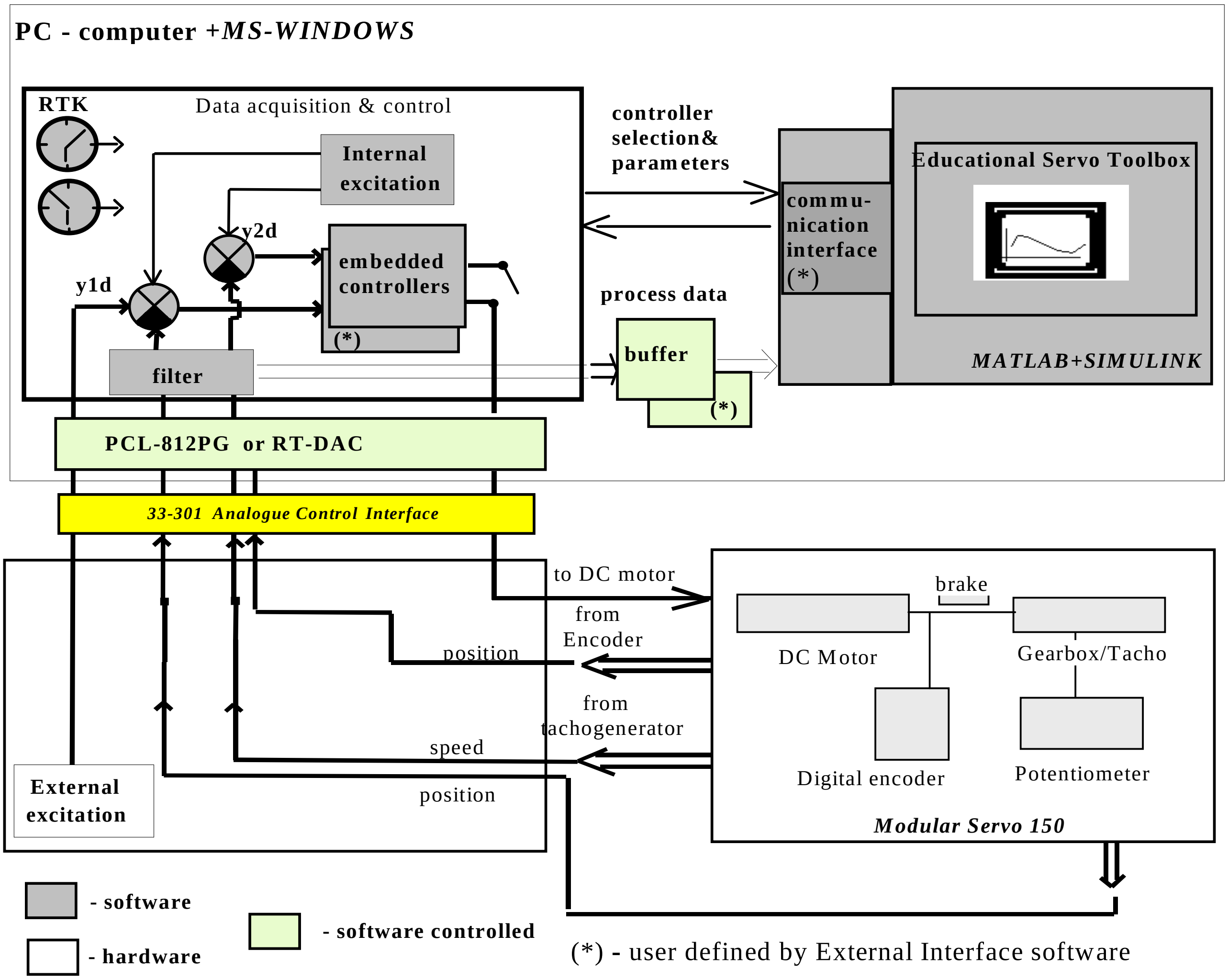


Figure 1-2: Closed-loop control system of the Modular Servo

The *Modular Servo Toolbox*, using MATLAB matrix functions, provides the functions specialised for real-time control of the Feedback digital servomechanism. It is the general assumption that the toolbox is an open system. This approach by its nature forces the basic functions of the toolbox to be in the user domain. It enables the user to maintain his own system, and to customise control algorithms to satisfy the requirements better.

In addition, however, in the *Main Control Window* typical "paths" through the toolbox are provided, and typical control problems are introduced.

The collection of the toolbox functions can divided into two main categories:

- Real-Time Kernel communication functions,
- Real-Time Kernel configuration functions.

The toolbox functions and MATLAB functions can be used at different application levels. In this manual the following convention is used:

- application level 1 - RTK communication and configuration functions, MATLAB functions,
- application level 2 - menu-driven experiments and simulations,
- application level 3 - external interface software for advanced users to create his own algorithms.

Application level 1 means a direct use of the toolbox functions controlling the information flow between process sensors, RTK and MATLAB environment. The functions used at this level configure a real time kernel, embedded controllers and excitation sources (see Reference Manual – 33-008-2M5 for details).

At application level 2 the function ***es*** initialises the menu-driven part of the toolbox .

At application level 3 the EXTERNAL INTERFACE software allows the creation of user-defined algorithms and the modification of communication functions (see the External Interface Manual 33-008-3M5 for details).

Notes

2. MODEL AND CONTROL ALGORITHMS

In this section a brief review of the embedded control algorithms is given. For a detailed description of control algorithms refer to the Advanced Teaching Manual (33-008-4M5).

The design process for the controllers listed below was performed assuming a linear model of the DC motor in the form of two first-order linear differential equations. This representation is generally known as a state-space model:

$$\frac{dx}{dt} = Ax + Bu$$
$$y = Cx + Du$$
$$x(t = 0) = x_0$$

For the DC motor we obtain:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{T_s} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \frac{K_s}{T_s} \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad D = 0$$

where: x is the vector with components x_1 (position of the DC motor shaft or angle) and x_2 (the speed) and u is the control input.

The general solution of the state-space equations is:

$$y(t) = C\left\{ e^{At}x0 + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau \right\} + Du(t)$$

The discrete model of DC motor can be obtained by transformation of the continuous system
dx/dt = Ax + Bu assuming a zero-order hold and sample time T0:

$$x[n+1]=A_d\, x[n] + B_d\, u[n]$$
$$y[n] = C_d\, x[n]$$

The system can be classified as a multivariable (SIMO) because it has two measurable states and one control variable.

The following control algorithms are used for the first control experiments:

- PID
- state feedback controllers : LQ and deadbeat
- time-optimal
- adaptive controller

The user can develop and apply his own algorithm using the tools described in the External Interface Manual - 33-008-3M5.

2.1. PID control

The PID controller has the form:

$$u = K_p(\varepsilon_1(k) + K_i \sum_{k=0}^{n} \varepsilon_1(k) + K_d[\varepsilon_1(k) - \varepsilon_1(k-1)])$$

where:

ε_1 is position error, K_p is gain coefficient, K_i is integration gain, and K_d is derivative gain.

The constants Kp, Ki and Kd can be chosen according to the Ziegler-Nichols rule or another controller designing procedure.

2.2. State feedback control

The state feedback controller has the form:

$$u = -(K_1\varepsilon_1 + K_2\varepsilon_2)$$

where:

ε_1 - is position error,
ε_2 - is speed error,
K_1,K_2 are controller gains.

The LQ optimal feedback gain vector K = [K_1,K_2] can be calculated in such a way that the feedback law u = -Kε; where ε = [ε_1,ε_2] minimises the cost function:

$$J = \int_0^\infty (x^T Qx + u^T Ru)\,dt\,,$$ where Q and R are the weighting matrices,

subject to the state equation: dx/dt = Ax + Bu

Another selection of state-feedback gains is based on 'dead-beat' principle. The feedback gain vector K = [K_1,K_2] is calculated in such a way that eigenvalues of the state matrix of the discrete closed-loop system are equal to zero:

$$\mathrm{eig}(A_d - B_d * K) = 0$$

The controller has the property that he drives the states of the closed-loop system from an arbitrary initial state to zero value in at most N-steps (N = dim(A_d)).

2.3. Time - optimal control

In this case typical for linear systems „bang-bang” control principle is used. A full description and principles of operation of the time-optimal controller are given in the Advanced Teaching Manual – 33-008-4M5.

2.4. Adaptive control

The adaptive controller is based on the reference model idea. A full description of the controller is given in the Advanced Teaching Manual - 33-008-4M5.

Notes

3. STARTING , TESTING AND STOPPING PROCEDURES

3.1. Starting procedure

Perform all connections as described in Section 6 of this manual. They will ensure correct communications between Computer – Advantech PCL812PG i/o board – 33-301 Analogue Interface unit and the rest of modular servo MS150 units.

Ensure that the Attenuator Unit AU150B has the top gain-pot set to a value around 8 and the Pre-Amplifier Unit PA 150C has the pot to ‘zero set’. The connections to these units to be made as shown in Section 6. There should be approximately ‘no motion’ noticed on the motor unit.
If the motor is turning, then the output produced by the computer (i/o board) is set to high. Follow the Calibration 33-301 routine in order to adjust offsets and gains for all 4 channels.

Note: Values stated in these manuals would not provide good performance on other systems. The numbers should be around the ones shown in manual.

In the MS-WINDOWS environment invoke MATLAB by double clicking on the MATLAB icon. The *Matlab Command Window* opens. Then type:

es

and then the *Main Control Window* of MSW opens (see Figure 3-1)

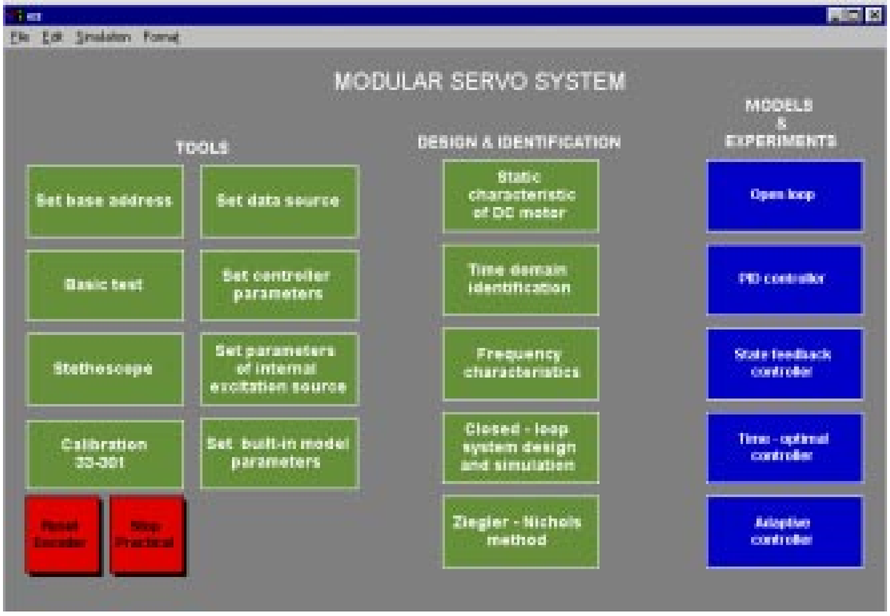
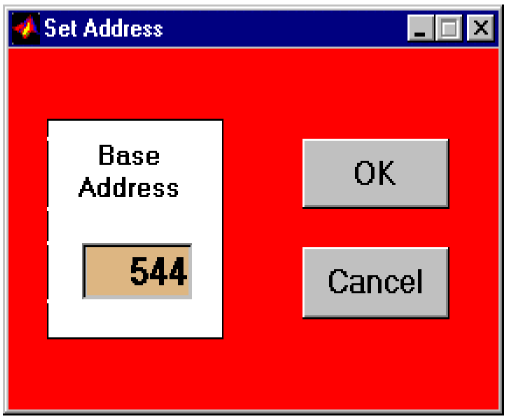

File Edit Simulation Format
MODULAR SERVO SYSTEM
TOOLS
DESIGN & IDENTIFICATION
MODELS & EXPERIMENTS
Set base address
Set data source
Basic test
Set controller parameters
Stethoscope
Set parameters of internal excitation source
Calibration 33-301
Set built-in model parameters
Reset Encoder
Stop Practical
Static characteristic of DC motor
Time domain identification
Frequency characteristics
Closed - loop system design and simulation
Ziegler - Nichols method
Open loop
PID controller
State feedback controller
Time - optimal controller
Adaptive controller

Figure 3-1: The Main Control Window

Then go through the following steps:

- double click on the *Set base address* button. This set the base address of the i/o card.



Base Address

544

- If the address of your data acquisition board agrees with the default value then accept it, otherwise type the proper address. System may terminate if entered the incorrect number.

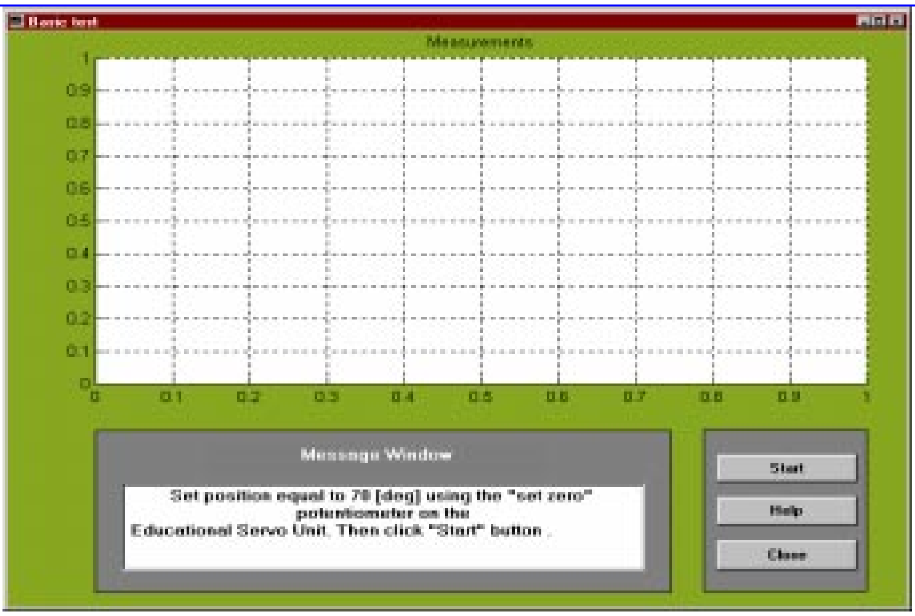
> ***Note: You can set the base address equal to zero. Then you can work with the control software without a connection to the external hardware. In this case the measurement data are produced by a built-in model of the DC motor, which allows off-line testing of new control software.***

Calibrate the 33-301 Analogue Control Interface Unit following step-by-step instruction brought up by each individual window.

3.2. Testing and troubleshooting

Now you are ready to start an experiment. This will give an indication if the output measurements are correct. First go through the following steps to check the proper operation of your hardware:

- double click the *Basic test* button. The following window (Figure 3-2) appears:



Set position equal to 70 [deg] using the "set zero" potentiometer on the
Educational Servo Unit. Then click "Start" button .

Figure 3-2: *Basic test* window

- set the position of the OP150K manually to 70 degrees anticlockwise. If this is not set to 70 degrees, the Basic Test program will not provide results similar to the ones shown in Fig. 3.3. This is because when applying a square wave (sudden reverse in the input signal) the pot on the OP150K will rotate more than 180 degrees.

- click the *Start* button and wait for results.

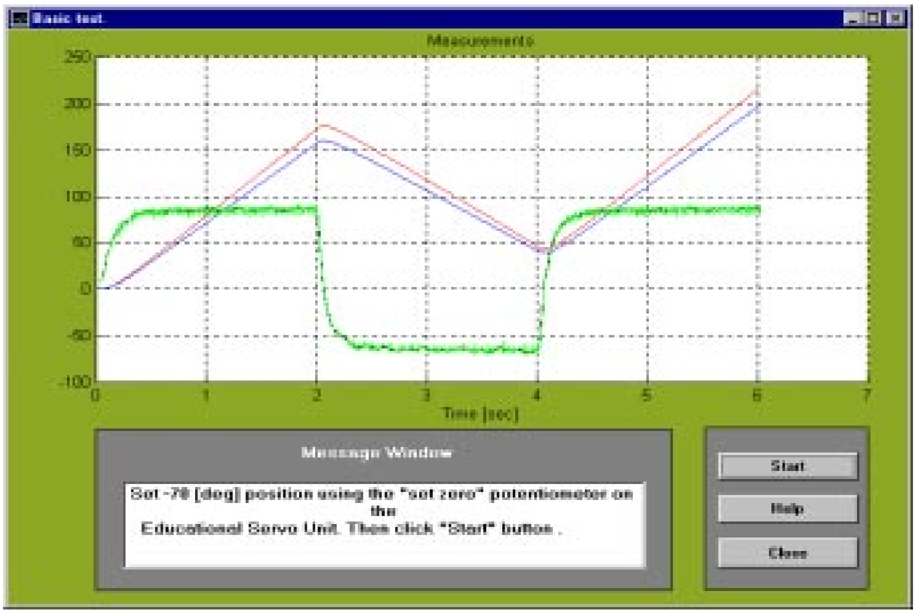
This test performs the action in three steps:

1. sets the source of data as follows:
    - Channel 1 - angle measured from A/D converter [deg] (using A/D1 channel of the Analogue Control Interface),
    - Channel 2 - angle measured from digital encoder [deg],
    - Channel 3 - velocity measured from A/D converter [deg/s] (using A/D2 channel of the Analogue Control Interface),
    - Channel 4 - velocity calculated from encoder measurements [deg/s].
2. sends a step input to system, collects the data,
3. plots the results.

Typical results of the basic test are shown in Figure 3-3.

On the screen you will see the following colours: red-channel 1, blue-channel 2, green-channel 3 and black-channel 4. In this case all components of the system are working correctly.

If the results of your test are not similar you must check the hardware connections (see section 6).



Set -70 [deg] position using the "set zero" potentiometer on the
Educational Servo Unit. Then click "Start" button .

Figure 3-3: Results of the basic test (example)

Below, we give a number of hints to avoid most common faults that can occur during the first experiments with the digital servo (see Table 3.1).

Table 3.1. Troubleshooting

Message	Solution
Undefined function or variable *es_call*	Check if *es_call.dll* file exist. Otherwise repeat installation process.
GetHistory - memory allocation error	Not enough memory to perform *GetHistory* function. Free some memory in your Windows system (e.g. close unused applications)
Cannot open *es_par.ini* file. Base address is set to zero.	Create the file named *es_par.ini*. using the Windows Notepad program or copy this file from the CD-ROM into your MATLAB\FEEDBACK subdirectory.

3.3. Stopping procedure

Any practical can be stopped at any time. Double click the *Stop Practical* button in the *Main Control Window*. If you wish to interrupt the visualisation process for any time click once on the *Pause* item in the *Simulation* menu (located on the top menu bar of the screen)

Notes

4. MENU DESCRIPTION

The *Main Control Window* shown in Figure 3-1 contains three groups of menu buttons:

- TOOLS
- DESIGN & IDENTIFICATION
- MODELS & EXPERIMENTS

Generally, the *Main Control Window* was developed for application level 2. However, some of the windows activated by buttons from the TOOLS group can support the experiments performed from application level 1, e.g. *Set Controller Parameters, Set Parameters of Internal Excitation Source.*

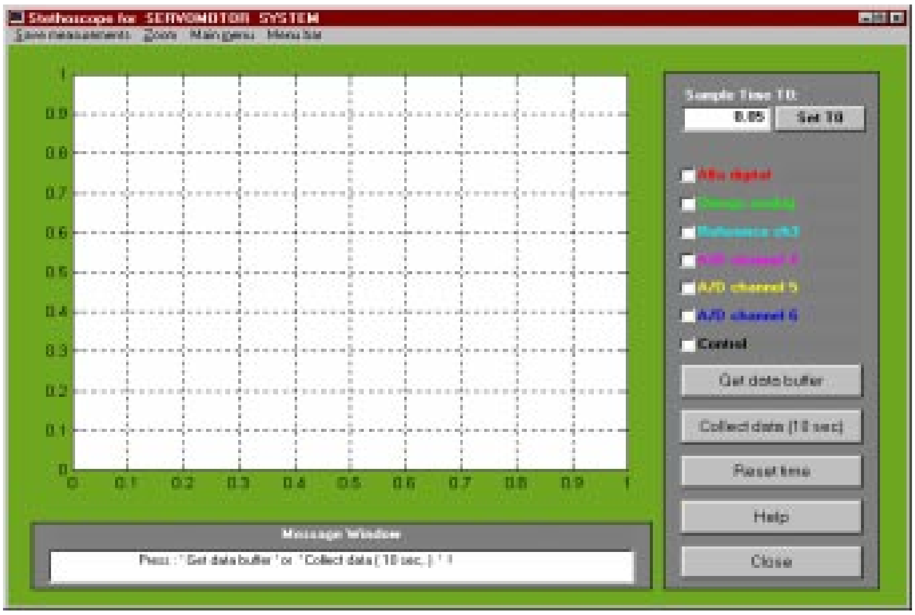
4.1. Tools

The respective buttons in the TOOLS column perform the following tasks:

1) *Set base address* - sets the base address of the I/O board,

2) *Basic test* - checks the measurements.

3) *Stethoscope* - opens the oscilloscope-like (with 7 channels), window for the system (Figure 4-1)



Press : ' Get data buffer ' or ' Collect data (10 sec.) ' !

Figure 4-1: *Stethoscope* window

Using this tool you can collect and plot real-time data. This tool allows you to:

- set sample time,
- read contents of the data acquisition buffer,
- clear buffer and collect data for 10 seconds,
- plot measurement channels and control . Six measurement channels selected by check boxes and control can be plotted. The accessibility of signals in this window can be changed using *SetDataSource* button.

In this case defined signals are:

channel 1 - position from encoder	(Red)
channel 2 - speed from A/D converter	(Green)
channel 3 - reference input from A/D converter	(Cyan)

You can also use the following edit windows and pushbuttons on the right side of the window:

Sample Time T0	- sampling time setting ,
Set T0	- transfer T0 to RTK,
Get data buffer	- sends data to MATLAB and display marked data,
Collect data (10 s)	- 10 sec data acquisition,
Reset time	- sets time to zero.

In order to use the power of this tool, the following steps are required.

- Set, for example, a square wave input of 2 seconds period from the *Set Parameters Internal Excitation Source block* of the Main Control menu. Click Set and then leave it open in order to use the *Update* feature for visualising the input parameters being used.
- Choose any of the Real-Time experiments from the *Main Control Menu*. Set the Data Source as *Internal Generator* in the *Real-Time-Task* block (i.e. the green block opened up in *Models and Experiments*)
- Start the *Simulation* from the top bar menu.
- After a few seconds click “pause”
- Open the Oscilloscope-like window, Stethoscope and then select signals you would like to visualise.
- Click *Get Data Buffer* and signals will be displayed in the window.

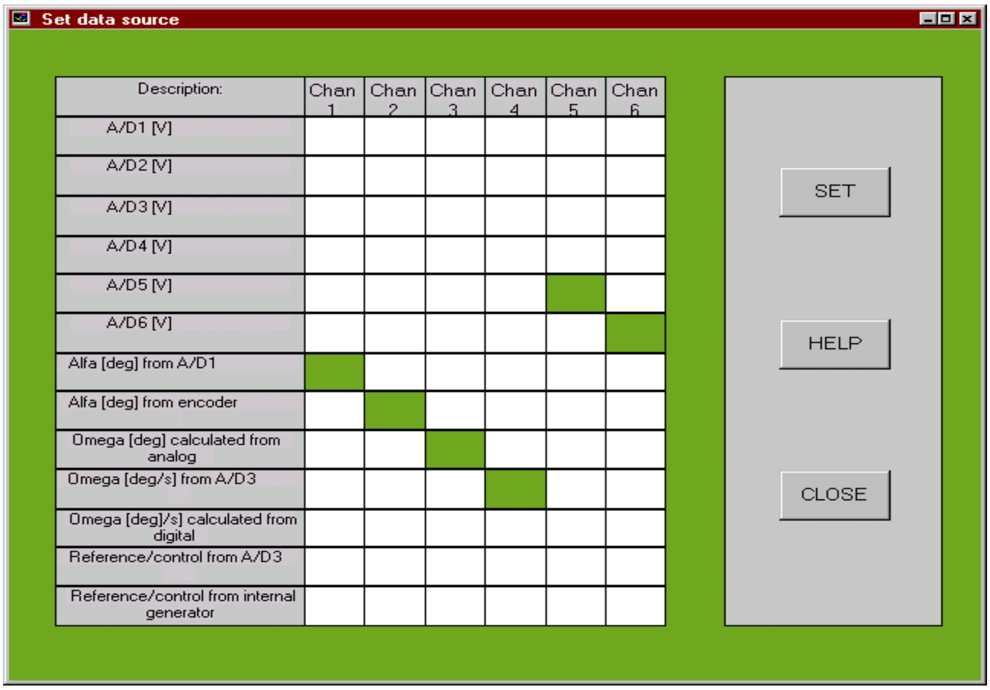
The reason for not being able to monitor the plant while the simulation is running from Simulink is that the display and data transfer conflict between the two applications, However the real-time control is still running when the display process is paused.

4) *Stop practical* - stops the actual control task,

5) *Reset Encoder* - sets encoder to zero for actual position of the servo,

6) *Calibration of 33-301 – calibration of analogue control interface*

7) *Set data source* - defines the source of the process output data and the source of the reference inputs. After clicking on it the window shown in Figure 4-2 opens.



Description:	Chan 1	Chan 2	Chan 3	Chan 4	Chan 5	Chan 6
A/D1 [V]						
A/D2 [V]						
A/D3 [V]						
A/D4 [V]						
A/D5 [V]					🟩	
A/D6 [V]						🟩
Alfa [deg] from A/D1	🟩					
Alfa [deg] from encoder		🟩				
Omega [deg] calculated from analog			🟩			
Omega [deg/s] from A/D3				🟩		
Omega [deg]/s] calculated from digital						
Reference/control from A/D3						
Reference/control from internal generator						

Figure 4-2: *Set data source* window (alfa is angle and omega angular velocity)

In this window you can define the source of data for the six logical data measurement channels available in the RTK buffer (see Figure 4-2).

The following hardware and software data sources can be specified (see Table 4.1).

Table 4.1

Item	Description	Signal source or sink
1	voltage of the signal connected to the analogue input channel A/D1 of the Analogue Control Interface expressed in [V]	150K output pot.
2	voltage of the signal connected to the analogue input channel A/D2 of the Analogue Control Interface expressed in [V]	150X Tacho
3	voltage of the signal connected to the analogue input channel A/D3 of the Analogue Control Interface expressed in [V]	150H input pot.
4	voltage of the signal connected to the analogue input channel A/D4 of the Analogue Control Interface expressed in [V]	
5	voltage of the signal connected to the analogue input channel A/D5 of the Analogue Control Interface expressed in [V]	
6	voltage of the signal connected to the analogue input channel A/D6 of the Analogue Control Interface expressed in [V]	
7	angle expressed in [deg] - signal connected to the analogue input channel A/D1 of the Analogue Control Interface	150K output pot
8	angle expressed in [deg] measured from incremental encoder sensor	Output position
9	angular velocity from tachogenerator connected to the analogue input channel A/D2 of the Analogue Control Interface expressed in [deg/sec]	150X Tacho
10	angular velocity calculated from position signal connected to the analogue input channel A/D1 of the Analogue Control Interface expressed in [deg/sec]	150K output pot
11	angular velocity calculated from position signal obtained from incremental encoder expressed in [deg/sec]	Output velocity
12	reference value from the signal connected to the analogue input channel A/D3 of the Analogue Control Interface expressed in [deg]	150H input pot
13	internal excitation generator from RTK expressed in [deg] as reference signal or in normalised units from the range [-1 to +1] as a control signal.	

In the example shown in Figure 4-2 the data source is defined as follows:

Channel 1- angle expressed in [deg] - signal connected to the A/D1 input channel of the Analogue Control Interface,

Channel 2 - angle expressed in [deg] measured from the incremental encoder sensor,

Channel 3 - angular velocity from tachogenerator connected to the A/D2 input channel of the Analogue Control Interface expressed in [deg/sec],

Channel 4 - angular velocity calculated from the position signal connected to the A/D1 channel of the Analogue Control Interface expressed in [deg/sec],

Channel 5 - signal (if connected) from A/D5 channel of the Analogue Control Interface expressed in [V],

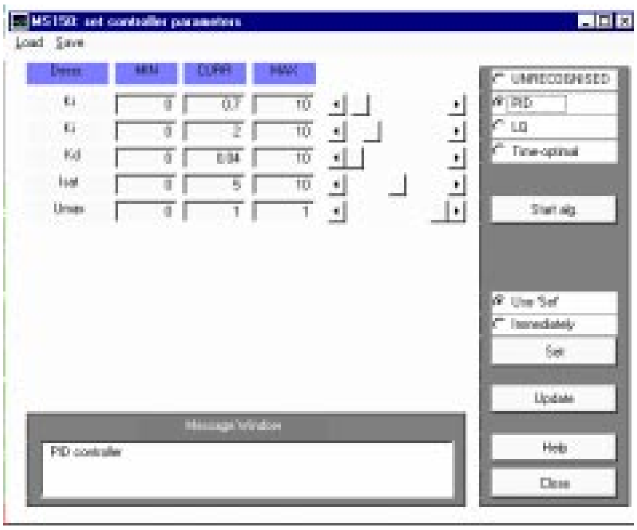
Channel 6 - signal (if connected) from A/D6 channel of the Analogue Control Interface expressed in [V],

See the Reference Manual - 33-008-2M5 for a detailed description.

> ***Note: a definition of the data source using the*** **SetDataSource** ***block is useful for application level 1. For menu-driven experiments, the appropriate S-functions set the data source themselves.***

---

8) *Set controller parameters* - displays the parameters of embedded controllers. The window can be used to select controller and set its parameters before starting the experiment or during the experiment.



Descr.	MIN	CURR	MAX
Kp	0	0.7	10
Ki	0	2	10
Kd	0	0.04	10
Ival	0	5	10
Umax	0	1	1

○ UNRECOGNISED
● PID
○ LQ
○ Time-optimal

● Use 'Set'
○ Immediately

PID controller

Figure 4-3: *Set controller parameters* window

If an arbitrary algorithm is running and this window is then opened, then the algorithm is marked (the black dot) with the current active values of parameters. If none of algorithms are running then the UNRECOGNISED button is marked without any visible parameters.

You can select the type of the algorithm by clicking the *PID, LQ* or *Time optimal* radio-button. Each algorithm has its own set of the default parameters, which appears automatically when the window opens and which can be changed by typing in the appropriate value or by dragging the slider.

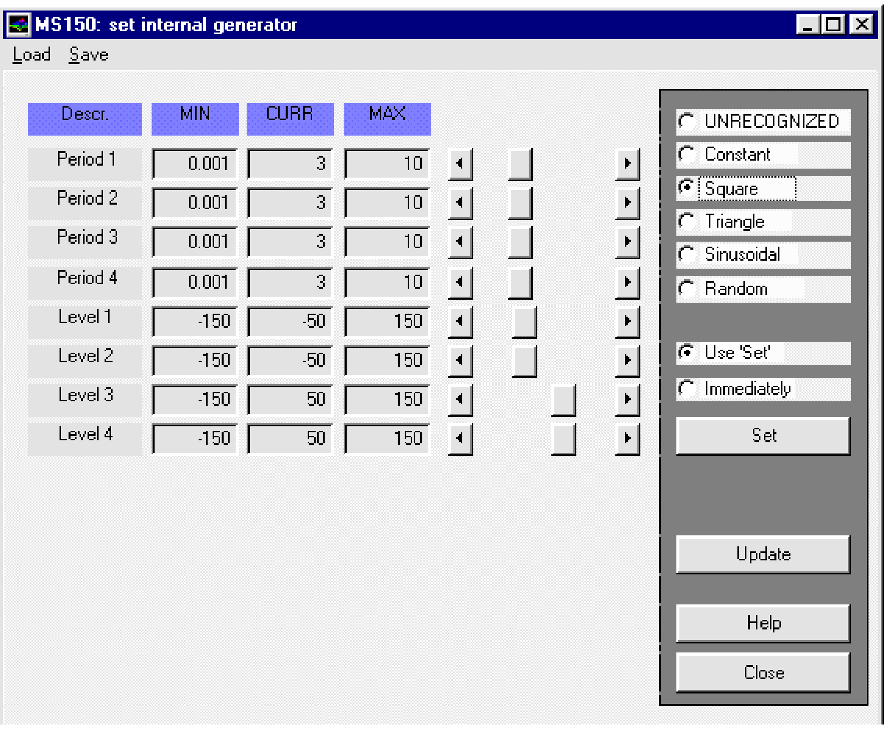
When the *Use 'Set'* button is selected the parameters are passed to the RTK after clicking the *Set* button.

When the *Immediately* button is selected each change is passed to the RTK immediately.

Any set of parameters previously saved, can be loaded using the *Load* menu option. The default parameters are loaded from the *es_cpar.mat* file for inactive algorithms and for the active algorithm (working in the background) are loaded from the RTK.

The Update button is used to show the current parameters of the RTK. If you click the *Update* button then the parameters window is updated. It means that current values of parameters from the RTK are transmitted to the parameters window. Therefore all changes introduced most recently outside the visible window, for example: in the MATLAB Command Window, etc. are transmitted to that window. If you mark a control algorithm and click the *Start alg.* button then the new control algorithm will be chosen and it will start.

9) *Set parameters of internal excitation source* - opens the window to set the shape and parameters of the signals from the internal excitation generator (Figure 4-4).



Descr.	MIN	CURR	MAX
Period 1	0.001	3	10
Period 2	0.001	3	10
Period 3	0.001	3	10
Period 4	0.001	3	10
Level 1	-150	-50	150
Level 2	-150	-50	150
Level 3	-150	50	150
Level 4	-150	50	150

○ UNRECOGNIZED
○ Constant
● Square
○ Triangle
○ Sinusoidal
○ Random

● Use 'Set'
○ Immediately

Figure 4-4: *Set parameters of internal excitation source* window

Here you can set parameters of the internal excitation source. This window allows you to select the shape of the wave generated by the internal excitation source and to set parameters of this wave.

If the *Use "Set" radio-* button is marked then the modification of the signal parameters consists of two steps. First, you can introduce necessary changes in the parameter values. Next, you must activate the new values clicking the *Set* button.

When the *Immediately* radio-button is selected each change is passed to the RTK immediately after a change of any parameter.

The Update button is used to show current parameters of the recently set excitation signal (in RTK). If you click the *Update* button then the parameters of the recently set signal are updated. All changes done outside the visible window are transferred to the window.
You can **save** the current set of parameters using *Save* menu option and **load** them using the *Load* menu option.

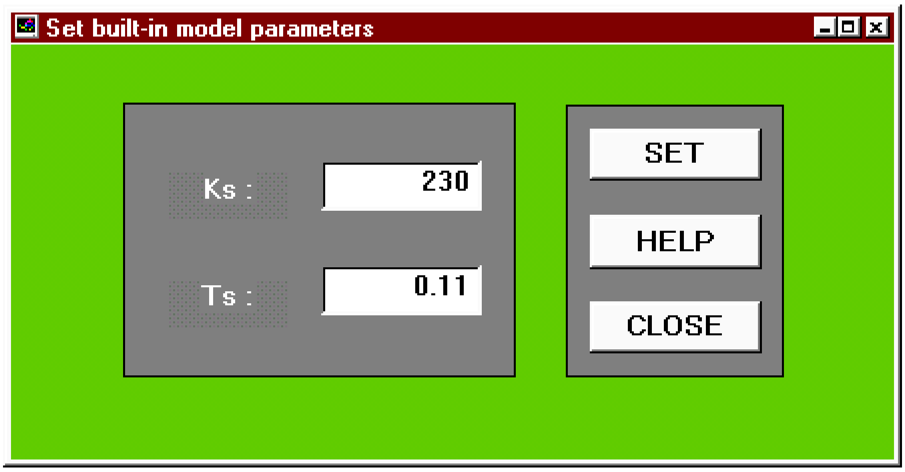
The signals generated by the internal excitation source are interpreted as a control input if the system is in the open-loop mode, or as a reference signal if the system is in the closed-loop mode.

If an arbitrary internal excitation generator is running in RTK and this window is then opened, then the shape of the excitation is marked (the black dot) with the current active values of parameters. If not then the *Constant* radio-button is marked.

The UNRECOGNISED shape is marked in the case of an incorrect usage of the *es_call('setpw',arg)* function (see the Reference Manual – 33-008-2M5), for example in the MATLAB Command Window.

Also see the Reference Manual – 33-008-2M5 for details of the signal parameters.

10) *Set built-in model parameters* - opens the window to set the parameters of the transfer function of the built-in model which works in the background when the base address is set to zero and is only used for demos without hardware.



Ks : 230

Ts : 0.11

Figure 4-5: *Set built-in model parameters* window

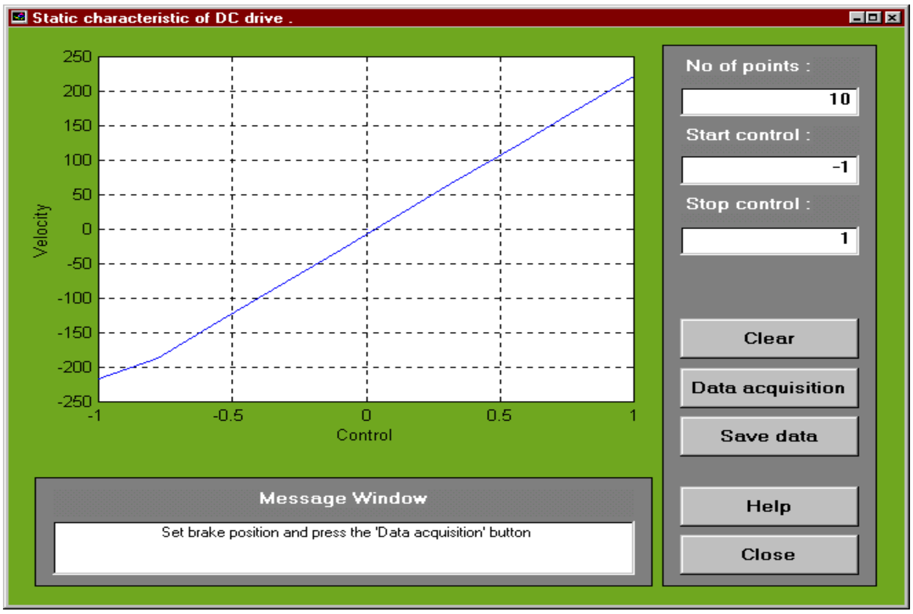
If the address of the data acquisition board is set to zero you can use the control software without a connection to the external hardware. In this case measurement data are simulated by built-in model of the DC motor. This option can be used for off-line testing of new software components. This represents the concept of “hardware-in-the-loop”.

4.2. Design and identification

The respective buttons in the group DESIGN & IDENTIFICATION of the *Main Control Window* perform the following tasks:

1) *Static characteristic of the DC motor* - performs an experiment to measure the static characteristic of the DC motor (rotation vs. input voltage in the steady state).

A Tachogenerator is mounted on the motor shaft and provides a voltage proportional to the motor speed, independent of the load torque applied to the motor. In an ideal motor the steady state speed is ideally proportional to the applied voltage, less an amount proportional to the load torque.



Set brake position and press the 'Data acquisition' button

Figure 4-6: *Static characteristic of DC motor* window

By using the appropriate edit windows and pushbuttons on the right side of the window, you are able to select parameters of the identification experiment:

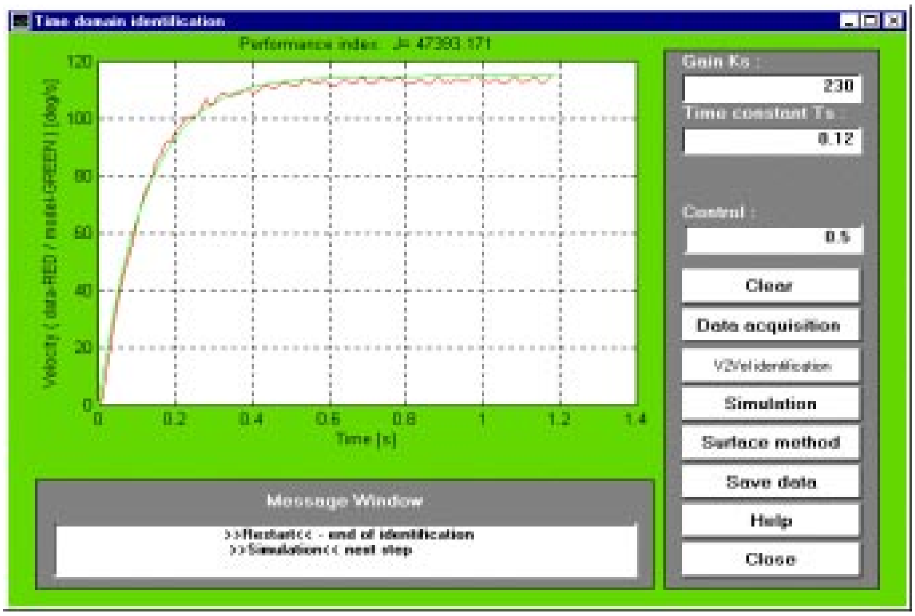
(i)	*No of points*	sets number of measured points on the characteristic,
(ii)	*Start control*	sets minimum control value for DC,
(iii)	*Stop control*	sets maximum control value for DC (remember that control is normalised in the range [-1, +1],
(iv)	*Clear*	clears measurement data and earlier plots,
(v)	*Data acquisition*	starts data acquisition for DC drive characteristic,
(vi)	*Save data*	saves collected data to the mat-file.

The *Message Window* in the bottom gives you auxiliary messages.

Figure 4-6 shows an example of measured characteristics.

2) *Time domain identification* - opens the window to start identification on the base of a step system response (Figure 4-7).

Using this window you can perform the step response experiments for digital servo. The identification of transfer function coefficients can be repeated for different magnetic brake positions, with the real system outputs being compared to those of the model. (See Section 2 for a definition of the model). The assumed transfer function is $\frac{1}{s}.\frac{K_s}{(T_s.s+1)}$



Gain Ks : 230

Time constant Ts : 0.12

Control : 0.5

>>Restart<< - end of identification
>>Simulation<< next step

Figure 4-7: *Time domain identification* window

You can use the following edit windows and pushbuttons on the right side of the *Time domain identification* window:

Gain Ks	gain coefficient of the model,
Time constant Ts	time constant coefficient of the model,
Control	value of step input for identification experiment,
Clear	clears measurement data and earlier plots,
Data acquisition	starts data acquisition for identification using step response method,

V2Vel identification	starts experiment of *V2Velocity* coefficient identification [see the Reference Manual – 33-008-2M5],
Simulation	starts simulation for fixed K_s and T_s,
Surface Method	fits a smooth function of two variables to the real response and calculates the optimum K_S and T_S
Save data	saves collected data to the mat-file.

The steps in the identification sequence are as follows

i) Click the Data acquisition button to acquire real system data

ii) Introduce your own values of K_S and T_S into the edit fields in the right hand side block

iii) Click *Surface Method* which calculates the best fit for K_S and T_S

iv) Introduce these values of Ks and Ts (shown in the message window) into the edit fields in the right hand side block

v) Click *Simulation* to compare the simulated response with the real one

vi) Click *Save Data* to save the data in a .mat file

vii) Click Clear to clear the data and plots ready for another pass through the sequence

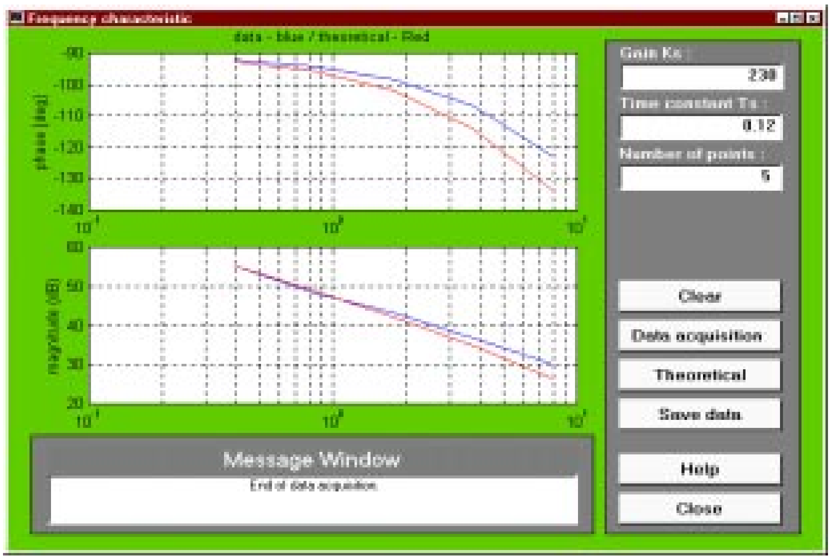
Figure 4-7 shows results of an identification procedure.

3) *Frequency characteristic* - opens the window to start measurement of the frequency response of the DC motor (Figure 4-8).

Here you can perform real-time experiments, collect data, and calculate frequency characteristic of the DC motor, which you can then compare against the frequency characteristics of its model.

You can choose the number of points of the frequency characteristic. The lower bound of frequency is equal to 0.4 [rad/sec] and the upper bound is equal to 12.6 [rad/sec]. These parameters are fixed and can be changed only within the m-file. The model coefficients are available in the edit windows and can be changed after each simulation. The real and theoretical characteristics are displayed.

Edit windows and pushbuttons are very similar to the previous one (Figure 4-7).



Gain Ks : 230

Time constant Ts : 0.12

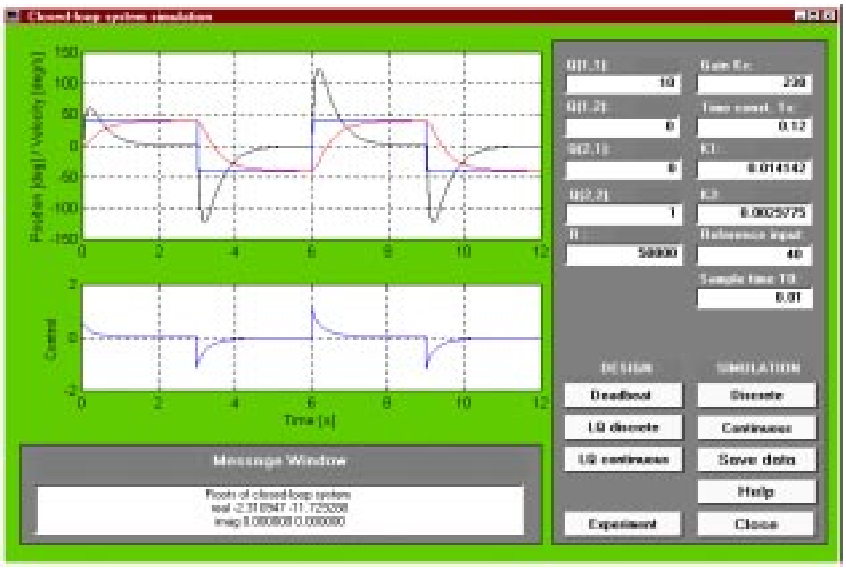
Number of points : 5

End of data acquisition.

Figure 4-8: *Frequency characteristic of DC motor* window

4) *Closed-loop system design and simulation -* opens the window shown in Figure 4-9.

This simulation window allows you to display step response of the linear, closed-loop system with feedback gains from position and velocity. You can design parameters for discrete or continuous LQ and "deadbeat" controllers.



Q(1,1): 10

Q(1,2): 0

Q(2,1): 0

Q(2,2): 1

R: 50000

Gain Ks: 230

Time const. Ts: 0.12

K1: 0.014142

K2: 0.0025775

Reference input: 40

Sample time T0: 0.01

Roots of closed-loop system
real -2.318947 -11.725298
imag 0.000000 0.000000

Figure 4-9: *Closed-loop system design & simulation* window

The following edit windows and pushbuttons located on the right side of the window can be used:

Gain Ks	- model gain,
Time constant Ts	- model time constant,
Sampling time T0	- setting of the sampling time for discrete version calculations,
	K1, K2- position and velocity feedback gains,
Reference input	-reference step input,
Discrete or *Continuous*	- start simulation in discrete or continuous mode,
Save data	- save collected data to the mat-file,
Q and R	- weighting matrices for LQ controller design,
Deadbeat	- calculates deadbeat controller gains,
LQ discrete	- calculates LQ discrete controller gains,
LQ continuous	- calculates LQ continuous controller gains.
Experiment	-calls LQ control demo (state feedback)

LQ controller design (discrete mode)

Clicking the *LQ continuous* or *LQ discrete* button calculates the optimal feedback gain matrix K such that the feedback law u = -Kx minimises the cost function (discrete case):

$$J(u) = \sum_{n=0}^{N} [x^T(n)Qx(n) + u^T(n)Ru(n)]$$

subject to the discrete-state equation:

$$x[n+1]=A_d\, x[n] + B_d\, u[n]$$
$$y[n] = C_d\, x[n]$$

The matrix Q is nonnegative definite, matrix R is positive definite.

You can design and test an LQ controller using the simulation tools available in the window by undertaking the following steps.

i) choose Q and R matrix (i.e. set the values of these)

ii) select sample time T0,

iii) calculate feedback gains,

iv) simulate closed-loop system ,

v) if the control signal saturates, change the Q or R matrices and repeat the steps from 1 to 4.

Deadbeat controller design

It is the control method, unique to discrete systems, in which we calculate feedback gains in such a way that roots of the closed system are equal to zero. This control strategy has the property that it drives the states of a closed-loop system from an arbitrary initial state to zero in at most N steps (dim(Ad)=N). In this method the sample time T0 is the only design parameter. The magnitude of the control variable u can be tuned by changing the sample time T0. For a given range of the reference input step, a suitable sample time can be determined experimentally.

A Deadbeat controller can be designed by undertaking the following steps in the *Closed-loop system simulation* window:

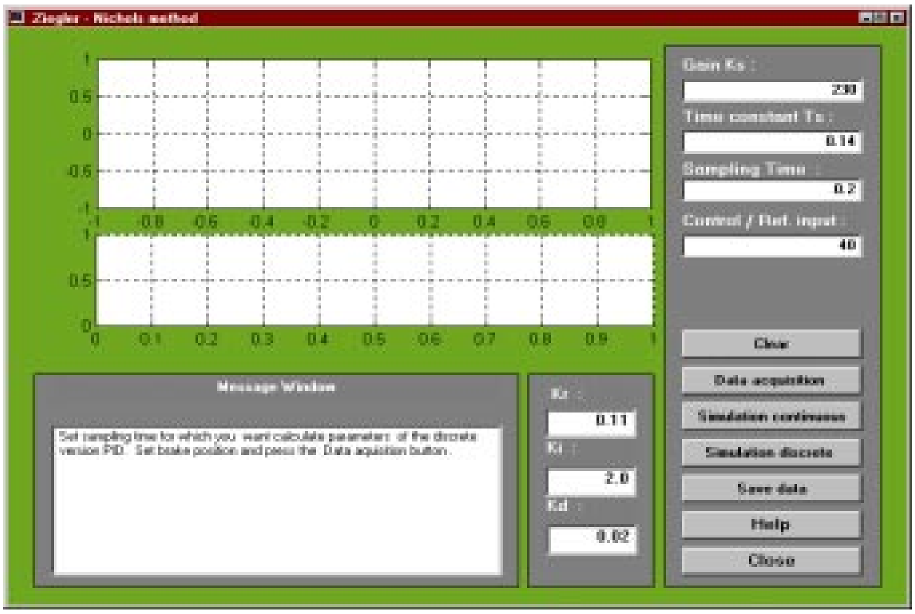
i) select sample time T0,

ii) calculate feedback gains,

iii) simulate closed-loop deadbeat control system ,

iv) if the control saturates increase the sample time T0 and repeat the steps from i) to iv)

5) *Ziegler Nichols method -* opens the window for PID controller design (Figure 4-10).



Gain Ks : 230
Time constant Ts : 0.14
Sampling Time : 0.2
Control / Ref. input : 40

Set sampling time for which you want calculate parameters of the discrete version PID. Set brake position and press the Data aquistion button.

Kr : 0.11
Ki : 2.0
Kd : 0.02

Figure 4-10: *Ziegler - Nichols method* window

This window allows the calculation of the transfer function of the open-loop system step response using second order curve fitting. The parameters of the continuous and discrete PID controller can be calculated for different magnetic brake positions.

Firstly, the acquisition of data must be completed. Then, the experimental time response and Ziegler-Nichols model response are displayed (see Fig.5.2 in the next section). The closed loop system simulation for discrete and continuous version can be started. The simulation is performed for the values of the Kr, Ki and Kd parameters of PID controller, displayed in the edit windows. These values can be edited in the right hand column.

The following edit windows and pushbuttons are situated on the right side of the window:

i) *Gain Ks* model gain coefficient ,

ii) *Time constant Ts* model time constant,

iii) *Sampling Time* sampling time T0 applied both for discrete simulation and calculations of PID discrete controller coefficients,

iv) *Control/Ref.input* level of step input for identification experiment (range +/- 1) or level of the reference signal for simulation (the best about is 50 deg),

v) *Clear* clears draw window,

vi) *Data acquisition* starts experiment,

vii) *Simulation Continuous or discrete* starts simulation with fixed Kr, Ki and Kd coefficients,

viii) *Save data* saves collected data to the mat-file,

ix) *Kr, Ki, Kd* PID controller parameters used in the simulation.

4.3. Models and experiments

The respective buttons in the column MODELS & EXPERIMENTS in the *Main Control Window* perform the following tasks:

1) Open loop opens the Simulink model to start real time experiments for the open loop system with no control

2) *PID controller* opens the Simulink model to start real time experiments for a closed-loop system with PID controller,

3) *State feedback closed-loop system* opens the Simulink model to start real time experiments for a closed-loop system with state feedback. This model can be used for experiments with LQ or deadbeat controllers,

4) *Time optimal* opens the Simulink model to start real time experiments for a closed-loop system with time optimal controller.

5) *Adaptive controller* opens the Simulink model to start real time experiments for a closed-loop system with adaptive controller.

The windows appearing after clicking on the respective buttons in the column MODELS & EXPERIMENTS have a similar structure, so that only *PID controller* window is described below. To activate this window double click *PID controller* button (Figure 4-11).

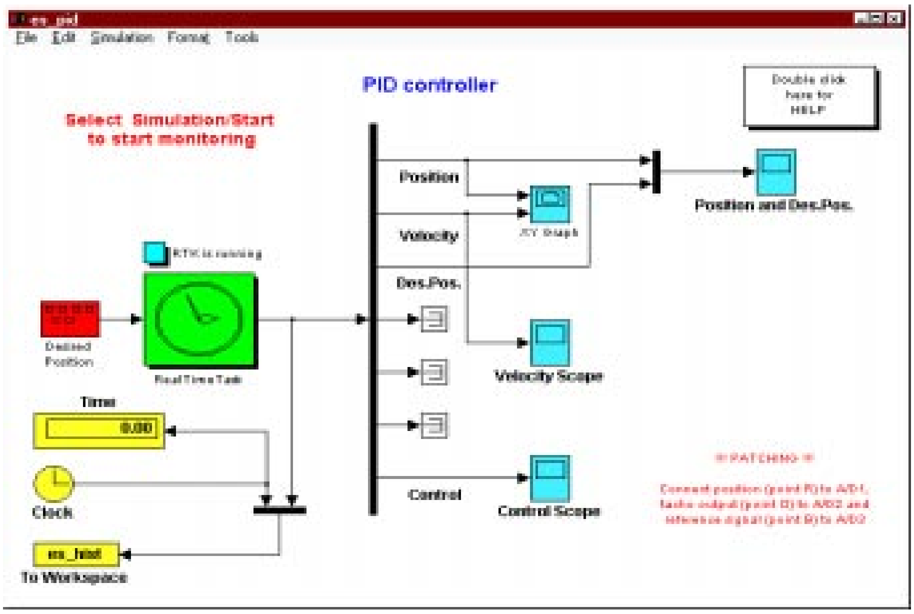
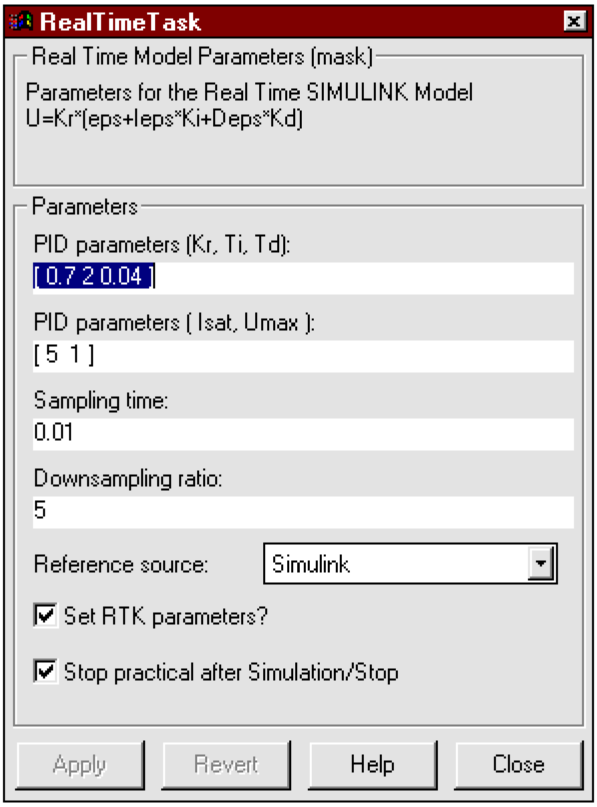


Figure 4-11: *PID controller* window

After double clicking the Real Time Task block you can set the following parameters (Figure 4-12):



Real Time Model Parameters (mask)

Parameters for the Real Time SIMULINK Model
U=Kr*(eps+Ieps*Ki+Deps*Kd)

Parameters

PID parameters (Kr, Ti, Td):
[0.7 2 0.04]

PID parameters (Isat, Umax):
[5 1]

Sampling time:
0.01

Downsampling ratio:
5

Reference source: Simulink

☑ Set RTK parameters?

☑ Stop practical after Simulation/Stop

Figure 4-12: *Real Time Task* window

- *Sampling time* defines sampling period during the experiment,
- *PID controller parameters,*
- *Downsampling ratio* - this parameter decreases the rate of output data flow from the output of *Real Time Task.* For example, if downsampling ratio is equal to 10, for each 10 samples only one is transferred to the output of the block*,*
- *Reference source* defines source of the reference signal for RTK task. There are three options selected using the list box:
  1. Simulink Signal Generator,
  2. External excitation (Generated by the Input Potentiometer IP150H)
  3. Internal excitation. – in this case the wave shape and parameters are set in the Window "*Set Param of the Internal Excitation Source*" from the Main Control menu
- *Set RTK Parameters -* if this check box is selected the parameters from the mask are passed to RTK as actual working parameters. In the opposite case the previously set parameters are valid.
- *Stop practical after Simulation/Stop* - the check box is used to define an action associated with the termination of simulation. If the box is marked the control is turned off after clicking the *Simulation/Stop* button in the window menu option.

If the box is not marked, the real time control should be stopped from the "*Stop practical*"(Red Button) in the Main menu window.

5. FIRST EXAMPLES

In this section two examples are presented. The first one uses *Modular Servo Toolbox* functions directly, the second shows how to use menu.

5.1. Example 1. Closed-loop control system with state feedback gains (application level 1).

We wish to design a closed-loop system having no oscillations. An illustration is provided in the M.file - mservo1.m.

Start this file by typing `mservo1` from the MATLAB Command window. The output from this program is shown in Figure 5-1.

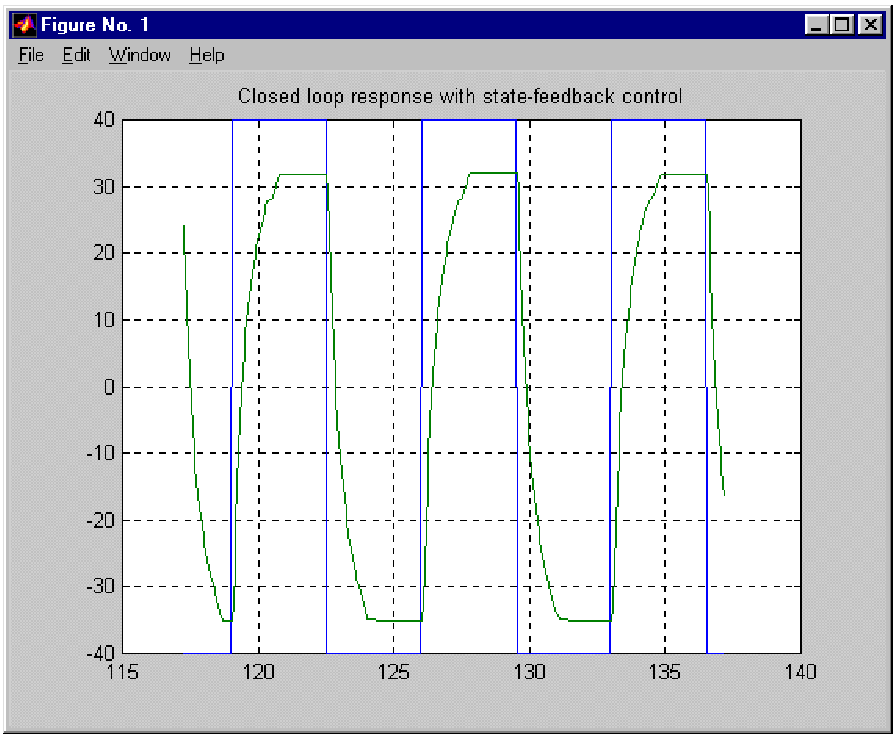


Figure 5-1: State feedback control experiment results

Note: The M files Mservo1, Mservo2, Mservo 3 and Mservo5 are set within these files to a PCL812 base address of 210H (528 decimal), and will therefore not work for any other base address unless modified. Open these files and change the base address to the one which is set for your system.

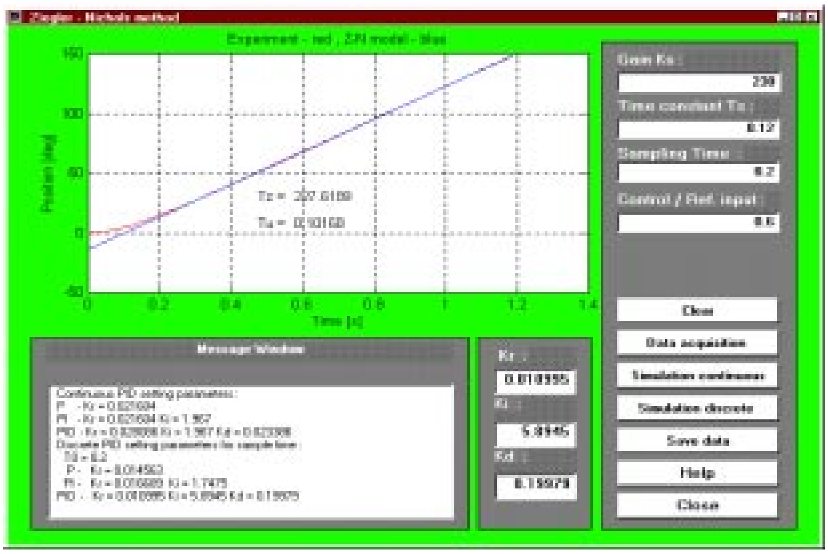
5.2. Example 2. Ziegler-Nichols method for designing PID controller
(application level 2).

After double-clicking *Ziegler-Nichols method* button in the *Main Control Window* (Figure 3-1) the screen presented in Figure 4-10 appears. Assume, that the model parameters are correctly identified, using the “Time Domain Identification toolbox” and you wish to design a discrete PID controller for a sampling time of T0 = 0.2 s.

Then click *Data acquisition* button to start the experiment. The step input excites the DC motor open-loop system and the output position is measured. The results are shown in Figure 5-2.

You will see initial part of the step response of the system and the straight line approximation which represents the Ziegler-Nichols model step response.

Two constants are calculated from the plot: - T_z – the tangent of the model response and T_u – the delay of the model. According to Ziegler-Nichols rules (see Advanced Teaching manual – 33-008-4M5) the coefficients of the continuous and discrete PID controllers can be calculated, and are displayed in the *Message Window*.



Gain Ks : 238

Time constant Ts : 0.12

Sampling Time : 0.2

Control / Ref. input : 0.6

Kr : 0.010995

Ki : 5.8945

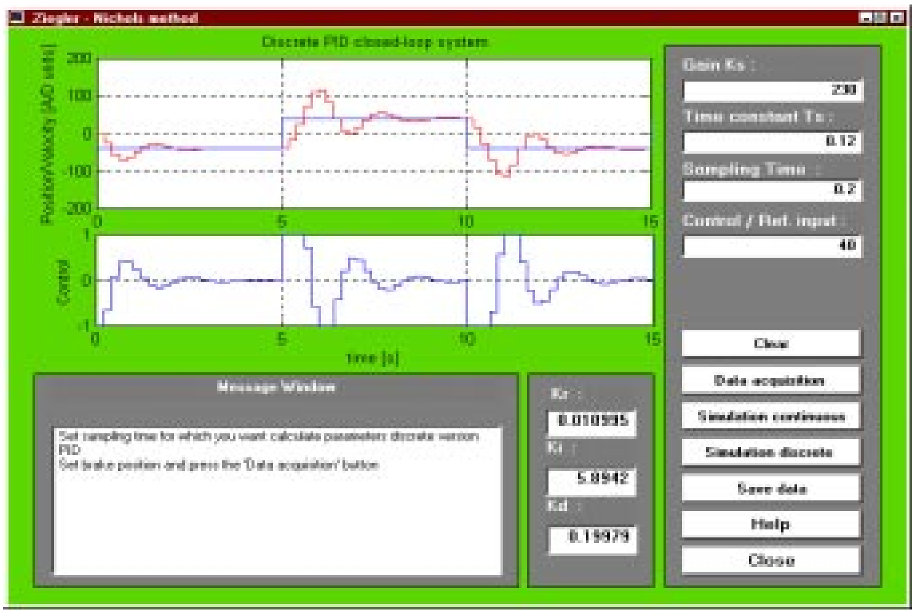
Kd : 0.15979

```
Continuous PID setting parameters:
P   - Kr = 0.021694
PI  - Kr = 0.021694 Ki = 1.967
PID - Kr = 0.028098 Ki = 1.967 Kd = 0.023398
Discrete PID setting parameters for sample time:
  T0 = 0.2
   P -  Kr = 0.014563
  PI -  Kr = 0.016689 Ki = 1.7475
PID -  Kr = 0.010995 Ki = 5.8945 Kd = 0.15979
```

Figure 5-2: *Ziegler-Nichols method* window after end of experiment

Now, define the level of a reference signal for a simulation experiment. Type the control value equal to 40 in the upper edit windows. Notice, that the parameters of the PID discrete controller are displayed in the edit windows. Then, click the *Simulation discrete* button. Figure 5-3in shows in the upper part the response of the closed-loop system to a square wave as reference signal. In the lower window you can see the control which is produced by PID controller. Note that the control exceeds the lower and upper limits (-1 and 1), and that the damping of the step response is very good with an overshoot of about 30%.

We conclude that this PID controller can properly control the servo system.



Gain Ks : 230

Time constant Ts : 0.12

Sampling Time : 0.2

Control / Ref. input : 40

Message Window

Set sampling time for which you want calculate parameters discrete version PID
Set brake position and press the 'Data acquisition' button

Kr : 0.010995

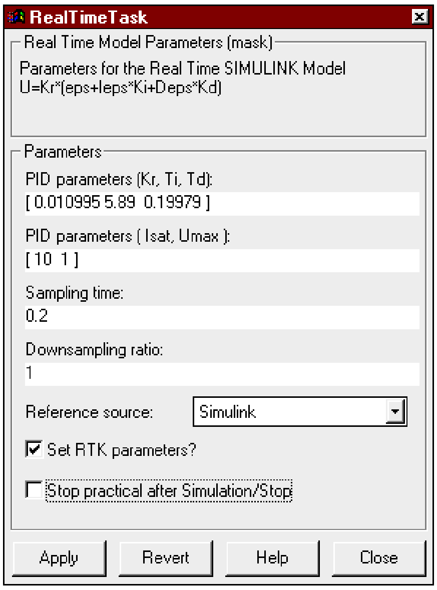
Ki : 5.8942

Kd : 0.19979

Figure 5-3: Simulation results (discrete version)

The values of Kr, Ki and Kd calculated by the Ziegler-Nichols method could then be used to optimally control the digital servo using a PID control algorithm.

Next go to *Main Control Window* and click *PID controller* button. The Figure 4-11 window opens. Open *Real Time Task* block (Figure 5-4). Type in the parameters of the PID controller and then close window.



Real Time Model Parameters (mask)

Parameters for the Real Time SIMULINK Model
U=Kr*(eps+Ieps*Ki+Deps*Kd)

Parameters

PID parameters (Kr, Ti, Td):
[0.010995 5.89 0.19979]

PID parameters (Isat, Umax):
[10 1]

Sampling time:
0.2

Downsampling ratio:
1

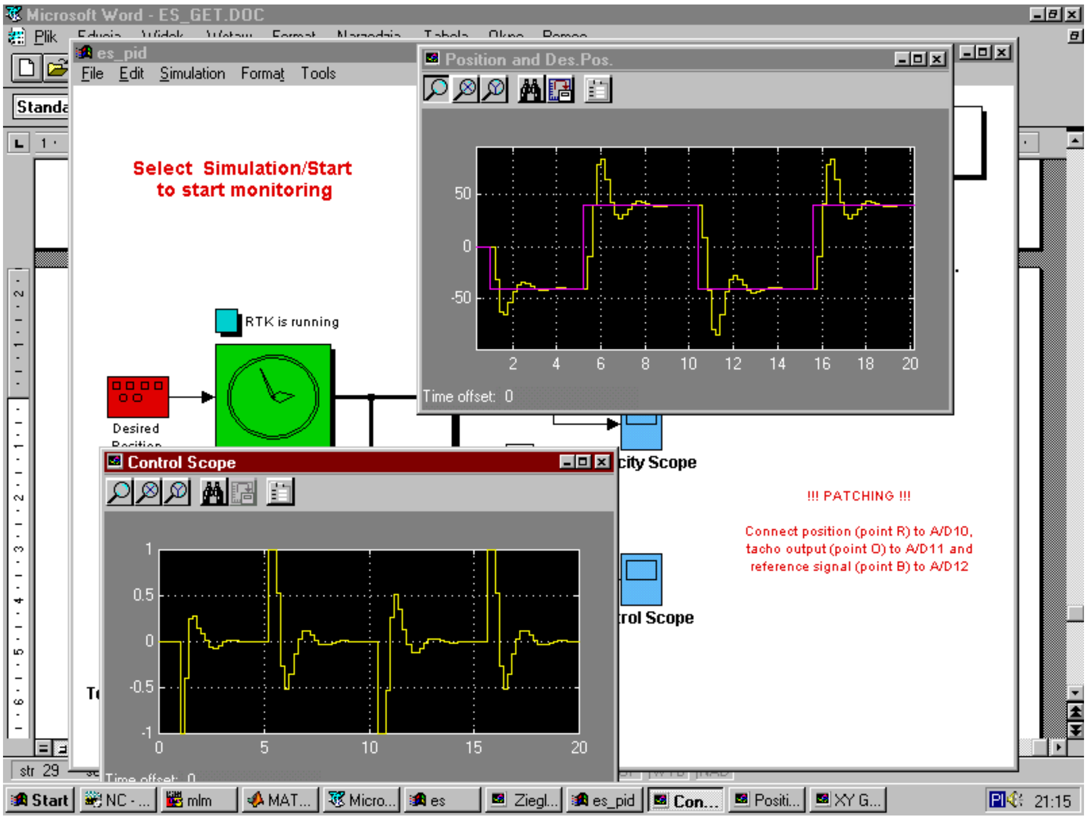
Reference source: Simulink

☑ Set RTK parameters?

☐ Stop practical after Simulation/Stop

Figure 5-4: Controller parameters (continuous version)

Click *Simulation/Start* in the window from Figure 4-11. The results of the experiment are shown in Figure 5-5.



Select Simulation/Start to start monitoring

!!! PATCHING !!!

Connect position (point R) to A/D10, tacho output (point O) to A/D11 and reference signal (point B) to A/D12

Figure 5-5: Results of the PID control experiment

Again we may conclude that the controller is effective.

6. CONNECTING THE PC AND THE SERVO

EQUIPMENT REQUIREMENTS- the following equipment is required:

Modular Servo system - Feedback part numbers

OA150A	Operational Amplifier
AU150B	Attenuator
PA150C	Pre-Amplifier
SA150D	Servo Amplifier
PS150E	Power Supply
DCM150F	DC Motor
IP150H	Input Potentiometer
OP150K	Output Potentiometer
LV150L	Magnetic Brake Load
GT150X	Gearbox/Tachometer
MS150Z	Accessories plus Magnetic Baseplate
33-301	Interface unit
33-300	Digital Encoder

PC with PCL-812PG or RT-DAC data acquisition board, and MATLAB, Simulink and the Modular Servo software installed. (See the Software Installation Manual - 33-000 for details of how to install the Modular Servo software).

6.1. PCL-812PG acquisition board:

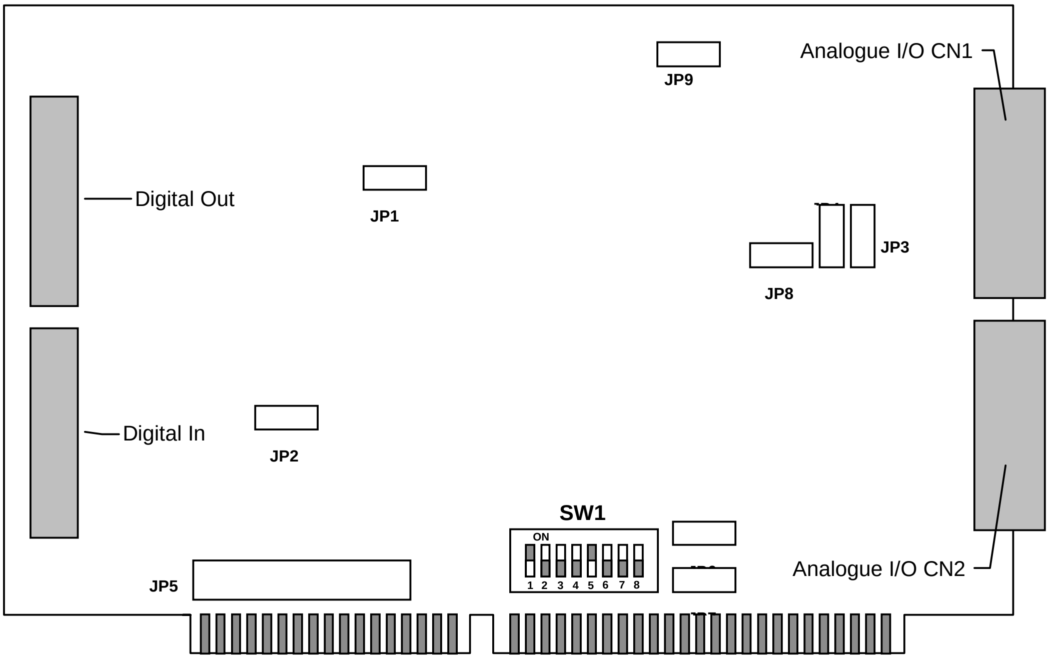


Figure 6-1: Switches and Jumpers of PCL-812 board

6.2. Switch settings - 16 bit board (PCL 812PG)

Switch SW1 controls the base address for I/O transfer (switches 1-6), and the CPU wait states (Switches 7-8). They should be set as follows (Figure 6-1).
(Note that the Modular Servo software always refers to the decimal settings, whereas the manual on the PCL-812 board refers to hexadecimal settings. For convenience, both decimal and hex addresses are shown in the table below).

SW1 Base Address Selection

I/O Address Range (hex and decimal)	**SW1 Switch position**					
	1	**2**	**3**	**4**	**5**	**6**
200-20F (512-527 Dec)	OFF	ON	ON	ON	ON	ON
210-21F (528-543 Dec)	OFF	ON	ON	ON	ON	OFF
220-22F (544-559 Dec) *****	OFF	ON	ON	ON	OFF	ON
230-23F (560-575 Dec)	OFF	ON	ON	ON	OFF	OFF
300-30F (768-783 Dec)	OFF	OFF	ON	ON	ON	ON
3F0-3FF (1008-1023 Dec)	OFF	OFF	OFF	OFF	OFF	OFF

***** Factory default setting

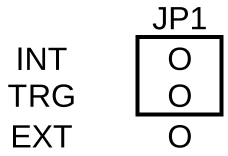
Note that some video cards or floppy disk drives use the higher base addresses such as 1008 (Hex 3F0), and should be avoided if possible.

Set the base address on the PCL-812PG board (switch SW1). The factory default base address is 220(HEX)

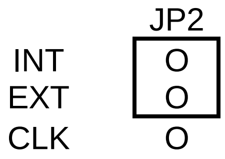
6.3. Jumper Settings

The Jumpers on the PCL-812PG board should be set in the positions indicated below.

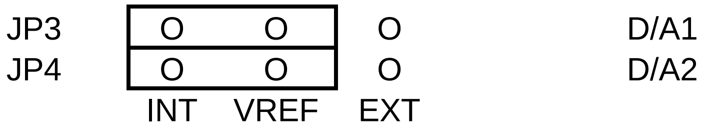
JP1 - Trigger Source Selection



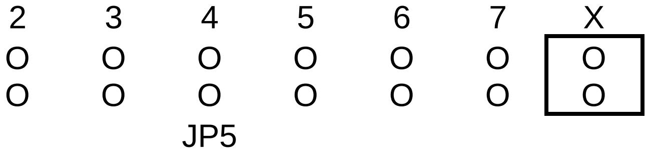
JP2 - Counter Input Clock Selection



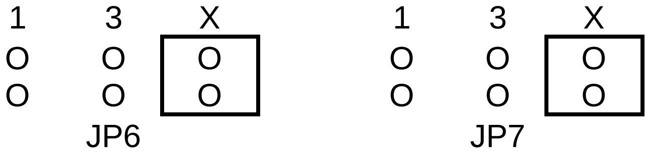
JP3, JP4 - D/A Reference Selection



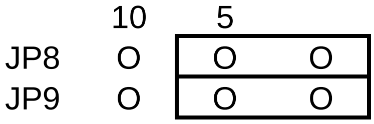
JP5 - IRQ Level Selection



JP6, JP7 - DMA Channel Selection



JP8, JP9 - D/A Internal Reference, A/D Maximum Input Voltage Selection



Depending on the specification and configuration of your PC you may need to change the Base Address and Wait State settings.

☞ *None of the other settings need to be changed.*

6.4. Switch SW1 settings for 8 bit full length PCL - 812 board

I/O Address Range (hex and decimal)	SW1 Switch position					
	1	**2**	**3**	**4**	**5**	**6**
200-20F (512-527 Dec)	ON	ON	ON	ON	ON	X
210-21F (528-543 Dec)	ON	ON	ON	ON	OFF	X
220-22F (544-559 Dec) *	ON	ON	ON	OFF	ON	X
230-23F (560-575 Dec)	ON	ON	ON	OFF	OFF	X
300-30F (768-783 Dec)	OFF	ON	ON	ON	ON	X
3F0-3FF (1008-1023 Dec)	OFF	OFF	OFF	OFF	OFF	X

Note: An “X” in the table above means that this switch is not used.

6.5. Installation of PCL-812 Board

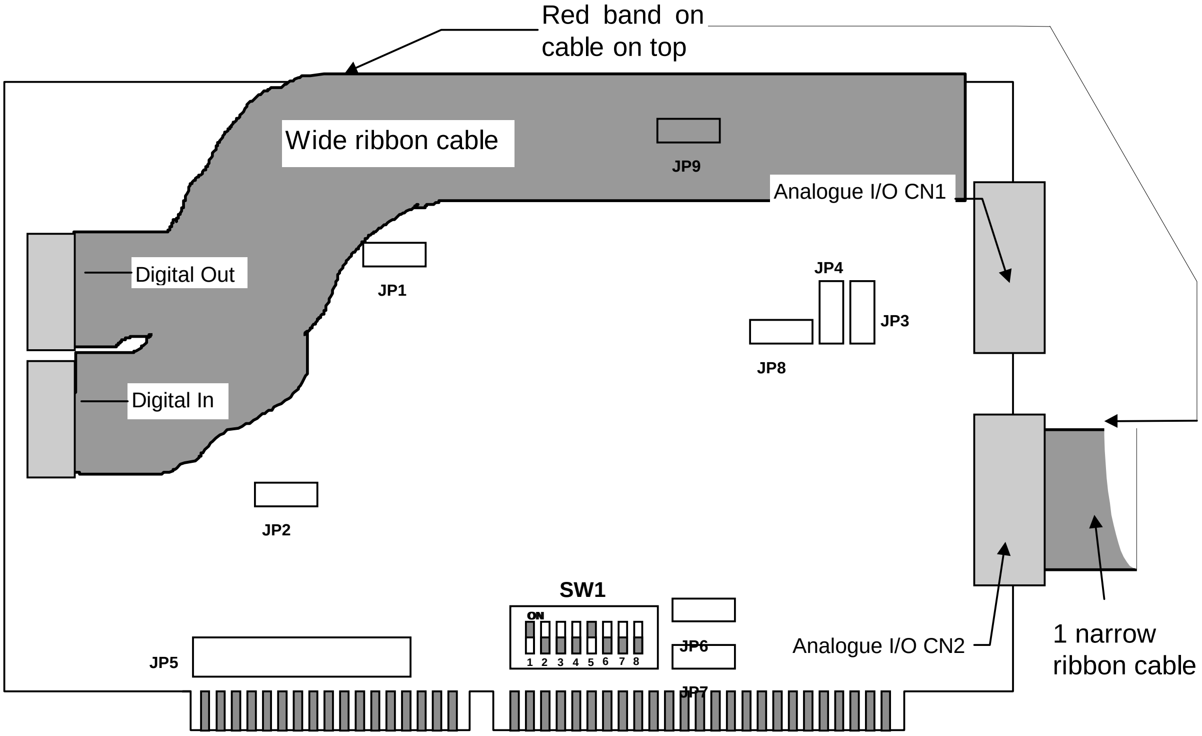


Figure 6-2: PCL- 812 Interface Card Ribbon Cable connections

See Figure 6-2. ***Two*** slots are required at the rear of the PC - one for the two 20 way analogue port sockets, and one for the 40 way socket at the end of the wide ribbon cable for the digital ports. Use screws to secure the card and ribbon socket plates to the casing of the PC.

Carefully install the interface card in a spare slot on the PC motherboard (or in a slot on the passive backplane if your PC has this type of architecture - common with some 19 inch rack mounted PC's). If in any doubt as to the correct procedure consult the documentation supplied with the PC.

Turn OFF and disconnect the mains power supply BEFORE opening the computer

Before opening the computer it is important to ensure that the computer is switched off and that the mains power supply is not connected.

Once you have opened the computer, check to see if there are any other cards installed, for example a network or sound card. It is important to determine the addresses of any such cards as your Modular Servo System will not function correctly if there is another card using all, or part of, the address space assigned to the PCL-812PG interface card.

Following the installation of the interface card into your PC, you are now ready to close the PC case and connect the ribbon cables to the Modular Servo system.

Connect the wide ribbon cable from the PC to the Modular Servo Digital Encoder unit (33-300). Connect the narrow ribbon cable from the PCL-812 connector CN2 (or JP4 on the RT-DAC board) to the 33-301 Analogue Control Interface unit. See Figure 6-3 and Figure 6-4 for details.

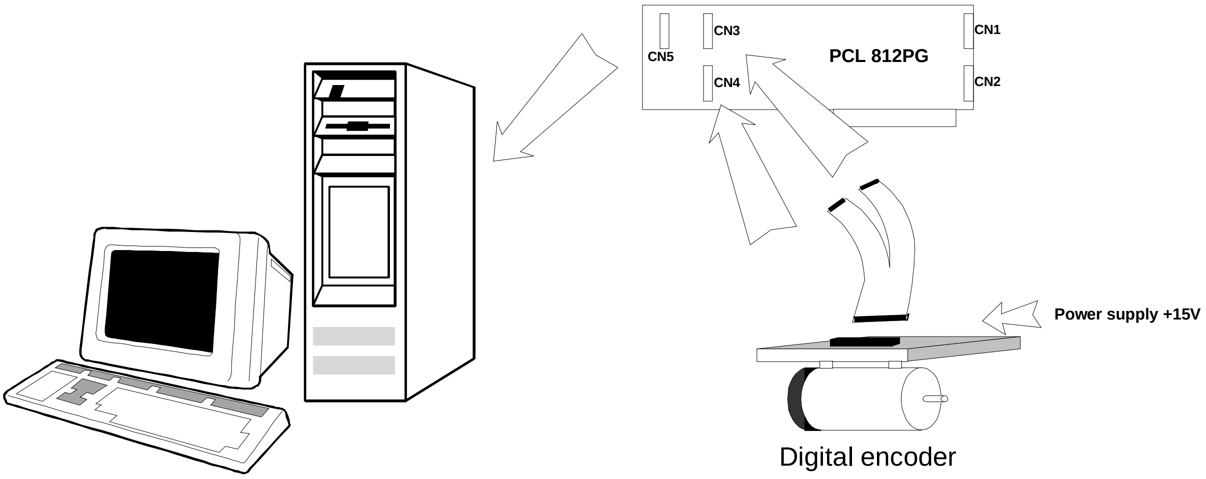


Figure 6-3: Connecting the digital encoder

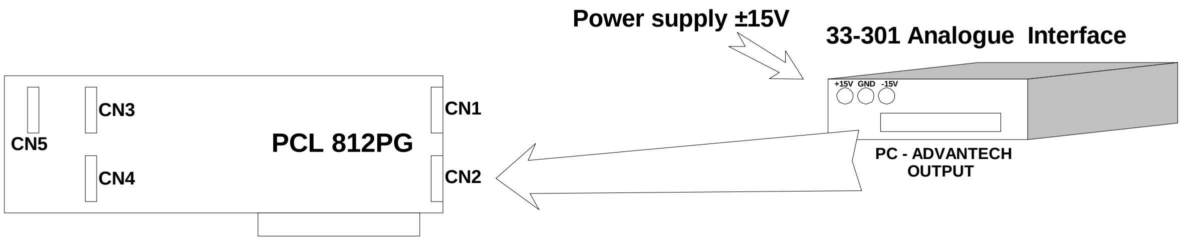


Figure 6-4: Data acquisition board and the 33-301 Analogue Control Interface connection

Connect the power supply PS150E ±15V and GND to the Analogue Control Interface unit 33-301.

Switch on all units,

Open the *Main Control Window* and click *Stop Practical* button,

If the DC motor turns, adjust the pre-amplifier 150C using the set zero potentiometer on the Modular Servo.

Run the tuning program - click *Calibration 33-301* button in the *Main Control Window* and follow the screen instructions.

Note: it is necessary to tune the Analogue Control Unit because the electronic circuits in the MS150 are not identical in all copies.

After calibration close Windows and turn off the computer.

Make general patching:

• connect **A/D1** input to point **3** on the output potentiometer θ_0,	OP150K
• connect **A/D 2** input to point **2** on the Gearbox Tachogenerator,	GT150X
• connect **D/A 2** output to point **1** on the Attenuator Unit	AU150B
• connect **A/D3** input to point **3** on the Input Potentiometer	IP150H

Connect up all the modules as shown in Figure 6-5.

Set the PA150C to the centre (normal) position.

Turn on the computer, start Matlab 5, and run *es* from the MATLAB command window

Your system is now ready to work.

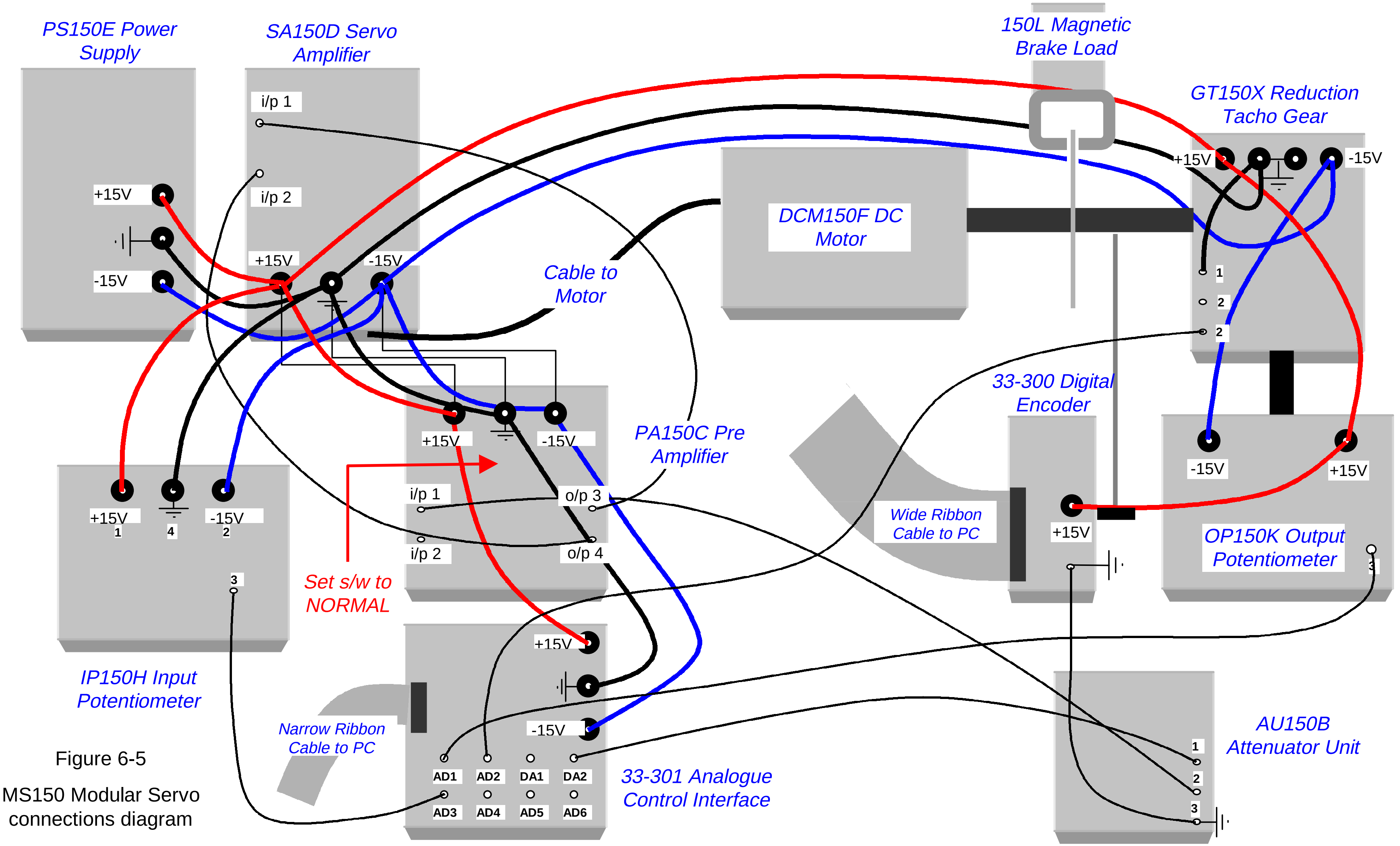


Figure 6-5

MS150 Modular Servo connections diagram

Notes